User's Guide

Release 7
User's Guide

Release 7
ISO 9001 Certification

This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by Bureau Veritas Quality International (BVQI) (Certification No. BVQI - 92053).

BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

Seventh Edition (October 2000)

This is a major revision of, and obsoletes, SH19-6795-05. See “Summary of Changes for NetView DM for MVS Release 7” on page xxiii for an overview of the changes made to this manual. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition applies to Release 7, Modification Level 0 of the licensed program NetView Distribution Manager for MVS, Program Number 5685-016, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, address your comments to:

Distribution Products Information Development
Rome Tivoli Laboratory
IBM Italia S.p.A.
Via Sciangai, 53
00144 Rome
Italy

Fax Number: (+39) 6 5966 2077
Internet ID: ROMERCF@VNET.IBM.COM

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

© Copyright International Business Machines Corporation 1988, 2000. All rights reserved.

May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished “as is” without warranty of any kind. All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Contents

Notices .......................................................... xv
Trademarks and Service Marks .............................. xvi

About This Document .............................................. xvii
Who Should Read This Document ........................... xviii
How This Document Is Organized ........................... xix
Conventions Used in This Document ....................... xix
Where to Find More Information ......................... xx
Tivoli NetView Distribution Manager for MVS Library ........ xx
Online Publications ........................................ xx
Tivoli NetView Distribution Manager for MVS Documents on CD-ROM ...... xxi
Related Publications .......................................... xxi

Summary of Changes for NetView DM for MVS Release 7 ................ xxiii
Sysplex Support ................................................... xxiii
SMS Extended Resource Repository ....................... xxiii
Delete Pending Request From Host ......................... xxiv
SQL Definition of SPMF Supergroups ....................... xxiv
Support for RACF Control on IOF Logon .................... xxiv
Control of Default Exit Options on SPMF Panels .............. xxiv
Extended CC Server and Clients Support .................... xxiv

Part 1. Using the Generalized Interactive Executive (GIX) .................. 1

Chapter 1. Getting Started ........................................ 5
Logging on to GIX .............................................. 5
The GIX Main Menu ........................................... 8
GIX Panel Layout ............................................. 10
Using Commands ................................................ 11
Moving between Panels ........................................ 13
How to Print Using GIX ....................................... 15
Exiting from GIX ............................................... 16

Chapter 2. Defining User Profiles ............................... 19
What Is a User Profile? ....................................... 19
Defining a User Profile Using GIX ........................... 20

Chapter 3. Defining Nodes ....................................... 33
What Is a Node Definition? ................................... 33
Defining Nodes Using GIX ..................................... 35
Using Transmission Profiles ................................. 46
Maintaining Logical Unit to Connection Profile Assignments ...... 50
Organizing Nodes into Groups ................................ 54
Defining a Group of Nodes Using GIX ....................... 56

Chapter 4. Defining Resources .................................. 61
What Is a Resource Definition? ............................... 61
Naming Resources ............................................. 65
Defining Resources Using GIX ............................... 69
<table>
<thead>
<tr>
<th>Chapter 14. Defining and Assigning Resources</th>
<th>239</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is a Resource Definition?</td>
<td>239</td>
</tr>
<tr>
<td>Naming Resources</td>
<td>243</td>
</tr>
<tr>
<td>Tracking Resources</td>
<td>247</td>
</tr>
<tr>
<td>SUBMIT Control Statements for LU0 Resource Types</td>
<td>248</td>
</tr>
<tr>
<td>SUBMIT Control Statements for LU 6.2 Resource Types</td>
<td>254</td>
</tr>
<tr>
<td>SUBMIT Control Statements for Tracking Resources</td>
<td>257</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15. Managing Resources in the Repository</th>
<th>261</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading Resources into the Resource Repository</td>
<td>261</td>
</tr>
<tr>
<td>Unloading Resources from the Repository</td>
<td>267</td>
</tr>
<tr>
<td>Using the DISP Operand</td>
<td>268</td>
</tr>
<tr>
<td>MAINTDAT Control Statements for All Resource Types</td>
<td>269</td>
</tr>
<tr>
<td>MAINTDAT Control Statements for Specific Resources</td>
<td>284</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 16. Preparing and Submitting Transmission Plans</th>
<th>313</th>
</tr>
</thead>
<tbody>
<tr>
<td>What Is a Transmission Plan?</td>
<td>313</td>
</tr>
<tr>
<td>Defining a Transmission Plan</td>
<td>317</td>
</tr>
<tr>
<td>Submitting a Transmission Plan</td>
<td>318</td>
</tr>
<tr>
<td>SUBMIT Control Statements for Transmission Plans</td>
<td>319</td>
</tr>
<tr>
<td>SUBMIT Transmission Functions for LU0 Node Types</td>
<td>327</td>
</tr>
<tr>
<td>SUBMIT Transmission Functions for LU 6.2 Node Types</td>
<td>343</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 17. Managing Submitted Plans</th>
<th>395</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printing the Contents of a Plan</td>
<td>395</td>
</tr>
<tr>
<td>Recovering from Transmission Failures</td>
<td>405</td>
</tr>
<tr>
<td>Managing Node Solicited Requests and Node Unsolicited Reports</td>
<td>408</td>
</tr>
<tr>
<td>TCFMAINT Control Statements</td>
<td>409</td>
</tr>
<tr>
<td>Erasing Plans Using the DB2 Environment</td>
<td>423</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 18. Maintaining NetView DM for MVS Files</th>
<th>429</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure for Maintaining NetView DM for MVS Files</td>
<td>429</td>
</tr>
<tr>
<td>Maintaining the Resource Repository</td>
<td>430</td>
</tr>
<tr>
<td>Maintaining the Plan Library</td>
<td>444</td>
</tr>
<tr>
<td>Maintaining the Distributed Resource Directory</td>
<td>449</td>
</tr>
<tr>
<td>Maintaining the TCF</td>
<td>452</td>
</tr>
<tr>
<td>Maintaining the Unsolicited Message File</td>
<td>457</td>
</tr>
<tr>
<td>Maintaining the NetView DM for MVS Work File</td>
<td>458</td>
</tr>
<tr>
<td>Maintaining the Exchange Information File (EIF)</td>
<td>459</td>
</tr>
<tr>
<td>Maintaining the TCP Work File</td>
<td>460</td>
</tr>
<tr>
<td>Maintaining the Request Queue File</td>
<td>462</td>
</tr>
<tr>
<td>Maintaining the GIX Data Sets</td>
<td>464</td>
</tr>
<tr>
<td>Maintaining the DSXJOBS File</td>
<td>466</td>
</tr>
</tbody>
</table>
### Part 3. Using the Interactive Operator Facility (IOF)

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Starting the Transmission Control Program (TCP)</td>
<td>471</td>
</tr>
<tr>
<td></td>
<td>What Is the TCP?</td>
<td>471</td>
</tr>
<tr>
<td></td>
<td>Preparing Job Control Statements</td>
<td>472</td>
</tr>
<tr>
<td></td>
<td>Specifying TCP Startup Parameters</td>
<td>475</td>
</tr>
<tr>
<td></td>
<td>Submitting the JCL</td>
<td>483</td>
</tr>
<tr>
<td></td>
<td>Handling Inconsistencies in the Network During TCP Cold Start</td>
<td>483</td>
</tr>
<tr>
<td>20</td>
<td>Logging On to IOF and Starting Transmission</td>
<td>485</td>
</tr>
<tr>
<td></td>
<td>Logging on to IOF</td>
<td>485</td>
</tr>
<tr>
<td></td>
<td>IOF Logon Panel</td>
<td>486</td>
</tr>
<tr>
<td></td>
<td>The Transmission Control Master Menu</td>
<td>488</td>
</tr>
<tr>
<td></td>
<td>IOF Panel Layout</td>
<td>489</td>
</tr>
<tr>
<td></td>
<td>IOF Operator Functions and Panel Flow</td>
<td>493</td>
</tr>
<tr>
<td></td>
<td>Option 1 – Start Transmission</td>
<td>494</td>
</tr>
<tr>
<td></td>
<td>Logging off from the IOF</td>
<td>494</td>
</tr>
<tr>
<td>21</td>
<td>Controlling Transmission Schedules</td>
<td>495</td>
</tr>
<tr>
<td></td>
<td>Option 2 – Schedule Display and Control</td>
<td>496</td>
</tr>
<tr>
<td></td>
<td>Displaying Selected Transmission Requests</td>
<td>502</td>
</tr>
<tr>
<td></td>
<td>Changing the Transmission Schedule</td>
<td>503</td>
</tr>
<tr>
<td>22</td>
<td>Controlling Transmission with the Nodes</td>
<td>515</td>
</tr>
<tr>
<td></td>
<td>Option 3 – Hold Transmission for a Node/Group</td>
<td>515</td>
</tr>
<tr>
<td></td>
<td>Option 4 – Release Transmission for a Node/Group</td>
<td>518</td>
</tr>
<tr>
<td>23</td>
<td>Changing Transmission Profiles</td>
<td>521</td>
</tr>
<tr>
<td></td>
<td>Using Transmission Profiles</td>
<td>522</td>
</tr>
<tr>
<td></td>
<td>Option 11 – Display/Change CURTASK or Transmission Profiles</td>
<td>524</td>
</tr>
<tr>
<td>24</td>
<td>Stopping Transmission and the Transmission Environment</td>
<td>529</td>
</tr>
<tr>
<td></td>
<td>Stopping Transmission</td>
<td>529</td>
</tr>
<tr>
<td></td>
<td>Stopping the TCP</td>
<td>530</td>
</tr>
<tr>
<td></td>
<td>Stopping the IOF</td>
<td>532</td>
</tr>
<tr>
<td>25</td>
<td>Handling Unsolicited Messages</td>
<td>533</td>
</tr>
<tr>
<td></td>
<td>Types of Messages</td>
<td>533</td>
</tr>
<tr>
<td></td>
<td>Routing of Messages</td>
<td>533</td>
</tr>
<tr>
<td></td>
<td>Viewing Unsolicited Messages</td>
<td>534</td>
</tr>
<tr>
<td>26</td>
<td>Handling Problems</td>
<td>537</td>
</tr>
<tr>
<td></td>
<td>The TCP Log File</td>
<td>537</td>
</tr>
<tr>
<td></td>
<td>Problems at the Host</td>
<td>538</td>
</tr>
<tr>
<td></td>
<td>Line and Node Problems</td>
<td>540</td>
</tr>
<tr>
<td></td>
<td>User Problems</td>
<td>541</td>
</tr>
<tr>
<td></td>
<td>Problems with the Service Provider</td>
<td>543</td>
</tr>
</tbody>
</table>
Part 4. Appendixes

Appendix A. Using MVS MODIFY Commands with NetView DM for MVS
- Summary of MVS MODIFY Commands Used in NetView DM for MVS
- Controlling the TCP
- Using the NetView DM for MVS Diagnostic Tools
- Ending the Service Provider

Appendix B. NetView DM for MVS Node, Function and Resource Types
- NetView DM for MVS Node Types
- NetView DM for MVS Functions
- NetView DM for MVS Resource Types

Appendix C. Resource Naming Conventions
- Resource Naming Conventions for LU 0 Resource Types
- Naming LU 0 Resources at the Node
- Naming Conventions for LU 6.2 Resource Types
- SNA/File Services Class Codes
- Character Sets

Bibliography
- TME 10 Software Distribution Version 3.1.5 Publications
- OS/390 Publications
- NetView Distribution Manager/2 Publications
- NetView Distribution Manager/6000 Publications
- AS/400 Publications
- MVS Publications
- VTAM Publications
- NCP Publications
- ISPF Publications
- Related Publications

Glossary

Index
Figures

1. Example of the Logon to GIX panel ........................................ 7
2. The NetView Distribution Manager – Main Menu ........................ 8
3. Example of GIX panel layout .................................................. 10
4. Example of a full-list panel .................................................... 13
5. Sample output of a GIX commands log ................................... 18
6. Example of a Set Profiles panel .............................................. 20
7. Example of a Change Password and GIX Authorizations panel ....... 21
8. Example of a Change Node Type Authorizations panel .................. 24
9. Example of a Change Resource Type Authorizations panel ............ 25
10. Example of a Specify Node Class Authorizations Of User panel ....... 26
11. Example of a Specify Resource Class Authorizations Of User panel ... 27
12. Example of a Specify User Defaults panel .................................. 28
13. Example of a Confirm the Deletion of a User panel ....................... 31
14. Example of a Configure Network panel ..................................... 35
15. Example of a Specify Node Attributes panel .............................. 37
16. Example of a Specify Node Authorization panel ........................... 41
17. Example of a Maintain LU/TP Assignments panel ......................... 47
18. Example of a Change LU/TP Assignments panel .......................... 48
19. Example of a Browse Selected Information panel (transmission profiles) 49
20. Example of a Maintain LU/CP Assignments panel ........................ 50
21. Example of a Change LU/CP Assignments panel .......................... 51
22. Example of a Browse Selected Information panel (connection profiles) 53
23. Example of a Manage Groups panel .......................................... 56
24. Example of a Define a Group of Nodes panel ................................ 57
25. Example of a Create/Change Node Groups panel .......................... 58
26. Example of a Define Resource panel ......................................... 70
27. Example of a Specify Operation On Resource panel ....................... 71
28. Example of a Specify Attributes For Resource panel ...................... 72
29. Example of Configure Network panel ........................................ 74
30. Example of a Specify Resource Assignment Operation panel .......... 74
31. Example of a Specify Node and Select Resource Type panel ............ 75
32. Example of an Assign Resources to Nodes panel ........................... 75
33. Example of a Select Resource Type For Grouping CMEP panel .......... 80
34. Example of a Specify Group and Select Operation panel ................ 81
35. Example of a Create/Change a Group panel ................................ 82
36. Example of a Specify Resource Assignment Operation panel .......... 86
37. Example of a Specify Node and Select Resource Type panel ............ 86
38. Example of an Assign Resources to Nodes panel ........................... 88
39. Example of a Create Resource History at the Node panel ............... 93
40. Example of a Browse Network panel ........................................ 98
41. Example of a Browse Selected Information panel .......................... 100
42. Format of variable-length records in user files ........................... 101
43. Variable-length records in the repository .................................. 102
44. Example of a Manage Resources panel ..................................... 107
45. Example of a List of Selected Resources panel ............................ 109
46. Example of an Identify the Resource panel ................................ 110
47. Example of an Identify the Resource panel for microcode ............... 110
48. Example of a Specify Attributes panel ...................................... 113
49. Structure of a transmission plan ............................................. 119
50. Example of a Prepare Plans panel .......................................... 123
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>List of GIX jump commands</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Examples of fastpaths</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Resource types and initial values</td>
<td>63</td>
</tr>
<tr>
<td>4</td>
<td>Comparison of naming conventions for LU0 and LU 6.2 resources</td>
<td>65</td>
</tr>
<tr>
<td>5</td>
<td>Resource history information for resource types</td>
<td>91</td>
</tr>
<tr>
<td>6</td>
<td>Possible NetView DM for MVS host CCSIDs</td>
<td>103</td>
</tr>
<tr>
<td>7</td>
<td>Possible NetView DM for MVS source CCSIDs</td>
<td>104</td>
</tr>
<tr>
<td>8</td>
<td>Possible DBCS CCSIDs</td>
<td>106</td>
</tr>
<tr>
<td>9</td>
<td>Values assigned to LBLRIND parameter</td>
<td>114</td>
</tr>
<tr>
<td>10</td>
<td>Plan status</td>
<td>151</td>
</tr>
<tr>
<td>11</td>
<td>Phase status</td>
<td>153</td>
</tr>
<tr>
<td>12</td>
<td>Phase by node status</td>
<td>155</td>
</tr>
<tr>
<td>13</td>
<td>Function status</td>
<td>159</td>
</tr>
<tr>
<td>14</td>
<td>Selecting transmission requests by plans, phases, node, or group name</td>
<td>174</td>
</tr>
<tr>
<td>15</td>
<td>Transmission function termination return codes</td>
<td>180</td>
</tr>
<tr>
<td>16</td>
<td>Batch Utility return codes</td>
<td>201</td>
</tr>
<tr>
<td>17</td>
<td>DD statements to run MAINTDAT</td>
<td>203</td>
</tr>
<tr>
<td>18</td>
<td>DD statements to run SUBMIT</td>
<td>205</td>
</tr>
<tr>
<td>19</td>
<td>DD statements to run TCFMAINT</td>
<td>207</td>
</tr>
<tr>
<td>20</td>
<td>Resource types and initial values</td>
<td>241</td>
</tr>
<tr>
<td>21</td>
<td>Comparison of naming conventions for LU0 and LU 6.2 resources</td>
<td>243</td>
</tr>
<tr>
<td>22</td>
<td>Resource history information for resource types</td>
<td>248</td>
</tr>
<tr>
<td>23</td>
<td>Applicability of ADD operands</td>
<td>250</td>
</tr>
<tr>
<td>24</td>
<td>Applicability of DELETE HISTORY Operands</td>
<td>259</td>
</tr>
<tr>
<td>25</td>
<td>Possible NetView DM for MVS host CCSIDs</td>
<td>264</td>
</tr>
<tr>
<td>26</td>
<td>Possible NetView DM for MVS source CCSIDs</td>
<td>264</td>
</tr>
<tr>
<td>27</td>
<td>Possible DBCS CCSIDs</td>
<td>265</td>
</tr>
<tr>
<td>28</td>
<td>(new,old,mod) values in relation to functions and resources</td>
<td>268</td>
</tr>
<tr>
<td>29</td>
<td>(keep,delete) values in relation to functions and resources</td>
<td>268</td>
</tr>
<tr>
<td>30</td>
<td>Applicability of CONVERT RESOURCE operands</td>
<td>272</td>
</tr>
<tr>
<td>31</td>
<td>Applicability of ERASE RESOURCE operands</td>
<td>276</td>
</tr>
<tr>
<td>32</td>
<td>Applicability of PREPARE RESOURCE Operands</td>
<td>279</td>
</tr>
<tr>
<td>33</td>
<td>Applicability of PRINT RESOURCE operands</td>
<td>282</td>
</tr>
<tr>
<td>34</td>
<td>Transmission functions for LU0 node types</td>
<td>327</td>
</tr>
<tr>
<td>35</td>
<td>Applicability of RECEIVE and RETRIEVE operands</td>
<td>334</td>
</tr>
<tr>
<td>36</td>
<td>Using asterisks to construct the name at the host</td>
<td>334</td>
</tr>
<tr>
<td>37</td>
<td>Transmission functions for node types</td>
<td>343</td>
</tr>
<tr>
<td>38</td>
<td>Functions you can use with LU 6.2 resources</td>
<td>344</td>
</tr>
<tr>
<td>39</td>
<td>Applicability of DELETE RESOURCE operands</td>
<td>355</td>
</tr>
<tr>
<td>40</td>
<td>Applicability of RETRIEVE RESOURCE operands</td>
<td>380</td>
</tr>
<tr>
<td>41</td>
<td>Applicability of SEND RESOURCE operands</td>
<td>389</td>
</tr>
<tr>
<td>42</td>
<td>Plan status</td>
<td>396</td>
</tr>
<tr>
<td>43</td>
<td>Phase status</td>
<td>397</td>
</tr>
<tr>
<td>44</td>
<td>Phase by node status</td>
<td>398</td>
</tr>
<tr>
<td>45</td>
<td>Function status</td>
<td>401</td>
</tr>
<tr>
<td>46</td>
<td>Transmission function termination return codes</td>
<td>405</td>
</tr>
<tr>
<td>47</td>
<td>VSAM Characteristics of the Library</td>
<td>430</td>
</tr>
<tr>
<td>48</td>
<td>VSAM Characteristics of the Holding File</td>
<td>430</td>
</tr>
<tr>
<td>49</td>
<td>JCL DD Statements Used by DSXBUBK and DSXBURS</td>
<td>443</td>
</tr>
<tr>
<td>50</td>
<td>VSAM Characteristics of the Plan Library</td>
<td>444</td>
</tr>
</tbody>
</table>
51. Space Required by the Plan Library Data Portion ........................................ 444
52. JCL DD Statements Used by PLCOPY .......................................................... 447
53. VSAM Characteristics of the DRD ................................................................. 449
54. Space Required by the DRD ........................................................................... 450
55. VSAM Characteristics of the TCF ................................................................. 452
56. Space Required by the TCF ........................................................................... 453
57. Space Occupied by Each Resource Record in the TCF ............................... 453
58. Space Occupied by Working Objects in the TCF ....................................... 454
59. VSAM Characteristics of the Unsolicited Message File ............................ 457
60. VSAM Characteristics of the NetView DM for MVS Work File .................. 458
61. VSAM Characteristics of the TCP Work File .............................................. 460
62. VSAM Characteristics of the Request Queue File ..................................... 462
63. Characteristics of GIX Data Sets ................................................................. 465
64. Displaying Selected Phases ......................................................................... 497
65. Displaying Selected NSRs ........................................................................... 498
66. Displaying Selected D&CC API Requests ................................................... 498
67. Abbreviated Forms of Functions and Resources ........................................ 511
68. MVS MODIFY Commands Used in NetView DM for MVS ...................... 549
69. Controlling the TCP: Mapping IOF Commands to MVS Commands .......... 551
70. NetView DM for MVS LU 0 Node Types ..................................................... 561
71. NetView DM for MVS LU 6.2 (user-defined) Node Types ....................... 562
72. Function Type Names and Type Codes ....................................................... 564
73. Functions and Nodes .................................................................................... 564
74. NetView DM for MVS Resource Types, Type Names, Type Codes, and Location ................................................................. 566
75. Resources and Nodes ................................................................................... 567
76. LU 6.2 Resources and Functions ................................................................. 568
77. Identification Tokens for All Objects Using Enterprise-Structured Names .... 575
78. Identification Tokens of the Architected Subtree for Group Objects .......... 576
79. Identification Tokens of the Architected Subtree for Refresh Change Objects ................................................................. 577
80. Identification Tokens of the Architected Subtree for Update Change Objects ................................................................. 578
81. Identification Tokens of the Architected Subtree for Fix Change Objects ...... 579
82. Identification Tokens of the Architected Subtree for Library Change Objects ................................................................. 580
83. Identification Tokens of the Architected Subtree for File Member Change Objects ................................................................. 581
84. Identification Tokens of the Architected Subtree for AS/400 Save Restorable Object Type Change Objects ................................................................. 582
85. Identification Tokens for Microcode Objects Using Structured Names ....... 583
86. Identification Tokens for Microcode Customization Structured Names ....... 583
87. NetView DM for MVS Resource Types and SNA/File Services Class Codes ................................................................. 584
88. Character Set 1134 ..................................................................................... 585
References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user.

Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

You can send license inquiries, in writing, to the

IBM Director of Licensing,
IBM Corporation,
North Castle Drive,
Armonk,
New York
10504-1785,
U.S.A.

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:

IBM Corporation
P.O. Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including, in some case, payment of a fee.
## Trademarks and Service Marks

The following are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both:

<table>
<thead>
<tr>
<th>IBM</th>
<th>MVS/XA</th>
</tr>
</thead>
<tbody>
<tr>
<td>The IBM logo</td>
<td>NetView</td>
</tr>
<tr>
<td>Tivoli</td>
<td>Operating System/2</td>
</tr>
<tr>
<td>The Tivoli logo</td>
<td>Operating System/400</td>
</tr>
<tr>
<td>ACF/VTAM</td>
<td>OS/2</td>
</tr>
<tr>
<td>AIX</td>
<td>OS/390</td>
</tr>
<tr>
<td>AIXwindows</td>
<td>OS/400</td>
</tr>
<tr>
<td>Application System/400</td>
<td>PC/XT</td>
</tr>
<tr>
<td>APPN</td>
<td>Presentation Manager</td>
</tr>
<tr>
<td>AS/400</td>
<td>PS/2</td>
</tr>
<tr>
<td>AT</td>
<td>RACF</td>
</tr>
<tr>
<td>AnyNet</td>
<td>RETAIN</td>
</tr>
<tr>
<td>BookManager</td>
<td>RISC System/6000</td>
</tr>
<tr>
<td>CICS</td>
<td>RMF</td>
</tr>
<tr>
<td>Common User Access</td>
<td>SAA</td>
</tr>
<tr>
<td>CUA</td>
<td>Systems Application Architecture</td>
</tr>
<tr>
<td>C/370</td>
<td>SystemView</td>
</tr>
<tr>
<td>DATABASE 2</td>
<td>System/36</td>
</tr>
<tr>
<td>DB2</td>
<td>System/370</td>
</tr>
<tr>
<td>DFMS/MVS</td>
<td>System/390</td>
</tr>
<tr>
<td>Extended Services</td>
<td>S/390</td>
</tr>
<tr>
<td>LANDP</td>
<td>Tivoli</td>
</tr>
<tr>
<td>Library Reader</td>
<td>Tivoli Enterprise</td>
</tr>
<tr>
<td>MVS</td>
<td>TME</td>
</tr>
<tr>
<td>MVS/ESA</td>
<td>VSE/ESA</td>
</tr>
<tr>
<td>MVS/DFP</td>
<td>VTAM</td>
</tr>
<tr>
<td>MVS/SP</td>
<td></td>
</tr>
</tbody>
</table>

Microsoft, MS-DOS, Windows, Windows NT, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of the Open Group in the United States and other countries.

C-bus is a trademark of Corollary, Inc.

Java, Hot Java, and all Java-based trademarks or logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

PC Direct is a trademark of Ziff Communications Company.

Other company, product, and service names mentioned in this document may be trademarks or service marks of others.
About This Document

This document is designed to help system administrators, end users, and network operators use the IBM NetView Distribution Manager/MVS (NetView DM for MVS), running under the MVS operating system.

The objective of this document is to familiarize you with the concepts of distributing data objects and installing software and microcode changes throughout a Systems Network Architecture (SNA) network. The document provides step-by-step instructions on how to use the following NetView DM for MVS user interfaces:

- The *generalized interactive executive (GIX)*
- The *batch utilities*
- The *interactive operator facility (IOF)*

This document does not detail how to use the other NetView DM for MVS interfaces and the *Software Profile Management Facility (SPMF)*.

Who Should Read This Document

This document is intended for people that have the following types of jobs:

- **System administrators**, responsible for setting up and maintaining the NetView DM for MVS environment after it has been installed. The *administration tasks* include:
  - Defining a user profile for each NetView DM for MVS user
  - Defining the nodes in the network to NetView DM for MVS
  - Defining the resources in the network to NetView DM for MVS
  - Organizing nodes and resources into groups
  - Assigning resources to nodes
  - Tracking resources using resource history records
  - Maintaining the NetView DM for MVS files

Administrators should have a basic knowledge of IBM Multiple Virtual Storage (MVS), virtual storage access method (VSAM), and Virtual Telecommunications Access Method (VTAM/TCAM).

- **Users**, performing the central NetView DM for MVS task of distributing resources throughout the network. The *end use tasks* include:
  - Preparing resources for transmission
  - Preparing and submitting transmission plans
  - Managing transmission plans
  - Monitoring the progress of transmissions
  - Unloading resources after transmission
  - Preparing messages

Users should have a basic knowledge of MVS and the IBM Interactive System Productivity Facility (ISPF), and also the particular operating system used at the node.
Network operators, responsible for controlling and monitoring the transmission of resources. The operation tasks include:
- Starting transmission
- Monitoring and controlling transmission
- Stopping transmission
- Handling unsolicited messages, and transmission problems

Operators should have some knowledge of MVS and VTAM.

All users should be familiar with data communications, particularly the basic concepts and terminology of SNA.

How This Document Is Organized

Use this document after NetView DM for MVS has been installed, and after the users, nodes, and resources in the network have been clearly identified during the customization task. You should also read the General Information document for an introduction to NetView DM for MVS and definitions of the main NetView DM for MVS terms.

There are five user interfaces to NetView DM for MVS:
- The generalized interactive executive (GIX)
- The batch utilities
- The interactive operator facility (IOF)
- The Software Profile Management Facility (SPMF), which is not described in detail in this document.

System administrators and end users can use either the interactive GIX interface or the noninteractive batch utilities, to perform the administration and end use tasks. Network operators can either use the IOF interface to perform the operation task, or they can enter MVS commands directly from the system console.

The document is divided into three parts, so you only need to use the part which describes the interface that you are using:
- Read Part 1, “Using the Generalized Interactive Executive (GIX)”, to use GIX to perform the administration task of setting up and maintaining NetView DM for MVS, and the end use task of distributing resources.

Begin by reading Chapter 1, “Getting Started” on page 5 to find out how to use GIX. Chapter 2, “Defining User Profiles” on page 19 through Chapter 5, “Assigning Resources to Nodes” on page 85 describe the administration tasks. Chapter 6, “Preparing Resources for Transmission” on page 97 through Chapter 10, “Preparing Messages” on page 191 describe the end use tasks to distribute resources.

You may sometimes need to refer to the batch utilities part of this document, as the fields on GIX panels correspond to batch utility parameters. For help with a particular field, you can either use the online help or refer to the corresponding batch utilities parameter.

Each chapter presents introductory concepts, followed by instructions which you can follow at your terminal.
• Read **Part 2, “Using the Batch Utilities”**, to use a set of batch utilities to perform the administration task of setting up and maintaining NetView DM for MVS, and the end use task of distributing resources.

Begin by reading Chapter 11, “Getting Started” on page 197 to find out how to use the batch utilities. Chapter 12, “Defining User Profiles” on page 211 through Chapter 14, “Defining and Assigning Resources” on page 239, and Chapter 18, “Maintaining NetView DM for MVS Files” on page 429, describe the administration tasks. Chapter 15, “Managing Resources in the Repository” on page 261 through Chapter 17, “Managing Submitted Plans” on page 395 describe the end use tasks to distribute resources.

Each chapter presents introductory concepts, followed by examples of the control statements you can use.

• Read **Part 3, “Using the Interactive Operator Facility (IOF)”**, to use IOF to control and monitor the transmission of resources.

Begin by reading Chapter 19, “Starting the Transmission Control Program (TCP)” on page 471 to find out how to start transmission. Chapter 19, “Starting the Transmission Control Program (TCP)” on page 471 through Chapter 26, “Handling Problems” on page 537 describe the operation tasks.

To control the transmission of resources using MVS commands directly from the NetView console, refer to Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549.

Individual chapters are listed at the beginning of each part. The document also contains other appendixes, a bibliography, glossary, and index.

---

### Conventions Used in This Document

This document uses several typeface conventions for special terms and actions. These conventions have the following meaning:

**Bold**

Lowercase or mixed-case commands, command options in a syntax statement, the names of objects in procedures that you type, select, or click, or any other information that you must use literally appear like this, in **bold**.

**Italics**

Variables (except within code examples), new terms, and values that you must provide appear like *this*, in *italics*. Words and phrases that are emphasized also appear like *this*, in *italics*.

**Monospace**

Code examples, output, and system messages appear like *this*, in a monospace font. Within code examples, variables appear in monospace font within brackets, for example `<variable>`.
Where to Find More Information

This section describes the other documents in the Tivoli NetView Distribution Manager for MVS library.

Tivoli NetView Distribution Manager for MVS Library

The Tivoli NetView Distribution Manager for MVS library contains the following documents:

*General Information*, GH19-6792, introduces the NetView DM product family and describes the specific facilities offered by NetView DM for MVS. Begin by reading this book for an introduction to what NetView DM for MVS does.

*Installation and Customization*, SH19-6794, describes what you need to install NetView DM for MVS, and how to customize NetView DM for MVS for your environment, using either an interactive dialog or batch jobs. There are also details for migrating from previous releases, including how to migrate from VSAM data sets to IBM's DATABASE 2 (DB2) relational database. The book also describes how to install the SPMF feature.

*Overview and Scenarios*, GH19-6797, describes the structure of NetView DM for MVS, how it works, and how you use it. This book provides scenarios to illustrate the ways in which you can use NetView DM for MVS in different environments.

*SPMF User's Guide*, SH19-4148, describes how to use the Software Profile Management Facility (SPMF) feature of NetView DM for MVS to create transmission plans, distribute software, and manage your network.

*SPMF Database Model*, SH19-4160, provides information about the structure of the NetView DM for MVS and SPMF database model based on a DATABASE 2™ (DB2™) relational database.

*Base Application Programming*, SH19-6958, is for users who develop application programs that use the NetView DM for MVS base programming interface.

*Diagnosis*, LY19-6374, defines how to isolate the cause of a NetView DM for MVS problem and how to extract additional data for analysis. The book also explains how to apply a program temporary fix (PTF).

Online Publications

Softcopy versions of the documentation in BookManager format are available in the *Tivoli Online Library Omnibus Edition OS/390 Collection*, order number SK2T-6700 (December 2000).

Softcopy versions of the documentation in Adobe Acrobat PDF format are available at [http://www.support.tivoli.com/Prodman/html/AB.html](http://www.support.tivoli.com/Prodman/html/AB.html) Choose Availability, then Tivoli NetView Distribution Manager for MVS.
Tivoli NetView Distribution Manager for MVS Documents on CD-ROM

The NetView DM for MVS Release 7 publications are available in displayable softcopy form on CD-ROM as part of:

- IBM OS/390 Softcopy Collection Kit, SK2T-6700
- IBM MVS Softcopy Collection Kit, SK2T-0710

The entire library is available, except for the Diagnosis book.

Customers currently licensed for OS/390, MVS, and Networking Systems Software products are entitled to a free copy of the corresponding CD-ROM Collection Kits, which are shipped on request.

Extra copies of the CD-ROM Collection Kits can be ordered for an annual subscription charge.

Related Publications

The following publications may also be useful:

- System Modification Program Extended Reference, SC28-1107
- OS/390 SMP/E Reference, SC28-1806
- ISPF User's Guide, SC34-4484
- ISPF Dialog Developer's Guide and Reference, SC34-4486
- OS/390 TSO/E Command Reference, SC28-1969
- DFSMS/MVS Access Method Services for ICF, SC26-4906
- DFSMS/MVS Utilities, SC26-4926
- OS/390 SMP/E Commands, SC28-1805
- OS/390 SMP/E Messages and Codes, SC28-1738
- OS/390 MVS JCL Reference, GC28-1757
- DB2 for MVS/ESA Administration Guide, SC26-3265
- DB2 for MVS/ESA Messages and Codes, SC26-3268
- DB2 for MVS/ESA SQL Reference, SC26-3270
- IBM DB2 for OS/390 V 5 Administration Guide, SC26-8957
- IBM DB2 for OS/390 V 5 Application Programming and SQL Guide, SC26-8958
- IBM DB2 for OS/390 V 5 Command Reference, SC26-8960
- IBM DB2 for OS/390 V 5 Utility Guide and Reference, SC26-8967
- IBM DB2 for OS/390 V 5 Messages and Codes, GC26-8979
- IBM DB2 for OS/390 V 5 SQL Reference, SC26-8966
- IBM DB2 for MSV/ESA V 4 Administration Guide, SC26-3265
Where to Find More Information

- IBM DB2 for MVS/ESA V 4 Application Programming and SQL Guide, SC26-3266
- IBM DB2 for MVS/ESA V 4 Command Reference, SC26-3267
- IBM DB2 for MVS/ESA V 4 Utility Guide and Reference, SC26-3395
- IBM DB2 V 4 Messages and Codes, SC26-3268
- IBM DB2 V 4 SQL Reference, SC26-3270
- IBM DATABASE 2 Version 3 Administration Guide, SC26-4888
- IBM DATABASE 2 Version 3 Application Programming and SQL Guide, SC26-4889
- IBM DATABASE 2 Version 3 Command and Utility Reference, SC26-4891
- IBM DATABASE 2 Version 3 Messages and Codes, SC26-4892
- IBM DATABASE 2 Version 3 SQL Reference, SC26-4890 SC26-4378
Summary of Changes for NetView DM for MVS Release 7

NetView DM for MVS Release 7 contains the following enhancements:

- Sysplex support
- SMS extended resource repository
- Delete pending request from host
- SQL definition for SPMF supergroups
- Support for RACF control on IOF logon
- Control of default exit options on SPMF panels
- Extended CC clients support

Sysplex Support

Each component of NetView DM for MVS Release 7 can now run in a separate OS/390 Parallel Sysplex, by implementing the XCF cross system communication protocol and SMSVSAM for sharing VSAM files.

This means that, for example, TCP, GIX, SPMF and BU can be run simultaneously in different OS/390 partitions of the same Sysplex.

To take advantage of this facility, SMSVSAM must be configured and activated, and the appropriate options selected in the installation and customization phases.

Warning: If you attempt to run different modules of NetView DM for MVS in different OS/390 partitions of the same Sysplex without configuring and activating SMSVSAM, there is a risk of data corruption. However, if you run all modules of NetView DM for MVS in the same Sysplex partition, there is no need to activate SMSVSAM.

D&CC API No Longer Supported

The Distribution and Change Control Application Programming Interface (D&CC API), is no longer supported or delivered as part of the product.

SMS Extended Resource Repository

It is now possible to extend the maximum size of the SMS Resource Repository above the previous limits of 4 GB. By taking advantage of the XRBA architecture the maximum size has been raised to 400 GB, the limitations of the initialization procedure making it impossible to realize the theoretical maximum of $2^{64}$ bytes.

To take advantage of this facility you must be using DFSMS Version 1.5 or later, and must define an SMS Data Class for the Extended Addressability (see full details in the NetView DM for MVS User's Guide, and Installation and Customization).
**Delete Pending Request From Host**

Implemented concurrently in this release and in TME 10 Software Distribution Version 3.1.5, this new feature allows NetView DM for MVS Release 7 to delete any request in *pending* status at the remote destination, whenever the plan or phase in which the request is included is deleted at the host.

By *pending* status is intended plans or phases that have been transmitted to the remote destination but have not yet been executed.

---

**SQL Definition of SPMF Supergroups**

The Software Profile Management Facility (SPMF) allows you to define supergroups as being names of nodes, groups and other supergroups linked by Boolean operators (for example, and, or). It is now also possible to define supergroups using an SQL expression to search the DB2 tables maintained by SPMF, or any other accessible DB2 tables.

This allows you to use a different naming convention for nodes and groups, taking advantage of the power of the SQL search language.

---

**Support for RACF Control on IOF Logon**

NetView DM for MVS now supports Resource Access Control Facility (RACF) control for operators logging on to Input-Output Formatting (IOF). An additional parameter has been added to one of the customization macros (see the NDMTCP macro description in the *NetView DM for MVS Installation and Customization*).

---

**Control of Default Exit Options on SPMF Panels**

A new feature has been added whereby it is possible to set parameters that insert an ‘E’ in the command line of the SPMF panels, such that SPMF exits from the panel flow without processing the panel if Enter is pressed inadvertently. A separate parameter controls the default exit for each panel.

The default exits are enabled when SPMF is started, and it is necessary to close and restart SPMF to enable any changes to them. The parameters are detailed in the table NVDM_PARAMETER in the *SPMF Database Model Manual*.

---

**Extended CC Server and Clients Support**

TME 10 Software Distribution Version 3.1.5, which carries out NetView DM for MVS-originated change control operations at distributed workstations, now supports the following platforms:
<table>
<thead>
<tr>
<th>Platform</th>
<th>Version</th>
<th>Server</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>2000 Professional and Server</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>NT 3.51 and 4.0 (SP5 &amp; 6A)</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>3.11, 95 and 98</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>OS/2</td>
<td>3.0x - 4.5</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AIX</td>
<td>3.2.5 - 4.3.3</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>NetWare</td>
<td>4.11 - 4.2x</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Part 1. Using the Generalized Interactive Executive (GIX)

This part describes GIX, a menu-driven ISPF application program with online help facilities.

**Note:** Not all of the attached nodes support the parameters that NetView DM for MVS can generate. For specific limitations, please refer to the appropriate node documentation.

Chapter 1. Getting Started ...................................... 5
Logging on to GIX ........................................... 5
Logging on from TSO ..................................... 5
Logging on from ISPF/PDF ................................. 6
The Logon to GIX Panel ................................. 7
What Could Go Wrong ................................ 7
GIX in Katakana ........................................ 7
The GIX Main Menu ....................................... 8
Getting Help ............................................ 8
Selecting from the Main Menu .......................... 9
GIX Panel Layout ....................................... 10
Using Commands ......................................... 11
Primary Commands and PF Keys ....................... 11
Line Commands on Full-list Panels ..................... 12
Moving between Panels ................................... 13
Jump Commands ......................................... 14
Jump Functions ......................................... 14
Going to ISPF from GIX ................................. 15
How to Print Using GIX ................................ 15
Printing GIX Panels ..................................... 16
Printing GIX Reports ................................... 16
Printing Resources ..................................... 16
Exiting from GIX ....................................... 16
The GIX Commands Log ................................ 17

Chapter 2. Defining User Profiles .............................. 19
What Is a User Profile? ................................ 19
Defining a User Profile Using GIX ..................... 20
Changing a User Profile ................................ 29
Deleting a User Profile ................................ 30

Chapter 3. Defining Nodes ..................................... 33
What Is a Node Definition? ............................. 33
Node Types ............................................. 33
Defining Nodes Using GIX .............................. 35
Changing a Node Definition ............................ 43
Deleting a Node Definition ............................. 44
Browsing Node Definitions ............................. 45
Printing Node Definitions ............................. 45
Using Transmission Profiles ......................... 46
Changing Logical Unit to Transmission Profile Assignments ......... 47
Browsing Logical Unit to Transmission Profile Assignments .......... 49
Printing Logical Unit to Transmission Profile Assignments .......... 49
Maintaining Logical Unit to Connection Profile Assignments .......... 50
Defining Logical Unit to Connection Profile Assignments .................. 50
De-assigning a Connection Profile ........................................ 51
Changing the Logical Unit Assignment to a Connection Profile ........ 51
Browsing and Printing LU/CP Assignments ................................ 52
Browsing Logical Unit to Connection Profile Assignments ............ 52
Organizing Nodes into Groups .............................................. 54
   Node Groups of the Same Type ........................................ 54
   Group of LU 6.2 Nodes with Different Node Types .................. 54
   Automatic Node Groups (Server Node Groups) ....................... 55
Defining a Group of Nodes Using GIX ..................................... 56
   Changing a Node Group .............................................. 59
   Deleting a Node Group .............................................. 59
   Browsing Node Groups .............................................. 60
   Printing Node Groups .............................................. 60

Chapter 4. Defining Resources ............................................. 61
   What Is a Resource Definition? ...................................... 61
   Resource Types .................................................. 61
   Naming Resources .................................................. 65
      Naming LU0 Resources ......................................... 65
      Naming LU 6.2 Resources ...................................... 65
   Partial Naming and Matching for LU 6.2 Resource Types .......... 67
Defining Resources Using GIX .............................................. 69
   Changing a Resource Definition ................................... 76
   Deleting a Resource Definition ................................... 76
   Browsing Resource Definitions .................................. 77
   Printing Resource Definitions .................................. 77
Organizing Resources into Groups ........................................ 79
Defining a Resource Group Using GIX ................................... 79
   Changing a Resource Group ....................................... 82
   Deleting a Group of Resources ................................... 83
   Browsing Resource Groups ....................................... 83
   Printing Resource Groups ....................................... 84

Chapter 5. Assigning Resources to Nodes ................................. 85
   What Is a Resource Assignment? .................................. 85
   Assigning Resources Using GIX .................................... 85
      Changing the Attributes of a Resource Assignment ............ 89
      Deleting a Resource Assignment ................................ 90
   Tracking Resources .................................................. 90
   Defining a Resource History Record Using GIX ...................... 92
      Changing a Resource History Record .......................... 94
      Deleting a Resource History Record .......................... 95

Chapter 6. Preparing Resources for Transmission ........................ 97
   Browsing the Network Configuration ................................ 97
   Loading Resources into the Repository ................................ 100
      Variable-Length Records ...................................... 101
      Uniform Text Processing ....................................... 102
      Locking Resources in the Resource Repository .................. 106
   Loading Resources Using GIX ...................................... 107
      Replacing or Appending Resources .............................. 117
      Browsing or Printing the Repository Directory ................. 117
      Deleting Resources from the Repository ....................... 117
Chapter 1. Getting Started

This chapter shows you how to log on to the generalized interactive executive (GIX), and provides you with the information you need to begin using GIX. This includes:

- The NetView Distribution Manager main menu
- The layout of GIX panels
- The commands you can use
- How to move between panels
- How to print using GIX
- How to exit from GIX

Logging on to GIX

If you are the first person to use GIX after NetView DM for MVS is installed, you should also read Chapter 2, “Defining User Profiles.” If you are not the first person to use GIX, your system administrator will already have defined your user profile. Ask your administrator for an authorized user identification and password.

How you log on to GIX depends on how your system is set up. At present, GIX runs on the MVS operating system. There are two ways in which you can log on to GIX from MVS:

- Directly from the time sharing option (TSO)
- From the ISPF/PDF primary option menu

Before you can log on, you will need the following:

- Your TSO user identification (ID)
- Your TSO password
- A GIX profile, predefined at installation time

The instructions given below may vary, depending on how your environment is set up.

Logging on from TSO

To log on to GIX from TSO, follow these instructions:

1. Log on to TSO using an authorized TSO user ID and password obtained from your administrator.

   **Note:** It is recommended that you specify at least 8000 bytes in the Size field of the TSO logon panel.

   The following TSO system prompt is displayed:

   Ready

2. Type `GIX SUFFIX(nn)`, where `nn` is a two-digit number, at the TSO system prompt. This number is the GIX profile number defined when NetView DM for MVS was installed.

   Press Enter. The system displays the following message:

   GIX START-UP IN PROGRESS
Getting Started

If you have a GIX password, a panel is displayed for you to enter the password. If you do not have a GIX password, the primary menu panel of GIX is displayed.

3. After GIX start-up is complete, press Enter to display the NetView Distribution Manager–Logon to GIX panel. This panel is shown in Figure 1 on page 7.

4. Type the GIX password assigned to you by your administrator, and press Enter. If you log on successfully, the system displays the NetView Distribution Manager–Main Menu. This panel is shown in Figure 2 on page 8.

“What Could Go Wrong” on page 7 lists the error messages you might receive if the logon is unsuccessful.

Logging on from ISPF/PDF

You can log on from ISPF/PDF, providing your TSO administrator has changed the ISPF/PDF Primary Option Menu at installation time. The Installation and Customization book explains how to change this menu.

To log on to GIX from ISPF/PDF, follow these instructions:

1. Log on to TSO using an authorized TSO user ID and password obtained from your administrator. The system displays the following TSO system prompt:

   Ready

2. Type ISPF at the TSO system prompt, then press Enter.

   The system displays the ISPF/PDF Primary Option Menu.

3. Type G nn, where nn is a two-digit number, on the OPTION line. This number is the GIX profile number defined when NetView DM for MVS was installed.

   Press Enter. The system displays the following message:

   GIX START-UP IN PROGRESS

   If you have a GIX password, a panel is displayed for you to enter the password. If you do not have a GIX password, the primary menu panel of GIX is displayed.

4. After GIX startup is complete, press Enter to display the NetView Distribution Manager–Logon to GIX panel. This panel is shown in Figure 1 on page 7.

5. Type the GIX password assigned to you by your administrator, then press Enter. If you log on successfully, the system displays the NetView Distribution Manager–Main Menu. This panel is shown in Figure 2 on page 8.

   “What Could Go Wrong” on page 7 lists the error messages you may receive if the logon is unsuccessful.
The Logon to GIX Panel

This is the panel you use to log on to GIX:

```
NDMPMOAE NETVIEW DISTRIBUTION MANAGER–LOGON TO GIX
Command ==> 15:52
Enter:
Password . . .

Tracking information:
User ID . . . GIACOP
PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR
```

Figure 1. Example of the Logon to GIX panel

What Could Go Wrong

If the logon fails, either from TSO or ISPF/PDF, you might get one of the following error messages:

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Excessive number of incorrect passwords</td>
</tr>
<tr>
<td>16</td>
<td>User not authorized to use NetView DM for MVS</td>
</tr>
<tr>
<td>24</td>
<td>Open snap dataset failed</td>
</tr>
<tr>
<td>28</td>
<td>Print allocation/dealloc. open/close errors</td>
</tr>
</tbody>
</table>

The Messages and Codes book describes each of the error messages, and explains how to solve the problem.

Note: If you are working with the Service Provider (SVP) installed and you receive an error message while logging on to GIX, you should always check whether the SVP is active. Refer to “Problems with the Service Provider” on page 543 for information about how to handle problems related to the SVP.

GIX in Katakana

You can customize GIX to work with a Katakana terminal, keyboard, and printer. You can then write the user text for plans, phases, and node definitions in Katakana. The messages sent to nodes will also appear in Katakana. All of the characters on GIX panels will appear in uppercase (capital) letters.
To work in Katakana, you must perform the following steps:

- Customize the ISPF profile for each GIX Katakana user
- Specify the Katakana facility in the user profile of each GIX Katakana user
  (user profiles are described in Chapter 2, “Defining User Profiles” on page 19)

To customize the ISPF profile, follow these instructions:

1. Display the ISPF Primary Option Menu.
2. Select Option 0, ISPF Parameters and press Enter.
3. On the next menu, select Option 1, Terminal and press Enter.
4. Specify the type of terminal you are using.

The GIX Main Menu

Once you have logged on to GIX, the NetView Distribution Manager–Main Menu is displayed. This panel is called the GIX main menu throughout this book.

![Figure 2. The NetView Distribution Manager – Main Menu](image)

Getting Help

Once you have logged on to GIX, you can press PF1 at any time to get help. Pressing PF1 displays some help text that explains the particular panel. Press PF3 when you want to leave the help facility.

Sometimes, while you are using GIX, a message will appear on line 3 (just below the command or selection field). If you press PF1 when such a message is displayed, the help text will explain what the message means, and what to do about it. However, if you want help text for the panel and not the message, you must correct whatever caused the message to appear, and then you can use PF1 to get help for the panel.
Selecting from the Main Menu

The term select is used throughout this book to mean the following:

1. Type an option number in the selection field.
2. Press Enter. The system displays the next panel.

This is how you select an option from the GIX main menu.

The following information shows you which chapter of this book to turn to for information about each option on the GIX main menu:

**Option 1, Configure Network**, provides access to the distributed resource directory (DRD) for system administration. The DRD contains information about the NetView DM for MVS network. You use this option to define nodes and resources to NetView DM for MVS, and assign resources to nodes. These tasks are described in the following chapters:

- Chapter 3, “Defining Nodes” on page 33
- Chapter 4, “Defining Resources” on page 61
- Chapter 5, “Assigning Resources to Nodes” on page 85

**Option 2, Browse Network**, enables you to view or print the information about nodes and resources that you defined to NetView DM for MVS. This task is described in “Browsing the Network Configuration” on page 97.

**Option 3, Manage Resources**, provides access to the NetView DM for MVS resource repository, which contains software and data resources. Use this option to load and unload resources to and from the resource repository. These tasks are described in the following chapters:

- Chapter 6, “Preparing Resources for Transmission” on page 97
- Chapter 9, “Managing Resources after Transmission” on page 183

**Option 4, Prepare Plans**, enables you to prepare, validate, and submit a transmission plan. Transmission plans initiate and control transmissions between the host and the nodes. These tasks are described in Chapter 7, “Preparing and Submitting Transmission Plans” on page 119.

**Option 5, Manage Submitted Plans**, allows you to look at the status of plans you have submitted to the transmission control file (TCF) for processing. You can also perform operations on these plans, such as deleting or resetting a plan. These tasks are described in Chapter 8, “Managing Submitted Plans” on page 147.

**Option 6, Handle Messages**, shows you how to prepare messages at the host to send to nodes, and to view messages received from DPCX nodes. These tasks are described in Chapter 10, “Preparing Messages” on page 191.

**Option 7, Manage Groups**, enables you to define a set of nodes or resources as a group in the DRD. These tasks are described in the following chapters:

- Chapter 3, “Defining Nodes” on page 33
- Chapter 4, “Defining Resources” on page 61

**Option 8, Set Profiles**, enables you to define user authorizations and defaults in a user profile. This is described in Chapter 2, “Defining User Profiles” on page 19.
Option F, Go To ISPF, enables you to access ISPF from GIX. This is described in “Going to ISPF from GIX” on page 15.

GIX Panel Layout

Figure 3 shows an example of a GIX panel. The text below the illustration explains each element on the panel.

1 NDMPPYRE
2 SPECIFY ATTRIBUTES FOR VSE DATASET
3 Command ==> 4 scrollPAGE

Enter desired values or accept the ones shown:

1 Class . . . Resource authorization class
2 Restriction__ 1 = Yes 2 = No (only for INSERT operation)
3 Automatic assignment__ 1 = Yes 2 = No (only if Restriction = 1)

Tracking information:
Activity . . :
Resource name:

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR

Figure 3. Example of GIX panel layout

Note: The numbers 1 through 7 in the above figure are used for illustrative purposes only. They are not part of the GIX panel.

1 Panel Identification
Each panel has an identification name, for example, NDMPPYRE. This identification is not displayed automatically. To switch the display on, enter a PANELID ON command on the command line. Refer to “Using Commands” on page 11 for information about GIX commands.

2 Panel Title
Each panel in GIX has a title. It is always centered on the top line of the panel. Sometimes, the title includes the type of node and the type of resource that you are working with, in this case, a VSE data set.

3 Command Line
Line 2 of each panel is the command line or selection line, depending on the type of panel. You enter primary commands or selection options on this line. The commands you can enter are described in “Using Commands” on page 11.

4 Scroll
There are some types of panel which you can scroll forwards on, using the PF8 key, or backwards using the PF7 key. These panels are called full-list panels. In
the upper right corner of the screen there is a scroll field, which shows you how much the screen will scroll when you use the PF keys. To change this value, simply move the cursor to the scroll field and type in a new value over the one displayed. You can enter the following values:

- 1 to 9999 to scroll a certain number of lines
- P (Page) to scroll one page
- H (Half) to scroll half a page
- M (Maximum) to scroll to the top or bottom of the data
- C (Cursor) to move the line where the cursor is positioned to the top or bottom of the screen, depending on whether you press PF7 or PF8.

5 Input Fields
This area may contain a list of options, as on the GIX main menu, or data entry fields where you enter information. These fields vary from panel to panel.

6 Tracking Information
As you move from panel to panel performing a specific task, tracking information is displayed to remind you of what task you are performing. For example, while creating a new node (called ROME), you would see:

```
Tracking information:
Activity : CREATE
Node ... : ROME
```

7 Program Function (PF) Keys
The bottom lines of each panel show which PF keys you can use to carry out certain functions.

---

Using Commands

You can use two types of command on GIX panels:

- Primary commands and a set of corresponding PF keys
- Line commands on full-list panels

Primary Commands and PF Keys

To communicate with GIX, you enter primary commands in the command line at the top of the panel. If the cursor is not already on the command line, you can move it there using the PF12 key.

There are also a set of defined PF keys which correspond to these commands. The PF keys that you can use on the panel currently displayed are shown at the bottom of the panel. The primary commands you can enter, and their corresponding PF keys are listed below. The list shows two numeric values:

- If you have 12 PF keys on your terminal, use the lower number
- If you have 24 PF keys, you can also use the higher number

Note: You cannot change the functions assigned to PF keys.

**HELP (PF1 /13)** Displays help information for specific commands and operations.

**SPLIT (PF2 /14)** Splits the screen between GIX and ISPF/PDF (if permitted).
Using Commands

END (PF3/15) Terminates your current operation and returns you to the previous level panel. If you enter this command from the GIX main menu, you are logged off from GIX.

RETURN (PF4/16) Returns you directly to the GIX main menu. You can also use this command after you have entered a HELP or JUMP command. This returns you to the panel from which you entered the HELP or JUMP command.

UP (PF7/19) Scrolls a panel upwards.

DOWN (PF8/20) Scrolls a panel downwards.

SWAP (PF9/21) Moves the cursor to wherever it was previously positioned on the other logical screen of a split screen pair.

PRINT (PF10/22) Copies and saves the screen image for printing when you exit from GIX.

CURSOR (PF12/24) Moves the cursor to the command or selection line.

You can also enter the following commands in full on the command line:

PANELID ON Displays a panel identifier on line 1 of the current and subsequent panels. Use the PANELID OFF command to switch off the panel identifiers. The default is PANELID OFF, so panel identifiers are not automatically displayed when you first log on to GIX.

QSPACE Queries the amount of space left in the resource repository, or the plan library, depending on which panel you enter the command from.

You can also use the following commands, but only when a specific panel prompts you to do so:

SAVE Records the contents of a transmission plan in the plan library, or a group of plans in the DRD.

DELETE Deletes an item from a Delete Confirmation panel.

L name Scrolls directly to the item called name on a full-list panel.

Line Commands on Full-list Panels

You enter line commands in the line command area of full-list panels. There are various full-list panels that you can request using GIX, for example:

- All the nodes defined in the DRD
- All the resources defined in the DRD
- All the transmission plans stored in the plan library
- All the transmission plans, phases, and functions, submitted to the TCF
Here is an example of a full-list panel, to save a transmission plan that you are creating or changing:

```
NDMPK2AE CREATE OR CHANGE A PLAN
Command ==> ScrollPAGE
14:11
Primary command : SAVE
Line commands: S(Select)–R(Repeat)–M(Move)–C(Copy)–A(After)–B(Before)
D(Delete)–I(Insert)–BS(Begin Sequence)–ES(End Sequence)
FT(Change Function Termination values)–G(Get Plan)
FI(Fast Insertion from DRD)–FR(Fast Insertion from Repository)
CMD ACTIVITY (CREATE)

Figure 4. Example of a full-list panel
```

There are some line commands that are available on all full-list panels. These are the GIX line commands that you can use on full-list panels:

- **B (Browse)** To browse an entry on the panel
- **D (Delete)** To delete an entry from the panel
- **I (Insert)** To insert a new line after a line
- **P (Print)** To print an entry on a panel
- **R (Repeat)** To duplicate a particular line
- **C (Copy)** To copy a particular line
- **S (Select)** To select an entry on a panel for further operations

You can enter the Delete, Repeat, and Copy commands with double characters at the beginning and at the end of a block of lines. For example:

```
dd- PLANJUNE
--- OLDPLAN
dd- PLANJULY
```

This deletes the lines from PLANJUNE to PLANJULY (including these lines).

You can enter the Delete, Repeat, and Insert commands followed by a number composed of one or two digits (nn), to show how many lines the command applies to. For example:

```
r3- PLANJULY
```

This repeats the line PLANJULY three times.

There are also line commands that are specific to certain panels. These are described for each relevant panel later in this book.

---

**Moving between Panels**

A quick way to move between panels is to use *jump commands* and *jump functions*. 
Jump Commands

To use jump commands, enter a command on the command line (the top line of the panel), and then press Enter. The commands you can use are shown in the Command column of Table 1. The second column shows the corresponding jump function.

<table>
<thead>
<tr>
<th>Command</th>
<th>Jump Function</th>
<th>To select the following GIX option</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNET</td>
<td>=1</td>
<td>Configure Network</td>
</tr>
<tr>
<td>BNET</td>
<td>=2</td>
<td>Browse Network</td>
</tr>
<tr>
<td>RESO</td>
<td>=3</td>
<td>Manage Resources</td>
</tr>
<tr>
<td>PLAN</td>
<td>=4</td>
<td>Prepare Plans</td>
</tr>
<tr>
<td>TRACK</td>
<td>=5</td>
<td>Manage Submitted Plans</td>
</tr>
<tr>
<td>GROUP</td>
<td>=7</td>
<td>Manage Groups</td>
</tr>
</tbody>
</table>

Table 1. List of GIX jump commands

Once you have entered a command, such as BNET, the GIX panel for that option will be displayed. When you have finished, press PF4 to return to the panel you originally entered the jump command from.

Jump Functions

Jump functions are also called fastpaths in NetView DM for MVS. To use fastpaths, you enter a sequence of numbers on the command or selection line at the top of each panel. This sequence corresponds to the flow of panels that you want to move through. When you press Enter, GIX takes you to the panel you requested.

When you request a fastpath from any panel other than the GIX main menu, you must insert an equal sign (=) before the command string, to specify that the jump is to be made through the GIX main menu. For example:

Command ==> =. This returns you to the GIX main menu.
Command ==> =1;4 This moves you to Option 4, Define Node, of Option 1, Configure Network from the GIX main menu.

You must separate the numbers with delimiters. The default delimiter is a semicolon (;), but you can change this using the PARMS option of ISPF. If you do not insert the value between delimiters, the default value will be used. The default is equivalent to the following instruction:

Leave the Selection field blank and press Enter

For example, the following fastpath displays the List of Defined Resources full-list panel:

Command ==> =1;1;1;; 1; Configure Network 1; Define Resources 1; Dataset selected as resource type ; No specification of resource name
As you become more familiar with GIX, you will probably find that the fastpaths reduce the time you spend moving between panels. The following table shows some more examples of fastpaths:

<table>
<thead>
<tr>
<th>Function</th>
<th>To do this with GIX:</th>
</tr>
</thead>
<tbody>
<tr>
<td>=1;2;1</td>
<td>Assign a resource to a node</td>
</tr>
<tr>
<td>=1;2;2</td>
<td>Change the attributes of an assigned resource</td>
</tr>
<tr>
<td>=1;2;3</td>
<td>Delete resource assignment</td>
</tr>
<tr>
<td>=1;1;1</td>
<td>Define a resource</td>
</tr>
<tr>
<td>=3;1</td>
<td>Manage a data set in the resource repository</td>
</tr>
<tr>
<td>=4;;</td>
<td>Obtain a full-list of plans</td>
</tr>
<tr>
<td>=6;1</td>
<td>Handle the input messages for DPCX</td>
</tr>
<tr>
<td>=8;;</td>
<td>Obtain a full-list of all the defined users</td>
</tr>
</tbody>
</table>

Table 2. Examples of fastpaths

Going to ISPF from GIX

You can use Option F on the GIX main menu to access ISPF without ending the current GIX session. You can only use this feature on an MVS host when the licensed program ISPF/PDF is installed. There are two ways in which you can access ISPF without leaving GIX:

- Press PF3 repeatedly to return to the GIX main menu, if you are currently using another panel.
  
  Type F on the selection line of the GIX main menu and press Enter. The ISPF/PDF Primary Option Menu is displayed.

- Press PF2 to display a split screen dialog. Half the screen displays a GIX screen, while the other half displays an ISPF/PDF screen. Press PF5 if you want to move the cursor between screens.

There are two options on the ISPF/PDF Primary Option Menu that are particularly useful:

- **Option 2**, which is the ISPF Edit Facility. You can use this to prepare MVS procedures to be submitted at the end of a phase, or NetView DM for MVS batch utility control statements for SYSLIB files.

- **Option 3**, which is the ISPF Utility Menu. Option 3.8 on this menu is useful for reviewing the status and output of NetView DM for MVS batch jobs submitted from GIX.

How to Print Using GIX

You can use GIX to print the following items:

- GIX panels
- GIX reports
- Resources contained in the NetView DM for MVS resource repository

If the Katakana feature is installed, you can print GIX output on a Katakana printer. English letters will be printed in uppercase, but information entered as Katakana will be printed as Katakana.
Exiting from GIX

Printing GIX Panels
You can request the hard copy of any GIX panel by pressing PF10 or PF22. Some panels display the primary command P (Print). You can enter this command on the command line to print the specific panel. The screen image is stored in the ISPF file. You can then print this file from ISPF when you exit from GIX.

Printing GIX Reports
There are several reports that you can print using GIX, using the Print option, which is displayed on specific GIX panels.

If the JES/328X Print Facility is activated and defined in the NDMGIXnn generation macro, then these reports are routed to the local printer. The ID of the local printer is defined in your user profile. Printouts are automatically printed, and the data set deleted, at the end of the GIX session.

If the JES/328X Print Facility is not defined in the NDMGIXnn macro, then you can get a report on the system printer at the end of the GIX session. You do this using the data set called TSO userid.NDMGIX.PRINT01, which is automatically allocated when you start GIX. Refer to the Installation and Customization book for information about this data set, and how to set up GIX printing facilities.

Printing Resources
You can use GIX to request a printout of several types of resource, using a command called PDA (Print Data). You enter this command on the command line of the List of Selected Resources full-list panel. This panel, and how to print resources, is described in Chapter 9, “Managing Resources after Transmission” on page 183.

Exiting from GIX
Follow these instructions to exit from GIX:

1. Press PF3 or PF15 repeatedly, to return to the GIX main menu.
2. Press PF3 or PF15 once more on the GIX main menu. You will see a message like this:
   ENTRY (A) userid.NDMGIX.PRINT01 DELETED
3. Press Enter to display the Specify Disposition of Log/List Data Set panel. This panel asks you what you want to do with the log data set generated during this GIX session. The log parameters are not saved. This log is described below in “The GIX Commands Log” on page 17.
4. Enter one of these commands in the Process Option field:
   - Enter PD to print the data set and then delete it.
   - Enter D to delete the data set.
   - Enter K to keep the data set and allocate the same data set in the next GIX session.
   - Enter KN to keep the data set and allocate a new data set in the next GIX session.
5. Press Enter to exit completely from GIX, by terminating ISPF.
Exiting from GIX

A message will be displayed, depending on which process option you selected. The system returns you to the TSO Ready prompt.

6. If you want to log off from TSO, type `logoff` at this prompt.

**The GIX Commands Log**

When you exit from GIX, NetView DM for MVS logs all GIX commands that changed the status of the TCF in a file called the *GIX commands log*. These commands are as follows:

- Submit a plan
- Delete a plan
- Delete a phase from a plan

NetView DM for MVS also logs the following:

- Any error conditions that caused a GIX abnormal end (abend)
- The name of the data set used to store the dump for a GIX abend

A separate log is produced for each GIX user, and is stored in the DSXLOG data set. The default for this data set is SYSOUT (A). Refer to the *Installation and Customization* book for more information.
Exiting from GIX

Figure 5 shows an example of the contents of a GIX commands log.

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/15</th>
<th>TIME: 11:57</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB003I</td>
<td>REQUEST SUBMIT PLAN, ITEM MYPLAN, USER NDM37,</td>
<td>DATE 89/05/15, TIME 11:57:35.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/15</th>
<th>TIME: 13:01</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB003I</td>
<td>AN UNRECOVERABLE ERROR WAS ENCOUNTERED WHILE PROCESSING TCF SERVICE, UPDATE FUNCTION. MODULE=DSXG8003,</td>
<td>RETURN CODE=16, CONDITION CODE=12. ABEND 0023.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/15</th>
<th>TIME: 13:02</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB043I</td>
<td>THE SNAP HAS BEEN EXECUTED ON DATA SET 'DSXC.SNAP1'.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/16</th>
<th>TIME: 10:54:56.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSX8003I</td>
<td>REQUEST SUBMIT PLAN, ITEM JGFDANO00, USER NDM37,</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/17</th>
<th>TIME: 15:05</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB003I</td>
<td>REQUEST DELETE RECURSION, ITEM ERCOLE30 87/05/10,</td>
<td>USER NDM37, DATE 89/ 5/17, TIME 15:05:07.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/18</th>
<th>TIME: 12:59:19.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB003I</td>
<td>REQUEST DELETE PLAN, ITEM CLIST001, USER NDM37,</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/18</th>
<th>TIME: 15:33</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB003I</td>
<td>REQUEST DELETE PHASE, ITEM PHASE003, USER NDM37,</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/28</th>
<th>TIME: 12:46</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDMB30I</td>
<td>AN UNRECOVERABLE ERROR WAS ENCOUNTERED WHILE PROCESSING REQUEST QUEUE SERVICE, CONNECT FUNCTION. MODULE=DSXG8000,</td>
<td>RETURN CODE=12, CONDITION CODE=0. ABEND 22.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>USER ID: NDM37</th>
<th>DATE: 89/05/28</th>
<th>TIME: 12:47</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSX8035I</td>
<td>ALL SNAP DATASETS ARE FULL.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Sample output of a GIX commands log
Chapter 2. Defining User Profiles

Before you can use NetView DM for MVS, the system administrator has to perform a series of tasks to set up the NetView DM for MVS environment. Defining user profiles is the first step in the following sequence of tasks after NetView DM for MVS has been installed:

**Step 1. Define profiles for all NetView DM for MVS users**

**Step 2. Define each node**

**Step 3. Define each resource**

**Step 4. Assign resources to nodes**

Step 1 is highlighted because it is described in this chapter. The remaining steps are described in the chapters that follow.

This chapter explains what a user profile is, and how to create a new user profile using GIX. Once you have created profiles using GIX, you can also:

- Change an existing profile
- Delete a profile

What Is a User Profile?

A *user profile* is a file which contains a description of each user of NetView DM for MVS. Each user must be assigned a user profile before that user can access either GIX, Batch Utilities, or IOF. The profile describes the following:

- Who the user is, in terms of a *user identification* and password. The user identification corresponds to the TSO user ID; the password is specific to the NetView DM for MVS environment.
- Which GIX tasks the user is authorized to perform. The different functions are shown in Figure 7 on page 21 and described after the figure.
- Which nodes, resources, and functions the user is authorized to access.
- The default values that will appear on GIX panels, customized for the user.

If you are the system administrator using NetView DM for MVS for the first time after installation, you must create a user profile for each user before that user can access NetView DM for MVS. Your users will probably need different authorizations, depending on their role in your organization and the tasks they will have to perform. The user profiles are stored in the distributed resource directory (DRD).

**Note:** If you are a new user, but not the first person to use NetView DM for MVS, contact your system administrator, or whoever is authorized to define new user profiles.

The following pages show you how to create a new user profile, and describe the options that you can define.
Defining a User Profile Using GIX (System Administrator)

Defining a User Profile Using GIX

Use the instructions on the following pages at your terminal. For each panel, there is a set of instructions, followed by a description of the data you can enter on each panel. Begin by logging on to the GIX main menu, and follow these instructions:

**Step 1: Define the User Identification**
1. Select **Option 8, Set Profiles**, from the GIX main menu.

   The Set Profiles panel is displayed.

   ![Set Profiles Panel](Figure 6. Example of a Set Profiles panel)

2. Type the name of the user ID for which you are creating a user profile in the **User ID** field.

3. You can optionally type in the name of a user profile model in the **User profile model** field, to use as a model for the one you are creating.

4. Select **Option 1, Insert**, and press Enter.

   The Change Password and GIX Authorizations panel is displayed. This panel is shown in Figure 7 on page 21.

The fields on the Set Profiles panel are described below.

**User ID:** This is the name of the TSO user ID, which must be entered by GIX users when they log on to GIX, batch users when they create job files, and by IOF users when they log on to the IOF. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore). The first character must be alphabetic or special.

**User profile model:** This is the name of an existing user profile. The system displays the attributes defined for this profile, and you can overwrite them for the new profile.

---

NDMPS1AE SET PROFILES

Selection ==> 15:52

Enter:

User ID . . . . . TMSLEW Blank or partial name followed by * for full-list
User profile model MODEL Used for INSERT option only

Select one of the following:

1 INSERT Add a new user ID
2 CHANGE Change user authorizations and defaults
3 DELETE Delete a user ID

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 1/zerodot=PRINT 12=CURSOR

---

---
Step 2: Change Password and GIX Authorizations

NDMPS5AE CHANGE PASSWORD AND GIX AUTHORIZATIONS (TMSLEW)
Command =>

Change password, if applicable:

1 Password SAMPLE Applies to GIX, BU and IOF

Change the following NETVIEW DM FUNCTION AUTHORIZATIONS, as appropriate:

2 User profile . . . . . . . . . . . . . . 1 All fields: (1=Yes 2=No)
3 Plan preparation . . . . . . . . . . . 1
4 Submit and manage submitted plans . . . . . . 1
5 Network configuration update . . . . . . 1
6 Maintain repository and plan library . . . . 1
7 Resource repository deletion and update . . . 1
8 Change node profile authorizations . . . . 1
9 IOF . . . . . . . . . . . . . . . . . . . 1
10 Delete phase from IOF . . . . . . . . . . 1
11 Manage other users' plans . . . . . . . . 1

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR

Figure 7. Example of a Change Password and GIX Authorizations panel

1. Type the password in the Password field.

2. To change the default values that are currently displayed in fields 2 through 10 for NetView DM for MVS function authorizations, type either 1 or 2: 1 means Yes, and grants authorization to the user to perform that task; 2 means No.

3. Press Enter when you have made all the required changes.

   The Change Node Type Authorizations panel is displayed. This panel is shown in Figure 8 on page 24.

The fields on the Change Password and GIX Authorizations panel are described below:

**Password:** This is an optional password which must be entered by GIX and IOF users when they log on, batch users when they submit jobs, and D&CC API users. Even if this password is specified, the batch user does not have to code a password in the job file to be submitted, if the operand PSWD of the NDMBATCH installation macro has been set to OPT (optional). Refer to “How to Run the Batch Utilities” on page 202 for information about how to run the Batch Utilities.

You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore). The first character must be alphabetic or special.

**User profile:** This authorization allows the user to perform the following tasks:

- Create, change, print or delete a user profile
- Delete a group of nodes or resources created by another user
- Delete a plan, created by another user, from the plan library
These are tasks usually performed by the system administrator.

**Plan preparation:** This authorization allows the user to perform the following tasks:
- Create or change a plan in the plan library
- Validate a plan
- Browse or print any plan in the plan library
- Rename or delete a plan owned by the user from the plan library
- Query space left in the plan library

**Submit and manage submitted plans:** This authorization allows the user to perform the following tasks:
- Submit a plan to the transmission control file (TCF)
- Delete or erase a plan, plan recursion, or phase from the TCF

**Network configuration update:** This authorization allows the user to perform the following tasks:
- Define, change and delete node definitions
- Define, change and delete resource definitions
- Assign resources to nodes
- Define and delete groups of nodes and resources
- Define, change and delete tracking information related to nodes and resources

These are tasks usually performed by the system administrator.

**Maintain repository and plan library:** This authorization allows the user to perform the following tasks:
- Back up and restore the resource repository
- Unlock resources in the resource repository previously locked by an application program accessing the D&CC API
- Maintain the plan library

These are tasks usually performed by the system administrator.

**Resource repository deletion and update:** This authorization allows the user to perform the following task:
- Manage resources in the resource repository

**Change node profile authorizations:** This authorization allows the user to perform the following task:
- Change node authorization profiles. Node authorization profiles are described in “Node Types” on page 33.

**IOF:** This authorization allows the user to perform the following task:
- Log on to the interactive operator facility (IOF). You can authorize more than one person to use IOF, but only one person can log on to a transmission control program (TCP) at a time. Refer to Part 3, “Using the Interactive Operator Facility (IOF),” for more information.
**Delete phase from IOF:**  This authorization allows the user to perform the following task:

  Delete specific phases of transmission plans. The IOF authorization is also required. Refer to Part 3, “Using the Interactive Operator Facility (IOF),” for more information.

**Manage other users’ plans:**  This authorization allows a user who is not the owner of the plan, and who is not the system administrator, to perform the following tasks:

  - Delete, change, and rename the plan in the plan library using GIX
  - Delete the plan in the plan library using Batch Utilities

The user must also be authorized for the Plan Preparation option in order to use the Manage other users’ plans option.
Step 3: Change Node Type Authorizations

Figure 8. Example of a Change Node Type Authorizations panel

1. For each option you want to change, type 1 or 2 over the value currently displayed: 1 means Yes, and grants the user authority to use the node type; 2 means No.

2. Type SAVE on the command line when you have made all your changes, and press Enter.

   Note: Remember that PF12 or PF24 moves the cursor between the data entry fields and the command line.

The Change Resource Authorizations panel is displayed. This panel is shown in Figure 9 on page 25.

The data you can enter on the Change Node Type Authorizations panel is described below.

Node type: These are the types of node the user will be allowed to work with. Table 70 on page 561 and Table 71 on page 562 show the node types that you can define.
Step 4: Change Resource Type Authorizations

NDMPS8AE CHANGE RESOURCE AUTHORIZATIONS (TMSLEW)

Command ==> Scroll ==> PAGE

Primary command: SAVE

Line commands: 1 = Yes  2 = No

CMD RESOURCE TYPE
2 MICROCODE
2 SOFTWARE
2 PROCEDURE
2 FLAT DATA
2 RELATIONAL DATA
2 AS/400 OBJECT
2 AS/400 CTN
2 CONFIGURATION FILE
2 MSDUMP
2 TRACE INFO
2 MSERRLOG

Figure 9. Example of a Change Resource Type Authorizations panel

1. For each option you want to change, type 1 or 2 over the value currently displayed: 1 means Yes, and grants the user authority to use the resource type; 2 means No.

   **Note:** Use the Scroll field to display resources that are not currently displayed.

2. Type SAVE on the command line when you have made all your changes, and press Enter.

   The Specify Node Class Authorizations of User panel is displayed, shown in Figure 10 on page 26.

The data you can enter on the Change Resource Authorizations panel is described below.

**Resource type:** These are the types of resource the user will be allowed to work with. Table 3 on page 63 shows the resource types that you can define.
Step 5: Specify Node Class Authorizations of User

Figure 10. Example of a Specify Node Class Authorizations Of User panel

1. Type the node classes that the user can work with in the fields provided.

2. Press Enter when you have finished.

The Specify Resource Class Authorizations of User panel is displayed. This panel is shown in Figure 11 on page 27.

The data you can enter on the Specify Node Class Authorizations of User panel is described below.

Node class: The node class is a way of authorizing only certain users to work with specific nodes. When you define the nodes in the network, you will assign a node class authorization to each of them. A node class consists of two alphanumeric characters. The first character must be alphabetic. You can specify up to 16 different node classes. All users are authorized to use class A0 nodes.
Step 6: Specify Resource Class Authorizations of User

Figure 11. Example of a Specify Resource Class Authorizations Of User panel

1. Type the resource classes that the user can work with in the fields provided.
2. Press Enter when you have finished.

The Specify User Defaults panel is displayed. This panel is shown in Figure 12 on page 28.

The data you can enter on the Specify Resource Class Authorizations of User panel is described below.

**Resource class:** The resource class is a way of authorizing only certain users to work with specific resources. When you define a resource, you assign it a resource class authorization. A resource class consists of two alphanumeric characters. The first character must be alphabetic. You can specify up to 16 different resource classes. All users are authorized to use class A0 resources.
Step 7: Specify User Defaults

Figure 12. Example of a Specify User Defaults panel

1. Type the values for each default. If values are already displayed, you can type over them. These default values customize the user profile so that only the node or resource classes that the user is allowed to work with are displayed on the user’s panels. They also define other user defaults, described below.

2. Press Enter when you have finished. The new user profile is saved in the DRD, and GIX returns you to the Set Profiles. You can then define another profile, or press PF4 to return to the GIX main menu.

The fields on the Specify User Defaults panel are described below.

**Node class (default):** Specify the default value that will appear on the Specify Node Attributes panel when the user creates node definitions. Refer to “Node class” on page 26.

**Resource class (default):** Specify the default value that will appear on the Specify Attributes for node type/resource type panel when the user creates resource definitions. Refer to “Resource class” on page 27.

**Lines per page:** Specify the number of lines per page in printed reports. Enter a number from 21 to 99. The default value is 55.

**Block size:** Specify the record block size of the data set used by GIX print jobs. Enter a number of bytes from 133 to 32718. The number must be a multiple of 133.

**Page allocation:** Specify the maximum number of pages that the user can print in reports. Enter two values. These values are converted automatically (according to the output device type) to the required primary and secondary space allocations.
Changing a User Profile (System Administrator)

**Sysout class:** Specify the default SYSOUT class of Batch Utility jobs submitted through GIX.

**Plan model name:** Specify the name of the default transmission plan contained in the plan library. It can be from 1 to 8 characters long. The first character must be alphabetic. This is the name that will appear in the Plan model field of the panels used to prepare plans, whenever the user creates a new plan.

**Attention**

The model that is shipped with the NetView DM for MVS software refers to example nodes and resources that have not been defined in the DRD. Do not specify this model as the default plan for users.

**Printer name:** Specify the local printer name associated with the user. You can only specify a local printer when the JES/328X Print Facility is available. The name must match the one defined in the PRINTER parameter of the NDMGIXnn generation macro. Refer to the Installation and Customization book for information on this macro.

**User device type:** Specify the user's device type. Enter 1 if the user's terminal and printer use Roman characters. Enter 2 if the user's terminal and printer use Katakana characters. The default value is 1 (Roman).

**JCL cards:** Specify up to four job control language (JCL) cards containing job statement information. These statements are used when submitting a batch job through GIX. The user submits a batch job to load and unload resources from the resource repository. This is described in Chapter 6, “Preparing Resources for Transmission” on page 97. The first card you specify must be a job statement card. The other cards can be data definition (DD) statement cards, comment cards, or you can choose not to use them.

### Changing a User Profile

You can change an existing user profile. Follow the instructions given below:

1. From the Set Profiles panel (shown in Figure 6 on page 20), specify the name of the user profile you want to change. There are two ways in which you can do this:

   - If you know the name of the profile, type the user ID of the profile you want to change in the **User ID** field. Do not type anything in the **User profile model** field. Select Option 2, Change, and press Enter.

   - If you do not know the name of the profile, select the profile from a full-list panel. Do not enter anything on the Set Profiles panel, and press Enter. This displays the List of NetView DM for MVS Users full-list panel. Use the S (Select) line command to select the profile that you want to change, and press Enter.

   The Authorizations and Defaults of User panel is displayed.
Deleting a User Profile (System Administrator)

2. Type a number on the selection line, and press Enter. This takes you to one of the panels described in “Defining a User Profile Using GIX” on page 20. The numbers you can enter are:
   1. To change the password or NetView DM for MVS functions
   2. To change node type authorizations
   3. To change resource type authorizations
   4. To change node class authorizations
   5. To change resource class authorizations
   6. To change user defaults

3. Change the fields on these panels, as described in “Defining a User Profile Using GIX” on page 20. Press Enter to save any changes you make.

Deleting a User Profile

Follow the instructions below to delete an existing user profile from the DRD.

**Attention**

If you are the system administrator, do not delete your own user profile if it is the only profile remaining in the DRD. There must always be at least one user profile defined.

1. From the Set Profiles panel (shown in Figure 6 on page 20), specify the name of the user profile you want to delete. There are two ways in which you can do this:
   - If you know the name of the profile, type the user ID of the profile you want to delete in the User ID field. Do not type anything in the User profile model field. Select Option 3, Delete, and press Enter.
   - If you do not know the name of the profile, select the profile from a full-list panel. Do not enter anything on this panel, and press Enter. This displays the List of NetView DM for MVS Users full-list panel. Use the D (Delete) command to select the profile that you want to delete, and press Enter.

The Confirm the Deletion of a User panel is displayed, shown in Figure 13 on page 31.
Deleting a User Profile (System Administrator)

2. Type the primary command **D** (Delete) on the command line, and press Enter to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel.
Chapter 3. Defining Nodes

Defining nodes is the second step in the following sequence of tasks after NetView DM for MVS has been installed:

Step 1. Define profiles for all NetView DM for MVS users
Step 2. Define each node
Step 3. Define each resource
Step 4. Assign resources to nodes

This chapter explains how to create a node definition using GIX. Once you have created definitions using GIX, you can also:

- Change a node definition
- Delete a node definition
- Browse the contents of a single node definition, or all node definitions
- Print the contents of a single node definition, or all node definitions

This chapter also explains how to work with transmission profiles, connection profiles, and groups of nodes, using GIX.

What Is a Node Definition?

A node definition describes the characteristics of a node to NetView DM for MVS. You must define each node in your network to NetView DM for MVS before end users can refer to them in transmission plans.

For each node you must define:

- A symbolic name which identifies the node
- The type of node
- A set of node attributes

The following pages explain the node types you can define.

Node Types

NetView DM for MVS can use a variety of different types of node. The node type identifies which hardware and software the node comprises. The types of node that your host is able to access are defined at installation time, as described in the Installation and Customization book, which also describes the hardware, operating system, and other software required at the nodes.

Nine types of node have names predefined for NetView DM for MVS. These nodes use LU0 protocols for communications connections. The node type determines which resources and functions you can use.

There are also node types which do not have predefined names. Each of these node types has a user-defined name. These nodes use LU 6.2 protocols for communications connections. Again, the node type determines which resources and functions you can use.
Nodes can be connected either directly to the host, or through an intermediate node. The intermediate node queues resources and distribution requests from the host and sends them to the nodes. It also queues resources and requests received from the nodes and forwards them to the host. The end point nodes are called entry points. The host, where NetView DM for MVS usually runs, is called the focal point.

**LU0 Node Types**

Table 70 on page 561 contains a complete list of the LU0 node types you can define.

When you define one of these node types, you are also defining its capabilities in relation to the functions and resources it can use. The node types differ as to the functions and resources they can handle. When you are using GIX, the panels automatically display the functions and resources that you can use with the node type you are working with.

**LU 6.2 Node Types (User-defined)**

Table 71 on page 562 contains a complete list of the user-defined LU 6.2 node types you can define.

The functional capabilities of these nodes are defined at installation time in the following ways:

**CMEP  Change Management Entry Point**

Nodes which receive and implement changes that are managed and distributed by a focal point are called entry points. An entry point is any distributed processor or subsystem whose software, microcode, user data, updates for each of them, and maintenance information are controlled and tracked by a focal point. An entry point can also initiate distribution, retrieval, and deletion requests to other entry points and to the focal point, but only a focal point can initiate remote change management functions. An entry point is also referred to simply as a node throughout this book.

Each CMEP node type has a node authorization profile. This profile defines the functions that the CMEP node can request in a node solicited request (NSR), the resource types that the node can work with, and the classes of resource. NetView DM for MVS provides defaults for these three parameters, but they can also be changed at installation time in the NDMNODE generation macro. You can also change these parameters using GIX, when you define each node to NetView DM for MVS. Refer to “Step 4 (CMEP nodes only): Specify Node Authorizations” on page 41.

**CMFP  Change Management Focal Point**

You can define a node as a CMFP node, which means that it has the functional capability to administer changes to other nodes. A focal point is any host which carries out centrally controlled resource distribution, remote change management, and tracking. For example, you could define any host System/390 where NetView DM for MVS is running and controlling the retrieval, distribution, and remote installation of resources.
to and from distributed processors or subsystems. You can define several nodes in the same NetView DM for MVS network as CMFP nodes.

**Note:** GIX panels automatically display the resource types and functions that you can use with CMFP node types. CMFP node types support only LU 6.2 resource types, and not LU0 resource types.

**NDMT**  
**NetView DM for MVS Transfer**

You can define another NetView DM for MVS node in the network to your DRD (at the host) as an NDMT node. NDMT is the functional capability of one NetView DM for MVS focal point to exchange data objects from its resource repository with another NetView DM for MVS focal point resource repository.

NDMT nodes support only LU0 resource types.

You can define the functional capabilities of a single LU 6.2 (user-defined) node in any of the following ways:

- CMEP capabilities only
- CMFP capabilities only
- NDMT capabilities only
- Both NDMT and CMFP capabilities

---

**Defining Nodes Using GIX**

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

**Step 1: Specify the Node Type**

1. Select *Option 1, Configure Network*, from the GIX main menu.

   The Configure Network panel is displayed, as illustrated in Figure 14.

   ![Configure Network Panel](image)

   **Figure 14. Example of a Configure Network panel**
2. Type the node type in the **Node Type** field, selecting one of the options displayed, or accept the value shown.

3. Select **Option 4, Define Nodes**.

   The Specify Operation on Node panel is displayed.

The data you can enter on the Configure Network panel is described in the following section.

**Node type:** This defines the type of node. Table 70 on page 561 and Table 71 on page 562 show the possible node types you can define, depending on your installation.

**Step 2: Specify Operation on Node**

1. Type a symbolic node name in the **Node name** field.

2. You can optionally enter the name of a node model in the **Node model** field, to use an existing node definition as a model for the one you are creating.

3. Select **Option 1, Create**, to create a new node definition.

   The Specify Node Attributes panel is displayed, illustrated in Figure 15 on page 37.

The fields on the Specify Operation on Node panel are described in the following section.

**Node name:** This is a symbolic name which identifies the node to your end users. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, or _ (underscore). The first character must be alphabetic or special, but do not use the _ (underscore) character for the first or last character.

---

**SSP and PDOS node types**

When you define the type of connection between the focal point and the entry point for SSP and PDOS node types, described in “Step 3: Specify Node Attributes” on page 37, the value you enter can affect the node name that you enter here. For SSP and PDOS node types, the connection can be intermediate, rather than direct. In these cases, if you specify intermediate, the node name you specify here must be the node name which is already defined in the communication routes definition for the front-end SSP node.

**Node model:** This is the name of an existing node definition. The system displays the attributes defined for this definition, and you can overwrite them for the new definition.
Step 3: Specify Node Attributes

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node class</td>
<td>A0</td>
<td>Required</td>
</tr>
<tr>
<td>Status</td>
<td>2</td>
<td>1 = Production, 2 = Parallel, 3 = Test</td>
</tr>
<tr>
<td>Logical unit</td>
<td>AAA</td>
<td>Required (Logical unit name)</td>
</tr>
<tr>
<td>Logon mode</td>
<td>M3174A</td>
<td>Logon mode name</td>
</tr>
<tr>
<td>Linetype</td>
<td>1</td>
<td>1 = Leased, 2 = Switched</td>
</tr>
<tr>
<td>Rgn.</td>
<td></td>
<td>Network identification</td>
</tr>
<tr>
<td>Ren.</td>
<td></td>
<td>CP Logical unit name</td>
</tr>
<tr>
<td>Notes</td>
<td>2</td>
<td>Enter 1 if you want additional node information</td>
</tr>
<tr>
<td>Profile</td>
<td>1</td>
<td>Enter 1 if you want to change node profile</td>
</tr>
<tr>
<td>Server Name</td>
<td>MXWPO</td>
<td>Server name</td>
</tr>
<tr>
<td>Timezone</td>
<td>+00</td>
<td>Time Zone offset. Any value from -12 to +12</td>
</tr>
</tbody>
</table>

Tracking information:
- Activity: CREATE
- Node: CMEPnode

Figure 15. Example of a Specify Node Attributes panel. This example is for a CMEP node type. The fields shown on this panel may vary according to the node type.

1. Type in the values for each attribute.

2. If you want to enter additional information about the node, type 1 in the **Notes** field. This displays the Enter Additional Node Information panel. The default value for this field is 2 (No), which means that you do not want to enter additional information.

3. If you want to change the node profile, type 1 in the **Profile** field. This displays the Specify Node Authorizations panel, illustrated in Figure 16 on page 41. The default value for this field is 2 (No), which means that you do not want to change the profile.

4. Press Enter when you have finished. The new node definition is saved in the DRD, and GIX returns you to the Specify Operation on Node panel. You can then define another node, or press PF4 to return to the GIX main menu.

The change has immediate effect on transmission activities, if GIX is running with NETCHNG=IMMEDIATE in the GIX profile and the TCP is running with NETCHNG=IMMEDIATE defined either in the TCP profile or as a TCP startup parameter. The specified node type must already have been defined when the TCP was started.

The value of the NETCHNG TCP startup parameter overrides the value you defined at customization time in the TCP profile (refer to “Specifying the Effect of Network Changes (NETCHNG=)” on page 481).

If the TCP is not running, or is running with NETCHNG=DELAYED, then the change will not come into effect until the next time the TCP is cold-started.
The Specify Node Attributes panel may display any of the following fields depending on the node type.

**Connection:** SSP and PDOS node types only. This attribute describes how the entry point node is linked to the host. You can either enter 1, which means *intermediate*, or 2, which means *direct*. You can only specify a value of intermediate for PDOS or SSP nodes that are connected to the host via an SSP intermediate node. You must specify a value of intermediate for AS/400 nodes, even if the node is directly connected to the host, unless the AS/400 node runs with OS/400 Version 1 Release 3 or later.

The default value for this field is 2 (direct). Specify a value of direct for any node that is in direct session with the host.

**Directory name:** PDOS node types only. This attribute only applies to nodes in a multi-tiered network. If you specify a connection type of intermediate, you must also enter a directory name which specifies the name of the intermediate node to which the node is connected. You can enter from 1 to 8 alphanumeric characters, or the special characters $, #, _, %, &, and @. The directory name must match the system name defined in the communication routes definition of the intermediate node that connects the entry point node to the host.

**Linetype:** This specifies the type of line connecting the entry point node to the host. The type of connection must match the one defined in the VTAM/TCAM tables for the logical unit (LU) associated with the node. You can enter one of the following values:

- 1 if the node is connected via a leased line
- 2 if the node is connected via a switched line

The default value is 1 (leased).

**Logical unit:** This field is mandatory. You must specify the logical unit (LU) name that was assigned to the node when NetView DM for MVS was installed. This is the name defined in the VTAM/TCAM tables for the node (see the *Installation and Customization* book for information). If you are distributing resources by intermediate nodes, then the LU name must be the name of the next intermediate node.

Nodes of the same type can share the same LU name, if they share the same type of connection, described above. This does not apply to user-defined type nodes with CMEP, CMFP, or NDMT functional capabilities.

Nodes of different types can share the same logical unit name when PDOS or SSP type nodes are connected to the host through a front-end system (for example, a System/36 running under SSP, or an AS/400 system running under OS/400). The name you specify must match the LU name of the front-end SSP.

**Logon ID and password:** This is the node logon ID and password. The node logon ID authorizes the host to establish operations with DPPX, and SSP node types. You must specify the logon ID and password that were defined at the node when NetView DM for MVS was installed (see the *Installation and Customization* book for information).
Logon ID  This is mandatory for DPPX node types, and optional for SSP node types. You can enter up to eight characters (alphabetic, or the special characters $, %, &, #, @, or _).

Password  This field is optional. The characters you can enter for the password are from the same set as for the logon ID. You can enter the following values:

- Up to 8 characters for SSP nodes that are AS/400 systems, but only up to 4 characters for SSP nodes that are System/36s
- Up to six characters for DPPX nodes

Logon mode:  This field is optional. This is the logon mode name assigned to the node. Several nodes can share the same logon mode name. You should define the same logon mode name for nodes which have a connection value of intermediate, and which share the same LU name. Defining the logon mode will optimize line performance.

If you specify a logon mode name, it must match the LOGMOD name defined in the VTAM/TCAM tables for the node. If you do not specify a name, the default is the value defined in the LOGM parameter of the NDMNODE generation macro. If NetView DM for MVS cannot find a value, it uses the LOGMOD name defined in the VTAM/TCAM tables for the node.

Node class:  This is the authorization class for the node. The node class is a way of authorizing only certain users to work with specific nodes. A node class consists of two alphanumeric characters. The first character must be alphabetic, for example A1. You can specify up to 16 different node classes. All users are authorized to use class A0 nodes. Each user profile specifies which node classes the user can access.

Notes:  This option allows you to enter any additional information about the node. If you enter 1, a panel is displayed. You can enter up to nine lines of 71 characters of data on this panel. The default value is 2 (No), if you do not want to enter additional information.

Profile:  This field is displayed for CMEP user-defined node types only, and only if you are authorized to change node profile authorizations. This option allows you to choose whether you want to edit the node profile for a CMEP node. When NetView DM for MVS receives a node solicited request (NSR) from a CMEP node, it refers to the node profile to carry out the following checks:

- That the node is authorized to the function being requested by the NSR
- That the node is authorized to the resource type referred to by the NSR
- That the resource referred to by the NSR has a class that matches one of the classes to which the node is authorized

The node profile contains information about the resource classes, resource types, and functions that the node is authorized to use. The default values for the node profile authorizations are defined in the NDMNODE generation macro at installation time. You can edit this profile, by typing 1 in the Profile field, provided that you are authorized to change node profiles (defined in your user profile).
**RGN and REN:** These two fields are displayed for CMEP and CMFP user-defined node types only.

RGN is a SNA/DS term that means *routing group name*. REN is a SNA/DS term that means *routing element name*. Together these two fields specify the distribution services unit (DSU) identifier of the target node. You must specify the RGN and REN values that are defined at the node.

When defining NetView DM/6000 nodes, you must specify the RGN and REN values as follows:

- For a NetView DM/6000 node that is directly connected to the NetView DM for MVS host, the RGN and REN values are the same. The RGN must be the same as the node’s REN (or *short name*, as it is referred to in NetView DM/6000).
- For NetView DM/6000 nodes that are indirectly connected to the host as CC clients, RGN corresponds to the REN defined for the CC server to which the CC client is connected locally. REN corresponds to the REN defined for the CC client when it was configured.
- If the DRD contains node definitions for NetView DM/6000 nodes with NetView DM/6000 Release 1.0 installed and the CC server is upgraded to NetView DM/6000 Release 1.1, you must change the definitions for the CC server and all of the CC clients controlled by the server.

You can enter up to eight characters for each field. Each character must belong to the subset of character set 1134, which is defined in “Character Sets” on page 585.

If you do not specify the target node, the default is the name of the next intermediate node.

**Server name:** This field is optional and specifies the node name of the server node for the node you are defining here. This field is valid only for nodes with CMEP functional capability.

*Note:* The *Server Name* field always initially shows the default value, which is the node name.

**Status:** This field is optional and defines the node's operational status. It is used for information only. The values can be:

- Production, which means that no testing is done at the node
- Parallel, which means that production work and testing are both done at the node
- Test, which means that no production work is done at the node

The default value is 2 (Parallel).

**Timzoffs:** This field is optional and is the time-zone offset field. Use this field to specify the offset between the time zones where the host and the node are located. This allows polling to begin at the right time for the INSTALL RESOURCE, ACTIVATE NODE and INITIATE PROCEDURE functions, because the execution date and time are then expressed in local time.
The value you specify must be a whole number, ranging from −12 to +12 (the + sign can be omitted). The default value is 0 (that is, the host and the node are in the same time zone).

**Note:** The time zone offset function only applies to nodes with CMEP and CMFP functional capabilities.

**Step 4 (CMEP nodes only): Specify Node Authorizations**

The Specify Node Authorization panel is only displayed if you entered a 1 in the **Profile** field of the Specify Node Attributes panel for a CMEP user-defined node type.

You use the Specify Node Authorization panel to specify which classes of resource, which functions, and which resource types the CMEP node is authorized to use. The panel displays the default values that were defined in the NDMNODE generation macro at installation time. In this example, the installation macro is coded as follows:

```plaintext
NDMNODE TYPE=LU6NOD,FUNC=CMEP,XMFUNC=<SEND,RETR,>,
RESTYPE=(FLATD,SOFTW,MICR),CLASS=(A1,A2,A6,A9)
```

The default values displayed on this panel are taken from the NDMNODE macro defined at installation time for the node type. The panel displays only the resource types that you are authorized to use.

**Note:** Unless you are a system administrator, you can only change the values for those functions, resource types, and resource classes that were set when the node was defined at installation time. You cannot add more functions, resource types, or classes.

System administrators, however, can also enable a node to send NSRs involving more functions, resource types, and classes than those defined at installation time.
1. You can add or change the resource classes that the node can work with in the **Resource Classes** field. For example, add an A3 resource class. You can also delete resource classes, if you do not want to authorize the node to use the resource class.

2. Enter the line command 1 in the **CMD** field beside any other resource types that you want to authorize the node to use. For example, you could add the resource type called configuration file to this profile. This shows that the node is authorized to send node-solicited requests (NSRs) referring to this resource type.

   You can also enter 2, which means that you do not want to authorize the node to use the resource type.

3. Enter 1 in the **CMD** field beside each of the functions that the node is authorized to perform. For example, enter 1 beside the DELETE function to authorize the node to use the function.

   You can also enter 2, which means that you do not want to authorize the node to use the function.

4. When you have finished, press Enter to save the node authorizations in the DRD. If you do not want to save the modified values, press PF3 to return to the previous panel. In this example, the node called CMEPNODE is inserted in the DRD with a node profile of:

   Authorized functions : SEND, RETRIEVE, DELETE
   Authorized resources : FLAT DATA, SOFTWARE, MICROCODE, CONFIGURATION FILE
   Resource classes : A1, A2, A3, A6, A9

   GIX returns you to the Specify Operation on Node panel. You can then define another node, or press PF4 to return to the GIX main menu.

The data you can enter on the Specify Node Authorizations panel is described in the following section.

**Resource classes:** Specify the resource classes that the node is authorized to use. You can specify up to 14 classes. Each class consists of two alphanumeric characters, the first character must be alphabetic, for example, A1. This value is defined in the CLASS parameter of the NDMNODE generation macro. If no value was specified in this parameter, the default is A0.

**Resource type:** Specify the LU 6.2 resource types that the node is authorized to use. If you specify functions, you must also specify resource types here: 1 means that the node is authorized to send NSRs referring to that resource type; 2 means that the node is not authorized. The default values are defined in the RESTYPE parameter of the NDMNODE generation macro.

**Function:** Specify the transmission functions that the node is authorized to request: 1 means authorized; 2 means not authorized. The functions you can authorize are:

- **Send**
  This means that the node can send a resource to NetView DM for MVS.

- **Retrieve**
  This means that the node can request NetView DM for MVS to send a resource that is stored in the resource repository.

- **Delete**
  This means that the node can request NetView DM for MVS to delete a resource that is stored in the resource repository.
Changing a Node Definition

You can change the attributes of a node that is defined in the DRD. Follow these steps:

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node you want to change, in the **Node type** field, or accept the one shown.

2. Select **Option 4, Define Nodes**, and press Enter.

   The *Specify Operation on Node* panel is displayed.

3. Specify the name of the node you want to change. There are two ways in which you can do this:
   
   a. If you know the name of the node, type the node name in the **Node name** field. Do not type anything in the **Node model** field. Select **Option 2, Change**, and press Enter.

   b. If you do not know the name of the node, select the node from a full-list panel. Leave the **Node name** field empty, or type in a partial name followed by *.* Leave the selection line blank, and press Enter. This displays the List of Defined Nodes full-list panel. Use the **S** (Select) command to select the node definition you want to change, and press Enter.

   The **Specify Node Attributes** panel is displayed. This panel is shown in Figure 15 on page 37.

4. Change the values on this panel, as required. The fields are described in “Step 3: Specify Node Attributes” on page 37. Press Enter to save any changes you make. GIX returns you to the previous panel.

The change has an immediate effect on transmission activities, if GIX is running with NETCHNG=IMMEDIATE in the GIX profile *and* the TCP is running with NETCHNG=IMMEDIATE defined either in the TCP profile or as a TCP startup parameter. The change becomes effective:

- *Immediately*, if there is no active SNA session with the logical unit associated with this node
- *At the end of the session*, if an SNA session is active.

If, however, you change:

- The associated logical unit
- The connection type
- The RGN or REN definitions

the change always becomes effective *immediately* and all transmission activities involving this node are canceled. If you change the node class only, the change becomes effective *immediately*, but only the validation of requests for transmission activities coming from the D&CC API are affected.

When you change the logon mode value for an LU 6.2 node that is directly connected to the host (the REN value of the node is equal to the logical unit),
Deleting a Node Definition (System Administrator)

NetView DM for MVS automatically changes the logon mode to this value for all of the nodes that share this logical unit.

Changing the logon mode value for an LU 6.2 end node, however, does not affect the logon mode values of other nodes that share the same logical unit. Only changes to the logon mode of a node that is directly connected to the host, and that shares the same logical unit as the end node, affect transmission.

If the TCP is not running, or is running with NETCHNG=DELAYED, the change will not come into effect until the next time the TCP is cold-started.

Deleting a Node Definition

To delete an existing node definition from the DRD, follow these steps:

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node you want to delete, in the Node type field, or accept the one shown.
2. Select Option 4, Define Nodes, and press Enter.
   The Specify Operation on Node panel is displayed.
3. Specify the name of the node that you want to delete. There are two ways in which you can do this:
   a. If you know the name of the node, type the node name in the Node name field. Do not type anything in the Node model field. Select Option 3, Delete, and press Enter.
   b. If you do not know the name of the node, select the node from a full-list panel. Leave the Node name field empty, or type in a partial name followed by *. Leave the selection line blank, and press Enter. This displays the List of Defined Nodes full-list panel. Use the D (Delete) command to select the node definition you want to delete, and press Enter.

The Confirm the Deletion of Node panel is displayed.

4. Type the primary command D (Delete) on the command line, and press Enter to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel.

Note: If you delete a node that is the last remaining node associated with a logical unit, NetView DM for MVS also removes the LU from the transmission profile to which it was assigned. Transmission profiles are described in “Using Transmission Profiles” on page 46.

The deletion has an immediate effect on transmission activities, if GIX is running with NETCHNG=IMMEDIATE in the GIX profile and the TCP is running with NETCHNG=IMMEDIATE defined either in the TCP profile or as a TCP startup parameter. The TCP performs the following actions:

- Immediately cancels all existing requests for transmission activities that involve the node being deleted
- Rejects any further requests for transmission activities involving this node.
**Browsing Node Definitions**

You can look at the attributes defined in the DRD for a specific node, or for all of the nodes defined in the DRD. Follow these steps:

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node you want to browse in the **Node type** field, or accept the one shown.

2. Select **Option 4, Define Nodes**, and press Enter.

   The Specify Operation on Node panel is displayed.

3. If you want to look at a specific node, specify the name of the node. There are two ways in which you can do this:
   
   a. If you know the name of the node, type the node name in the **Node name** field. Do not type anything in the **Node model** field. Select **Option 4, Browse**, to look at the node definition for the node name you specified.

   b. If you do not know the name of the node, select the node from a full-list panel. Leave the **Node name** field empty, or type in a partial name followed by *. Leave the selection line blank, and press Enter. This displays the List of Defined Nodes full-list panel. Use the **B** (Browse) command to select the node definition you want to browse, and press Enter.

   The Browse Selected Information panel is displayed. This panel shows the attributes of the node you selected. Use the Scroll field if there is more information than is currently displayed on the screen.

4. Press PF3 when you have finished browsing, to return to the previous panel.

To browse *all* of the node definitions for the node type you specified, select **Option 6, Browse-All**. You do not need to enter a node name or node model.

**Printing Node Definitions**

You can print the attributes defined in the DRD for a specific node, or for all of the nodes defined in the DRD. Follow these steps:

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node definition you want to print in the **Node type** field, or accept the one shown.

2. Select **Option 4, Define Nodes**, and press Enter.

   The Specify Operation on Node panel is displayed.

3. If you want to print a specific node definition, specify the name of the node. There are two ways in which you can do this:
   
   a. If you know the name of the node, type the node name in the **Node name** field. Do not type anything in the **Node model** field. Select **Option 5, Print**, to print the node definition for the node name you specified.

   Press Enter, and you will receive this message:

   **OPERATION COMPLETED. ENTER A NEW REQUEST.**

   b. If you do not know the name of the node, select the node from a full-list panel. Leave the **Node name** field empty, or type in a partial name followed by *. Leave the selection line blank, and press Enter. This displays
Using Transmission Profiles (System Administrator)

the List of Defined Nodes full-list panel. Use the P (Print) command to select the node definition you want to print.

When you press Enter, the message PRINTED will be displayed next to the node name you specified.

When you log off from GIX, you can choose to send this job to the printer. Refer to “How to Print Using GIX” on page 15.

To print all of the node definitions for the node type you specified, select Option 7, Print-All. You do not need to enter a node name or node model.

Using Transmission Profiles

Transmission profiles provide an efficient way of using transmission lines. A transmission profile is a way of grouping together logical units that have the same transmission characteristics, referred to in this book as transmission profile attributes.

The most important transmission profile attribute is the type of transmission line used between the host and the logical units (LUs) associated with the target nodes (the nodes to which you are transmitting). The lines can either be leased or switched. This is defined in the Linetype field of a Specify Node Attributes panel. Figure 15 on page 37 illustrates this. For example, you could define a transmission profile for logical units connected to NetView DM for MVS by point-to-point leased lines.

The other transmission profile attributes are:

- Minimum number of transmission tasks. This is the minimum granted number of concurrent transmission tasks that can be active for this transmission profile.
- Maximum number of transmission tasks that can be active for this transmission profile.
- Retry interval (in seconds) for a line failure.
- Number of times NetView DM for MVS will attempt an automatic transmission retry.

When you define a node using GIX, you must specify the name of the logical unit associated with the node. Each logical unit is automatically assigned to a default transmission profile. There are two default transmission profiles:

- NDMTPL, which is for logical units connected to the host through leased lines
- NDMTPS, which is for logical units connected to the host through switched lines

When NetView DM for MVS is installed, the user can also define up to 1000 other transmission profiles, using a macro called NDMTP. NetView DM for MVS gives priority to transmission requests coming from nodes that belong to a user-defined transmission profile.

Using GIX, you can reassign a logical unit to a different transmission profile than the one it is currently assigned to. You can also browse or print the logical units that are associated with a specific transmission profile, or all transmission profiles. The instructions on the following pages show you how to do this using GIX.
Changing LU to TP Assignments (System Administrator)

**Note:** Network Operators can also redefine transmission profile parameters, such as the line type or the maximum number of concurrent tasks using IOF, described in Chapter 23, “Changing Transmission Profiles” on page 521 in Part 3 of this book.

### Changing Logical Unit to Transmission Profile Assignments

Each logical unit you define to NetView DM for MVS is assigned to a specific transmission profile (sometimes called TP on GIX panels). You can assign a logical unit to a different transmission profile using GIX.

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

#### Step 1: Specify the Transmission Profile Name

1. Select **Option 5, Maintain LU/TP Assignments**, from the Configure Network panel. Leave the **Node Type** field blank.

   The Maintain LU/TP Assignments panel is displayed.

   ![Example of a Maintain LU/TP Assignments panel](image)

   **Figure 17. Example of a Maintain LU/TP Assignments panel**

2. You can either specify the name of a particular transmission profile, or you can choose to look at all defined transmission profiles:

   a. To look at a specific profile, type the name of the transmission profile in the **Transmission Profile name** field. Select **Option 1, Change LU/TP Assignments**.

      The Change LU/TP Assignments panel displays the logical units (LUs) that are assigned to that transmission profile.

   b. To look at all of the defined transmission profiles, leave the **Transmission Profile name** field blank, and select **Option 1, Change LU/TP Assignments**.

      The Change LU/TP Assignments panel displays the LUs that are assigned to all defined transmission profiles.
Step 2: Change LU/TP Assignments

<table>
<thead>
<tr>
<th>NDMPNQAE</th>
<th>CHANGE LU/TP ASSIGNMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>==&gt; ScrollPAGE</td>
</tr>
</tbody>
</table>

Line command : S(SELECT)

<table>
<thead>
<tr>
<th>CMD</th>
<th>LU NAME</th>
<th>TP NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>AA</td>
<td>NDMTPS</td>
</tr>
<tr>
<td>___</td>
<td>AAA</td>
<td>NDMTPS</td>
</tr>
<tr>
<td>___</td>
<td>AAAA</td>
<td>NDMTPL</td>
</tr>
<tr>
<td>___</td>
<td>AAAAA</td>
<td>NDMTP2</td>
</tr>
</tbody>
</table>

Figure 18. Example of a Change LU/TP Assignments panel

This example shows the LUs assigned to all of the transmission profiles that are currently defined. Sometimes, you may see the message TP NOT FOUND on some lines. This means that the transmission profile name associated with the LU no longer exists in the transmission profile table. Refer to the Installation and Customization book for information on how to build a new transmission profile table.

1. Type the line command S (Select) in the CMD field next to the LU name that you want to reassign.

2. Move the cursor to the TP NAME field, and change the name of the transmission profile, overtyping the existing name as required.

3. Press Enter and a message is displayed on each changed line. The message can be one of the following:
   - CHANGED, which means the LU/TP assignment has been successfully changed.
   - LINETYPE CHANGED, which means that the LU/TP assignment has been changed but the line type specified in the new transmission profile does not match the line type of the old profile. This is a warning message.
   - TP NOT FOUND, which means that the transmission profile is not defined.
   - TP NAME REQUIRED, which means that you must specify a transmission profile name in the TPNAME field.
   - INCORRECT TP NAME, which means that the transmission profile name you specified is invalid, and the assignment has not been changed.

4. Press PF3 when you have finished, to return to the previous panel.

The change has an immediate effect on transmission activities, if GIX is running with NETCHNG=IMMEDIATE in the GIX profile and the TCP is running with NETCHNG=IMMEDIATE defined either in the TCP profile or as a TCP startup parameter. The change becomes effective:

- **Immediately**, if there is no active SNA session with the logical unit affected by the change

- **At the end of the session**, if an SNA session is active.

The transmission profile must already have been defined when the TCP was last cold-started.
Browsing Logical Unit to Transmission Profile Assignments

To look at the assignment of a logical unit to a specific transmission profile or the assignments for all transmission profiles, follow these steps:

1. From the Maintain LU/TP Assignments panel (shown in Figure 17 on page 47), you can either specify the name of a particular transmission profile, or you can choose to look at all defined transmission profiles:
   a. To select a specific transmission profile, type the name in the Transmission Profile name field.
   b. To look at all defined transmission profiles, leave the Transmission Profile field blank.

2. Select Option 2, Browse LU/TP Assignments.

The Browse Selected Information panel is displayed. Here is an example that shows the logical units assigned to each defined transmission profile:

```
NDMPG4AE BROWSE SELECTED INFORMATION
Command ==> Scroll ==> PAGE
09:05

LOGICAL UNITS ASSIGNED TO TRANSMISSION PROFILE : NDMTPL
AAA
N16JDSX3
TP50CT02
TP508T02
TP509T02

LOGICAL UNITS ASSIGNED TO TRANSMISSION PROFILE : NDMTPS
--NONE--

LOGICAL UNITS ASSIGNED TO TRANSMISSION PROFILE : NDMTP2
--NONE--
```

Figure 19. Example of a Browse Selected Information panel (transmission profiles)

3. Press PF3 when you have finished browsing, to return to the previous panel.

Printing Logical Unit to Transmission Profile Assignments

To print a list of the logical units assigned to a specific transmission profile or to all defined transmission profiles, follow these steps:

1. From the Maintain LU/TP Assignments panel (shown in Figure 17 on page 47), you can either specify the name of a particular transmission profile, or you can choose to print all defined transmission profiles:
   a. To select a specific transmission profile, type the name in the Transmission Profile name field.
   b. To print all defined transmission profiles, leave the Transmission Profile name field blank.

2. Select Option 3, Print LU/TP Assignments.

You will receive the message:
Maintaining LU to CP Assignments (System Administrator)

OPERATION COMPLETED. ENTER A NEW REQUEST.

When you log off from GIX, you can choose to send this job to the printer.
Refer to “How to Print Using GIX” on page 15.

Maintaining Logical Unit to Connection Profile Assignments

You can assign each logical unit a specific connection profile (sometimes called CP on GIX panels). A connection profile is a set of attributes that are common to a collection of logical units associated with nodes defined in the DRD.

Connection profiles are defined at installation time, using the NDMCP customization macro. The NDMCP macro specifies this common set of attributes. NetView DM for MVS uses these attributes to recognize the connection capabilities of the logical units assigned to a specific CP.

Up to 1000 CPs can be defined at installation time, using this macro once for each CP you want to define. Refer to the Installation and Customization book for information about the NDMCP macro and setting up connection profiles.

You can use GIX to define or change the assignment of logical units to a specific connection profile. You can also deassign a logical unit from a specific profile.

The instructions on the following pages show you how to work with connection profiles using GIX.

Defining Logical Unit to Connection Profile Assignments

Use the following instructions at your terminal. Begin by logging on to the GIX main menu, and follow these instructions.

1. Select Option 6, Maintain LU/CP Assignments, from the Configure Network panel. Leave the Node Type field blank.

   The Maintain LU/CP Assignments panel is displayed.

   Figure 20. Example of a Maintain LU/CP Assignments panel

2. To define an LU/CP assignment, select Option 1, Define/Change LU/CP Assignments.
The Define/Change LU/CP Assignments panel displays the logical units that are known to NetView DM for MVS.

3. To select a logical unit, enter \textbf{S} as a line command.

4. To assign the selected logical unit to a connection profile, enter the name of a connection profile you want this logical unit assigned to. For an example of the panel, see Figure 21.

\textbf{De-assigning a Connection Profile}

To de-assign a logical unit, erase the name of the connection profile that appears with the particular logical unit (for example, see Figure 21).

\textbf{Changing the Logical Unit Assignment to a Connection Profile}

To change the logical unit assignment, enter a new connection profile name.

\begin{verbatim}
NDMPNRAE DEFINE/CHANGE LU/CP ASSIGNMENTS
Line command : S(SELECT)
CMD LU NAME CP NAME
  AA N15JDSX3 NDMCPS
  ___ TPSQCT02 NDMCPL
  ___ TPSQ8T02 NDMCP2
  ___ TPSQ9T02
********** BOTTOM OF DATA **********
PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR
\end{verbatim}

\textit{Figure 21. Example of a Change LU/CP Assignments panel}

This example shows the logical units assigned to all of the connection profiles that are currently defined. Sometimes, you may see the message \textit{CP NOT FOUND} on some lines. This means that the connection profile name associated to the logical unit no longer exists in the connection profile table. Refer to the \textit{Installation and Customization} book for information on how to build a new connection profile table.

1. Type in the line command \textbf{S} (Select) in the \textbf{CMD} field next to the LU name that you want to reassign.

2. Move the cursor to the \textbf{CP Name} field, and change the name of the connection profile, overtyping the existing name as required.

3. Press Enter and a message is displayed on each changed line. The message can be one of the following:
   - \textbf{CHANGED}, which means the LU/CP assignment has been successfully changed.
   - \textbf{LINETYPE CHANGED}, which means that the LU/CP assignment has been changed but the linetype specified in the new connection profile does not match the linetype of the old profile. This is a warning message.
• CP NOT FOUND, which means that the connection profile is not defined.
• CP NAME REQUIRED, which means that you must specify a connection profile name in the CP Name field.
• INCORRECT CP NAME, which means that the connection profile name you specified is invalid, and the assignment has not been changed.

4. Press PF3 when you have finished, to return to the previous panel.

The change has an immediate effect on transmission activities, if GIX is running with NETCHNG=IMMEDIATE in the GIX profile and the TCP is running with NETCHNG=IMMEDIATE defined either in the TCP profile or as a TCP startup parameter. The change becomes effective:

• Immediately, if there is no active SNA session with the logical unit affected by the change
• At the end of the session, if an SNA session is active.

The connection profile must already have been defined when the TCP was last cold-started.

Browsing and Printing LU/CP Assignments
If you select 2 or 3 and enter a connection profile name, the selected CP name is displayed, followed by all the logical units assigned to it.

If you select 2 or 3 and do not enter a connection profile name, the names of all CP names and their associated logical units are displayed.

Browsing Logical Unit to Connection Profile Assignments
To look at the assignment of a logical unit to a specific connection profile or the assignments for all connection profiles, follow these steps:

1. From the Maintain LU/CP Assignments panel (shown in Figure 20 on page 50), you can either specify the name of a particular connection profile, or you can choose to look at all defined connection profiles:
   a. To select a specific connection profile, type the name in the Connection profile name field.
   b. To look at all defined connection profiles, leave the Connection profile field blank.
2. Select **Option 2, Browse LU/CP Assignments**.

   The Browse Selected Information panel is displayed. Here is an example that shows the logical units assigned to each defined transmission profile:

   ![Browse Selected Information Panel Example](image)

   **Figure 22. Example of a Browse Selected Information panel (connection profiles)**

   3. Press PF3 when you have finished browsing to return to the previous panel.

**Printing Logical Unit to Connection Profile Assignments**

To print a list of the logical units assigned to a specific connection profile or to all defined connection profiles, follow these steps:

1. From the Maintain LU/CP Assignments panel (shown in Figure 20 on page 50), you can either specify the name of a particular connection profile, or you can choose to print all defined connection profiles:
   a. To select a specific connection profile, type the name in the **Connection profile name** field.
   b. To print all defined connection profiles, leave the **Connection profile** field blank.

2. Select **Option 3, Print LU/CP Assignments**.

   You will receive the message:
   
   **OPERATION COMPLETED. ENTER A NEW REQUEST.**

   When you log off from GIX, you can choose to send this job to the printer. Refer to “How to Print Using GIX” on page 15.
Organizing Nodes into Groups

This section tells you how to organize nodes into groups using GIX.

Node Groups of the Same Type

You can organize nodes of the same type into a group. When you create a group of nodes you identify it by assigning it a group name. Using groups saves you time when you are defining transmission plans. It can also reduce the volume of data sent between the host and intermediate nodes. You can then specify this group name in your transmission plan, instead of specifying each node name. This means that you only have to perform the task once, instead of repeating the task for each node. Tasks include:

- Loading and unloading resources, into and out of the resource repository.
- Specifying functions in transmission plans. You can create a node group that logically corresponds to the task you are performing, for example, “send the same report to all sales nodes.”

When you define a group, your user ID becomes the owner of the group definition. Other users can use the definition, but only you can change or delete it. If you are a system administrator with the User profiles authorization, you can also delete any group.

You will usually define node groups for all the entry points that are connected to an intermediate node in a multi-tiered network. If you want to send the same resource to all the entry points connected to an intermediate node, the best way to do this is to define the entry points as a group. The resource needs to be transmitted once, between the host and the intermediate node. You can also define groups that specify a subset of the entry points connected to an intermediate node. When you specify the subset in a transmission plan, the intermediate node will only send the resource to the entry points in the subset group.

Group of LU 6.2 Nodes with Different Node Types

You can define a group of nodes where the nodes have different node types. This only applies to user-defined node types (LU 6.2 nodes) with either CMFP or CMEP functional capabilities. This means you can omit the node type when defining a group of nodes.

When the node type of a group of nodes is not specified, you can include in this group LU 6.2 nodes that have:

- Different node types
- Functional capabilities of the node type to which they belong, either CMEP or CMFP

You can define a node group without defining its node type only if you are authorized to use at least one LU 6.2 node type with either CMEP or CMFP functional capabilities.
Organizing Nodes into Groups (System Administrator)

Automatic Node Groups (Server Node Groups)
When defining a node group, you can also have NetView DM for MVS automatically create another node group that contains the server nodes of those nodes in the initial group. You can define the following:

- **Group name** of the automatically-created server group
- **Disposition** of the server group, which can have a value of either New or Old.

Define Group
Use the Define Group option in GiX to define the server group's name and disposition.

To create a server node group, the initial node group must meet these conditions:

- All the nodes in the initial group must be defined in the DRD.
- Each of the nodes in the initial group must have the same node type as the group's node type (if defined). For any initial group with an undefined node type, all nodes in the group must be LU 6.2 nodes with either CMEP or CMFP functional capabilities. If the initial group has an undefined node type:
  - For CMEP nodes, NetView DM for MVS adds the CMEP node server to the server group
  - For CMFP nodes (where no server node exists), to address the server group, NetView DM for MVS adds the CMFP node name to the server group

A server group is always created with an undefined node type (a mixed group).

GiX Change Group
Use the Change Group option in GiX to change the server group's name and disposition. Specifying the name of a server group when changing a node group lets you create or update a server group, starting from an existing node group. Here are your options:

- The node group to be changed is empty and you specify only the server group name and disposition to require the server group creation or update (no changes to the node group). An empty server group is created.
- The node group to be changed is not empty and you specify only the server group name and disposition to require the server group creation or update (no changes to the node group). The request is accepted and the server group is created or updated.
- You insert or delete nodes from the group of nodes, and specify the name of the related server group.

The request will be accepted and the server group updated or created (if not existing) depending on the disposition value.

GiX Delete Group
When deleting a node group from the GiX panel, and if you have specified a corresponding server group name, both groups are deleted regardless of the value of the disposition field.
Defining a Group of Nodes Using GIX

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

**Step 1: Specify Node Type of Group**

1. Select **Option 7, Manage Groups**, from the GIX main menu.

   The Manage Groups panel is displayed.

   
   ```
   NDMPQEAE MANAGE GROUPS
   Selection ==> Time ==> 15:52
   Enter allowed value or accept the one shown:
   NODE TYPE CMEP ALLOWED ENTRIES: CMEP CMFP SERV CLNT AS40
   
   Select one of the following:
   1 DEFINE GROUPS OF NODES
   2 DEFINE GROUPS OF RESOURCES
   ```

   Figure 23. Example of a Manage Groups panel

2. Type the node type, choosing from the list of options displayed, or accept the value shown.

   *Note:* You do not have to enter a node type for the group if the group has nodes with different LU 6.2 node types, as long as each LU 6.2 node type has either CMEP or CMFP functional capabilities.

3. Select **Option 1, Define Groups Of Nodes**.

   The Define a Group of Nodes panel is displayed.

The data you can enter on the Manage Groups panel is described in the following section.

**Node type:** This defines the type of node. Table 70 on page 561 and Table 71 on page 562 show the possible node types you can define, depending on your installation.

*Note:* Nodes in a group must be of the same type, unless you are defining a node group with different LU 6.2 node types.
Step 2: Specify a Group Name

NDMPQBBE DEFINE A GROUP OF NODES (CMEP)
Selection ==> 

Enter:

Group name ______ Blank or partial name followed by * for full-list
Group model ______ Allowed for the CREATE option only
(Enter * to have all defined
nodes as a group model)
Group creation 1 = Inclusion 2 = Exclusion
criteria : _ Allowed only for CREATE option and when group
model is specified
Server Group Name : ________ Server Group Name
Server Group Disposition : 1 1 = NEW 2 = OLD

Select one of the following:
1 CREATE Create a group of nodes
2 CHANGE Change one or more node group items
3 DELETE Delete a group of nodes
4 BROWSE-ALL Browse all node groups
5 PRINT-ALL Print all node groups
PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 12=CURSOR

Figure 24. Example of a Define a Group of Nodes panel

1. On the Define a Group of Nodes panel, type a symbolic group name for the
group in the Group name field.

2. You can optionally enter the name of a group model in the Group model field,
to use an existing group definition as a model for the one you are creating.
Enter * if you want to use all defined nodes as a group model.

3. If you specified a group model, you can specify whether you want to delete all
the nodes that you want excluded from the group, or select all the nodes that
you want to include in the group. Enter 2 in the Group creation criteria field
for the exclusion option. This is the default value. Enter 1 in this field for the
inclusion option.

4. Specify a server group name and server group disposition as follows:
   - **Server Group Name:**
     Type the name of the server group that you want to create in the Server
Group Name field. You can enter from 1 to 8 alphanumeric characters;
the first character must be alphabetic. This field is optional, but there is no
default value. If you do not specify a name, no server group is created.
   - **Server Group Disposition:**
     Type either 1, which means New, if the server group is a new one, or 2,
which means Old, if the server group already exists and must be replaced.
The default is 1 (New).

5. Select Option 1, Create, to create a group of nodes.
The Create Node Groups panel is displayed.
The fields displayed on the Define a Group of Nodes panel are described in the following section.

**Group name:** Enter a symbolic name for the group. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore). The first character must be alphabetic or special. Group names must be unique across node groups of different types.

**Group model:** This is the name of an existing node group definition. The system displays the attributes defined for this definition, and you can overwrite them for the new definition.

### Step 3: Create the Node Group

```
NDMPQSAE CREATE/CHANGE NODE GROUPS (CMEP)
Command ==> Scroll ==> PAGE
Primary command: SAVE
Line commands: I(Insert) R(Replicate) D(Delete) (Multiple commands allowed)

Node group name: SAMPLE   Node group owner: TMSLEW

CMD  ITEM
    CMEPNODE
    TMSEP1
    TMSEP2

Figure 25. Example of a Create/Change Node Groups panel
```

1. Use the following line commands to edit the group definition:
   - Use the I (Insert) or R (Replicate) commands to add new lines to the full-list. Once you have created a new line on the full-list, type the name of the node you want to include in the group.
   - Use the D (Delete) command to delete lines from the full-list.

2. Type **SAVE** on the command line when you have finished editing the group, and press Enter to save the new group in the DRD. To quit without saving the new group, press PF3 to return to the previous panel, or PF4 to return to the GIX main menu.

When you are editing the group definition, follow the guidelines described in the following section.

**Node groups:** A node can belong to more than one group. You can include nodes that are not yet defined in the DRD, but you must define them before you submit plans that address the nodes. If you do not define them, you will get a validation error when you submit the plan. If you define an empty group, you will also get a validation error when you try to use the group in a plan.

**Note:** If you delete a group definition, the node definition is not deleted.
Changing a Node Group

You can change a node group that is defined in the DRD. Follow these steps:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown. Leave the node type field empty to change groups with different LU 6.2 node types.

2. Select Option 1, Define Groups of Nodes.

The Define a Group of Nodes panel is displayed.

3. Specify the name of the group that you want to change. There are two ways in which you can do this:
   a. If you know the name of the group, type the group name in the Group name field. Select Option 2, Change, and press Enter.
   
   b. If you do not know the name of the group, select the group from a full-list panel. Leave the Group name field empty, or type in a partial name followed by *, and press Enter. This displays the Select or Delete Node Groups full-list panel. Use the S (Select) line command to select the group or groups, and press Enter.

4. The Create/Change Node Groups panel is displayed. Follow the instructions as described in “Step 3: Create the Node Group” on page 58, to change items in the group.

Deleting a Node Group

To delete an existing node group definition from the DRD, follow these steps:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown. Leave the node type field empty to change groups with different LU 6.2 node types. Do not specify a node type when deleting a node group for which you have specified a corresponding server group name.

2. Select Option 1, Define Groups of Nodes.

   The Define a Group of Nodes panel is displayed.

3. Specify the name of the group that you want to delete. There are two ways in which you can do this:

   a. If you know the name of the group, type the group name in the Group name field. Select Option 3, Delete, and press Enter.

   b. If you do not know the name of the group, you can select the group from a full-list panel. Leave the Group name field empty, or type in a partial name followed by *, and press Enter. This displays the Select or Delete Groups full-list panel. Use the D (Delete) line command to select the group or groups, and press Enter.

   The Confirm Deletion of Node Group panel is displayed.

4. Type the primary command D (Delete) on the command line to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel.
Browsing Node Groups

To browse the definitions of all node groups defined in the DRD, follow these steps:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown. Leave the node type field empty to browse groups with different LU 6.2 node types.

2. Select Option 1, Define Groups of Nodes.
   The Define a Group of Nodes panel is displayed.

3. Select Option 4, Browse-All, to look at all the node groups that are defined in the DRD, and press Enter.
   The Browse Selected Information panel is displayed. This panel shows information about each node group that is defined in the DRD. For each node group, the panel displays the group name, group owner (the user ID of the person who created the group), the node type, and the name of each node in the group. Use the Scroll key if there is more information than is currently displayed on the screen.

4. Press PF3 when you have finished browsing to return to the previous panel.

Printing Node Groups

To print the definitions of all node groups that are defined in the DRD, follow these steps:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown. Leave the node type field empty to print groups with different LU 6.2 node types.

2. Select Option 1, Define Groups of Nodes.
   The Define a Group of Nodes panel is displayed.

3. Select Option 5, Print-All, to print all of the definitions of node groups that are defined in the DRD, and press Enter. You will receive this message:
   OPERATION COMPLETED. ENTER A NEW REQUEST.
   When you log off from GiX, you can choose to send this job to the printer. Refer to “How to Print Using GiX” on page 15.
Chapter 4. Defining Resources

Defining resources is the third step in the following sequence of tasks after NetView DM for MVS has been installed:

Step 1. Define profiles for all NetView DM for MVS users
Step 2. Define each node
Step 3. Define each resource
Step 4. Assign resources to nodes

This chapter explains how to define resources in the distributed resource directory (DRD). Once you have created definitions using GIX, you can also:

- Change a resource definition
- Delete a resource definition
- Browse the contents of a resource definition
- Print the contents of a resource definition

What Is a Resource Definition?

A resource definition describes resources in the network to NetView DM for MVS. Resource definitions are stored in the DRD. There are some resource types that you must define to NetView DM for MVS before users can validate a transmission plan that refers to the resource. Table 3 on page 63 shows which resource types must be defined. If a transmission plan refers to a resource which you have not defined, NetView DM for MVS issues a warning or error message when it tries to validate the plan.

When you define a resource, you must specify:

- The type of node the resource can be transmitted to
- The resource type, listed in Table 3 on page 63
- A symbolic name which identifies the resource
- A set of resource attributes for the resource at the host and at the node

Once you have defined nodes and resources, you can assign the resource to a node, or group of nodes. You can also organize nodes and resources into groups, and track resources in resource history records, as described in “Organizing Resources into Groups” on page 79 and “Tracking Resources” on page 90.

Resource Types

Resources are classified into types. These types fall into four broad categories:

- Microcode resources, such as microcode, microcode customization data, and related changes, updates, or patches
- Software resources, such as system software, application software, and related updates and changes
- Application data, such as flat data, relational data, application logs, and related updates
- Problem documentation resources, such as dumps, configuration files, traces, and error logs
What Is a Resource Definition? (System Administrator)

Throughout this book, resources are called either LU0 resource types or LU 6.2 resource types. LU0 resource types are those resources that are supported by LU0 node types (as shown in Table 70 on page 561). Here are the LU0 resource types:

- CLIST
- Data set
- Errorlog
- Job
- Library
- Member
- Panel
- Print
- Program
- Program temporary fix (PTF)
- Storage dump

LU 6.2 resource types are those supported by LU 6.2 node types (as shown in Table 71 on page 562). Here are the LU 6.2 resource types:

- AS/400 data object containment structure
- AS/400 object
- Configuration file
- Dump
- Error log
- Flat data
- Microcode and microcode customization
- Procedure
- Relational data
- Software
- Trace

NetView DM for MVS supplies some initial values which show whether to define, assign, track, or group specific resource types. For LU 6.2 resource types, you can change these initial values at installation time using a macro called NDMRES, or after installation, using a partial installation step (GENTYPE=PROFILE). Refer to the Installation and Customization book for information.

For LU 6.2 resource types, the tracking value can be overridden while preparing a plan at phase time (see Chapter 7, “Preparing and Submitting Transmission Plans” on page 119), either at the function level or at the phase level.

Table 3 on page 63 shows the resource types you can define, and shows the initial values for defining, assigning, tracking and grouping resources.
<table>
<thead>
<tr>
<th>Type</th>
<th>DEF</th>
<th>RES</th>
<th>ASS</th>
<th>TRA</th>
<th>GRP</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data-object containment structure(CMEP)</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0158</td>
</tr>
<tr>
<td>AS/400 object (CMEP)</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0156</td>
</tr>
<tr>
<td>CLIST</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0040</td>
</tr>
<tr>
<td>Configuration file</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0220</td>
</tr>
<tr>
<td>Data set</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0030</td>
</tr>
<tr>
<td>Dump (CMEP)</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0230</td>
</tr>
<tr>
<td>Errorlog (DPPX)</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>0034</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0250</td>
</tr>
<tr>
<td>Flat data</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0100</td>
</tr>
<tr>
<td>Job</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>0041</td>
</tr>
<tr>
<td>Library</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>0052</td>
</tr>
<tr>
<td>Member</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y/N</td>
<td>0050</td>
</tr>
<tr>
<td>Microcode and microcode customization</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>Y</td>
<td>Y/N</td>
<td>0060</td>
</tr>
<tr>
<td>Panel</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0036</td>
</tr>
<tr>
<td>Print</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0038</td>
</tr>
<tr>
<td>Procedure</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0080</td>
</tr>
<tr>
<td>Program</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0042</td>
</tr>
<tr>
<td>Program temporary fix (PTF)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>0044</td>
</tr>
<tr>
<td>Relational data</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0120</td>
</tr>
<tr>
<td>Software</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0070</td>
</tr>
<tr>
<td>Storage dump (LU0)</td>
<td>Y/N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>0032</td>
</tr>
<tr>
<td>Trace</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0240</td>
</tr>
</tbody>
</table>

Table 3. Resource types and initial values

The first column (Type) shows the resource types that you can use with NetView DM for MVS.

The second column (DEF) shows whether you have to define the resource in the DRD. There are some types of resource that you do not have to define. The values can be:

**Y (Yes)**    You must always define resources of this type.

**N (No)**     You do not have to define resources of this type.

**Y/N (Yes/No)** You can choose whether to define resources of this type.

**Note:** If you change an initial value from Yes to No for a resource type, you will no longer be able to print or delete already existing definitions in the DRD for this resource type.
What Is a Resource Definition? (System Administrator)

The third column (RES) shows whether the resource is automatically restricted, optionally restricted, or not restricted. You can change this default value when you are defining the resource. If you want to assign the resource to a node, you must define it as restricted. The values are:

Y (Yes) The resource type is automatically defined as restricted.
N (N) The resource type is unrestricted, or you do not have to define it at all.
Y/N (Yes/No) If you define the resource, it is automatically restricted. If you choose not to define it, the resource is unrestricted.

The fourth column (ASS) shows whether you must assign the resource. If you want to assign a resource to a node, you must define the resource as restricted. NetView DM for MVS will only transmit these resource types to, or delete these resource types from, those nodes to which you have assigned the resource. You must already have defined the nodes to NetView DM for MVS. The values can be:

Y (Yes) You must assign the resource.
N (No) You cannot assign the resource.
Y/N (Yes/No) You can assign the resource only if it is defined as restricted.

The fifth column (TRA) shows whether the resource is tracked by NetView DM for MVS at transmission time. NetView DM for MVS tracks resources in resource history records. These are explained in “Tracking Resources” on page 90. The values can be:

Y (Yes) NetView DM for MVS automatically tracks the resource.
N (No) NetView DM for MVS does not track the resource.
Y/N (Yes/No) You can choose whether to track the resource.

The GRP column shows whether you can organize resources of this type into groups. Refer to “Organizing Resources into Groups” on page 79 for information. The values can be:

Y/N (Yes/No) You can choose whether to group the resources.
N (No) You cannot group the resources.
Naming Resources

Each resource in your network must be uniquely identified to NetView DM for MVS. To do this, you give the resource a **resource name**. You define the name when you are defining the resource, either using GIX or Batch Utilities.

The conventions for defining a resource name vary between the different resource types. The following pages begin by describing the naming conventions for LU0 resource types, and then describe the conventions for LU 6.2 resource types. The table below summarizes the basic differences between LU0 resource type conventions, and those for LU 6.2 resource types.

<table>
<thead>
<tr>
<th>LU0 Resource Types</th>
<th>LU 6.2 Resource Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific naming conventions for each resource type</td>
<td>Enterprise-structured names, or structured names for microcode</td>
</tr>
<tr>
<td>The name must be fully qualified</td>
<td>The name does not have to be fully qualified. You can use partial naming.</td>
</tr>
<tr>
<td>Names cannot use subtrees</td>
<td>Some resource types can use subtrees</td>
</tr>
</tbody>
</table>

Table 4. Comparison of naming conventions for LU0 and LU 6.2 resources

### Naming LU0 Resources

Each LU0 resource type has a different set of naming conventions that you should follow when defining resource names. The syntax conventions for each resource are described in Appendix C, “Resource Naming Conventions” on page 569.

### Naming LU 6.2 Resources

With NetView DM for MVS, you use a **global name** for each file, consisting of a set of **tokens**. For example, in the global file name CORPID.9135.NA.PATCH.1234, CORPID is a token, 9135 is a token, PATCH is a token, and so on. A global name can have up to 10 tokens separated by periods, and each token can have up to 16 characters (uppercase alphabetic and numeric characters only). The total name length cannot be more than 64 bytes, including the separator characters.

LU 6.2 resource types use **enterprise-structured names**. **Enterprise-structured** means that each company, or “enterprise,” can design the name of its own resources. The only predefined token is the first token, which is the **corpid**. The corpid is the enterprise code of your company, defined in the structured **netid** assigned to your company, and registered, by IBM. As for the remaining tokens, you can specify whatever you want.

This is the basic format of an enterprise-structured name:

```
corpid.token.token.token . . . . token
```

If you are using microcode, software, flat data, or relational data resources, then you can also use **subtree** and **version** tokens in the resource name. A subtree is a set of tokens that already have a structure defined. You specify a leading token before the subtree, called a **subtree identifier (STI)** token. The STI token shows the
Naming Resources (System Administrator)

type of object that you are naming. For NetView DM for MVS distribution or change control requests, you can use the following STI tokens:

- REF: Refresh object
- UPD: Update object
- FIX: Fix object
- LIB: Library object
- MEM: File member object
- OBJ: AS/400 save restorable object type

Each STI token determines a predefined subtree structure, optionally followed by version tokens. Here is the structure of an enterprise-structured name using a subtree:

```
corpid(mm).(mm or nnn)....(mm or nnn).subtree
```

where the subtree format is:

```
(mm).(mm or nnn)....(mm or nnn)
```

*mm* refers to tokens that must match. *nnn* refers to tokens that need not match. For each of the STI tokens, these are the subtree structures that are defined. *mm* in brackets shows that the token must match.

- REF(mm).level
- UPD(mm).oldlevel.newlevel
- FIX(mm).level.fixid(mm)
- LIB(mm).libraryname.targetrelease
- MEM(mm).libraryname.filename.membername.targetrelease
- OBJ(mm).libraryname.objectname.objecttype.targetrelease

You can also use one or more version tokens in the resource name, for example, to distinguish between the French and German versions of a change object.

### Example

Here is an example of an enterprise-structured name which uses both a subtree and a version token:

```
EURO.DONUTS.MESSAGES:UPD.12.13.FRENCH
```

This global name describes the level 13 update from level 12 for the French version of the messages sub-component of the donuts application owned by the Euro enterprise. The colon (:) introduces the subtree, which in this case, is UPD.12.13.

Microcode is the only LU 6.2 resource type which can use a convention other than enterprise-structured names. Microcode and microcode customization data can also use *structured* names, in which the first token must be MCODE or MCUST, respectively. Whenever you specify MCODE or MCUST as the first token of a name, using GIX or the SUBMIT or MAINTDAT Batch Utilities, NetView DM for MVS checks to make sure that the resource type is microcode, identified by a resource type code of 0060.

Whether you are using a structured name for microcode, or an enterprise structured name for any LU 6.2 resource type, you can also use *partial naming*, instead of

---

1 There is also an STI token called GRP (group object), but this is only used for local definition.
specifying all the qualifiers in a name. The next section describes how to use partial naming.

**Partial Naming and Matching for LU 6.2 Resource Types**

With NetView DM for MVS you can *partially name* a resource definition and *partially match* resources in a transmission plan.

**Partial Naming for Resource Definitions**

For LU 6.2 resources only, you can perform partial naming of resources, which means you can:

- Specify defaults for a resource name definition at customization time
- Update these values whenever necessary
- Define a unique resource name by a partial name so you will not have to repeat the definition and assignment for each resource name in the network

When you define a specific resource, you can enter a resource name that is a partially-defined name. A name is partially defined when one or more tokens of the name are set to a percent sign (%).

**Resource Name Masks:** Partial naming is possible because at customization time you defined a *resource name mask* for the resource type of the resource you are defining. You can, for each type of LU 6.2 resource, specify one *general mask* and up to 10 *particular masks*. These allow NetView DM for MVS to verify a resource name during a resource definition.

Each resource name mask contains 10 values separated by periods (.), where each value corresponds to one possible token position in a resource name. These values determine whether NetView DM for MVS requires a specific value to be given to a token (Y), or whether it can be set to % (N), during the resource name definition.

For example, a resource name mask might look like this:

```
Y.N.Y.N.Y.Y.Y.Y.N.N
```

If no resource masks are specified for a certain resource type at customization time, this default value is assumed for that resource type:

```
Y.Y.Y.Y.Y.Y.Y.Y.Y
```

The first token of a resource name is either:

- corporation id
- or
- MCODE or MCUST

This meant that the first value in a resource name mask must either be Y, or a specified value that NetView DM for MVS *explicitly* compares with the corresponding value that you enter.

- Example of a general mask:
  
  RMASK1=Y.Y.Y.N.N.Y.N.N

- Example of a particular mask:
  
  RMASK2=MYCORPID.Y.Y.N.N.Y.N.N

Chapter 4. Defining Resources 67
A general mask must have Y in the first position. A particular mask must have an explicit value in the first position.

To find out how the masks have been defined in your installation of NetView DM for MVS, refer to the NDMRES customization macro defined in the customization JCL used to customize NetView DM for MVS. The *Installation and Customization* book contains information about the NDMRES macro, and the JCL used to install NetView DM for MVS.

**Resource Partial Naming Using Resource Name Masks:** The general mask is used by NetView DM for MVS during definition of a resource name when the first token that you specify does not match any of the explicit values in any defined particular masks you might have defined.

Here is an example of resource definition using partial names.

- A file is produced by an application program every day and has to be distributed to the nodes to which it is assigned.
- The name of the file consists of four tokens. Three of the four tokens are always the same and the fourth represents the date.

You define the resource (and assign it), once, by specifying an explicit value for the three fixed tokens, and % for the fourth one, as shown below:

```
SMITH.APPL1.DATA.%
```

This is possible because for the corresponding resource type, you defined the following particular mask:

```
RMASK3=SMITH.Y.Y.N.N.N.N.N.N
```

**Partial Matching of Resources in Transmission Plans**

Use partial matching in transmission plans to find resources to send, retrieve, or delete. When you prepare a transmission plan you can use partial naming qualification, specifying an asterisk (*) instead of a qualifier in one or more of the tokens of the resource name.

A partially matched resource name lets you write a function in a transmission plan in a *generalized* way. However, the *qualified* tokens in the resource name (as specified in your plan) must be sufficient to identify a unique resource definition in the DRD and a unique object in the source or target repository.

For the Send function, the searched repository is the NetView DM for MVS resource repository. For a Retrieve or Delete function, the searched repository is the repository of the destination node.

If you specify a partial name in a transmission function and the resource (in the resource repository) belongs to a resource type that requires the existence of a resource definition (in the DRD), NetView DM for MVS looks for a resource that has the following definition in the DRD:

- A % value in the resource name definition for those tokens indicated by * in the transmission plan. In the resource name mask this token would be set to N.
- The *exactly matching values* in the resource name definition for the other tokens. In the resource name mask these tokens would be set to Y.
Here is an example of a transmission function using partial matching:

- This file, for the current day, exists in the resource repository:
  
  SMITH.APPL1.DATA.911201

- You prepare a transmission function that looks like this:

  SEND RESO NAME=SMITH.APPL1.DATA.*, RESTYPE=FLATD,
  MATCHIND=(4,H),...........

- When the plan is submitted, the most recent version of the resource, that is, SMITH.APPL1.DATA.911201, is fetched from the resource repository and sent to the specified node, if the following resource definition exists in the DRD:

  SMITH.APPL1.DATA.%

---

**Defining Resources Using GIX**

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

**Step 1: Define the Node Type**

1. Select **Option 1, Configure Network**, from the GIX main menu.

   The Configure Network panel is displayed. This panel is shown in Figure 14 on page 35.

2. Type the node type that the resource belongs to, selecting one of the options displayed, or accepting the value shown, in the **Node Type** field.

3. Select **Option 1, Define Resources**.

   The Define Resource panel is displayed, shown in Figure 26 on page 70.

The data you can enter on the Configure Network panel is described below.

**Node type:** This defines the type of node that the resource belongs to. Table 70 on page 561 and Table 71 on page 562 show the possible node types.
Step 2: Define the Resource Type

Figure 26. Example of a Define Resource panel. This example is for a CMEP node type. The panel shows the resource types that you can define for CMEP nodes.

1. Select one of the resource types displayed by typing the corresponding number in the selection field.

   The Specify Operation on Resource panel is displayed.

The data you can enter on the Define Resource panel is described below.

**Resource types:** GIX displays only those resource types that you can define for this node type. Table 3 on page 63 shows the resource types you can define.
Step 3: Specify Operation on Resource

<table>
<thead>
<tr>
<th>NDMP-PNXE</th>
<th>SPECIFY OPERATION ON RESOURCE (CMEP SOFTWARE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection ==&gt; 1</td>
<td>15:52</td>
</tr>
<tr>
<td>Enter desired values or accept the ones shown:</td>
<td></td>
</tr>
<tr>
<td>Resource name ==&gt; CORPID.%.%.%.%.%.%.%</td>
<td>Blank or partial name followed by * for full-list</td>
</tr>
<tr>
<td>Resource model ==&gt;</td>
<td>Allowed for the CREATE option only &lt;==&gt;</td>
</tr>
<tr>
<td>Partially defined Name 1 (1 = Yes, 2 = No)</td>
<td></td>
</tr>
</tbody>
</table>

Select one of the following:

1 CREATE  Create a new resource definition
2 CHANGE  Change a resource definition
3 DELETE  Delete a resource definition
4 BROWSE  Browse a resource definition
5 PRINT  Print a resource definition
6 BROWSE ALL  Browse all resource definitions
7 PRINT ALL  Print all resource definitions

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 12=CURSOR

Figure 27. Example of a Specify Operation On Resource panel. This example is to define a software resource type for a CMEP node type.

1. Type a symbolic resource name in the Resource name field.

   Note: Resource names starting with $DELETE.$PENDING are reserved, and may not be used.

2. You can optionally enter the name of a resource model in the Resource model field, to use an existing resource definition as a model for the one you are creating.

3. Specify whether the name you entered is a partial name. Type 1 in the Partially defined Name field to mean Yes, if the name is partially defined, or 2, which means No, if the name is fully qualified. This is the default value.

4. Select Option 1, Create, to create a new resource definition.

   The Specify Attributes for node type/resource type panel is displayed. The values node type/resource type are variables, indicating the type of node and type of resource that you are working with.

The fields displayed on the Specify Operation on Resource panel are described below.

**Resource name:** This is a symbolic name which identifies the resource in transmission plans. The name you assign can have up to 64 characters (for global names), including periods. The periods (.) are delimiters between qualifiers. Each qualifier can have from 1 to 8 characters.

There are specific naming conventions for each resource type. Resource naming conventions for resources used with nodes with CMEP functional capabilities (as
shown in Figure 27) are described in “Naming Resources” on page 65 and Appendix C, “Resource Naming Conventions” on page 569. For LU 6.2 resources, you can specify % values for need-not-match tokens in the resource name. You must also enter 1 (Yes) in the Partially Defined Name field. If the name you enter does not match the resource mask, the mask is displayed on the panel.

Note: The resource name must be unique within the resource type. You cannot define two resources, which are of the same resource type, with the same name.

Resource model: This is the name of an existing resource definition. The system displays the attributes defined for this definition, and you can overwrite them for the new definition. For LU 6.2 resources, partially defined names are allowed as resource models.

Partially defined name: This field is displayed for LU 6.2 resource types only. The field indicates whether or not the resource name you enter is a partially defined name (1 means Yes), or a fully qualified name (2 means No). You use a percent sign (%) to represent partially defined tokens in a resource name. Partially defined names are explained in “Partial Naming and Matching for LU 6.2 Resource Types” on page 67.

Step 4: Specify the Resource Attributes

Figure 28. Example of a Specify Attributes For Resource panel. This examples show the attributes for a software resource type and for a CMEP user-defined node type.

1. Type the values for each attribute in each field, described below. The attributes on this panel specify how the resource you are defining will be processed at the host and at the node.

2. If you specify 1, which means Yes, in the Automatic Assignment field, the Specify Node panel is displayed. Refer to “Step 5 (optional): Automatically Assign the Resource to a Node” on page 74 for information.

3. Press Enter when you have finished. The new resource definition is saved in the DRD and GIX returns you to the Specify Operation on Resource panel.
You can then define another resource, or press PF4 to return to the GIX main menu.

The Specify Attributes for Resource panel may display any of the following fields (listed alphabetically below), according to the node type.

**Automatic assignment:** You must assign restricted resources to a particular node, or group of nodes. Transmissions for restricted resources can only take place with nodes to which they are assigned. This field provides a quick way to assign a resource.

You can assign resources automatically by specifying 1, which means Yes, in this field. This displays the Specify Node panel, which you use to specify the node you are assigning the resource to. If you do not want to assign the resource, the default value in this field is 2, which means No.

**Class:** The resource class is a way of authorizing only certain users to work with specific resources. Only users authorized to use the specific class assigned to the resource can work with it. A resource class consists of two alphanumeric characters; the first character must be alphabetic. All users are authorized to use class A0 resources.

**Name at node:** This field is not shown in Figure 28 on page 72. It does not apply to LU 6.2 node types (such as those with CMEP functional capabilities).

If you want the resource to have a different name at the node, enter the name in this field. If you leave the field blank, the resource will have the same name as at the host.

You can enter the name as a string of up to 3 qualifiers, separated by periods (.). Each qualifier can have from 1 to 8 alphanumeric or special characters. You can use an asterisk (*) as a substitute for part of the name at the host. For example, if you enter a name at node of ITALY.*.*.* and the name at the host is HQ.PAYROLL.BONUS, the name generated at the node will be ITALY.PAYROLL.BONUS. The exact format of the qualifier depends on the node type.

Refer to Appendix B, “NetView DM for MVS Node, Function and Resource Types” on page 561 for information about resource qualifiers for each node type. “Naming LU 0 Resources at the Node” on page 570 explains how to name LU0 resources at the node.

**Note:** For EDX resources, you can only specify two qualifiers.

**Restriction:** You can either enter 1, which means Yes, or 2, which means No. If you enter 1, the resource is restricted to specific nodes which you must assign to the resource. Users can only transmit the resource to the specified node or nodes.
Step 5 (optional): Automatically Assign the Resource to a Node
To assign a resource to a specific node, use the following panels:

![Configure Network Panel](image1.png)

Figure 29. Example of Configure Network panel

![Specify Resource Assignment Operation Panel](image2.png)

Figure 30. Example of a Specify Resource Assignment Operation panel
Defining Resources Using GIX (System Administrator)

NDMPRSAE \ SPECIFY NODE AND SELECT RESOURCE TYPE (CMEP)
Selection ==> 26

Enter either of the following:

Node name CMEP1___ Blank or partial name followed by * for full-list
Group name _________ Enter * or partial name followed by * for full-list

Select one of the following:

24 MICROCODE
26 SOFTWARE
28 PROCEDURE
29 FLAT DATA
30 RELATIONAL DATA
31 AS/400 OBJECT
32 AS/400 CTN
33 CONFIGURATION FILE
34 MSDUMP
35 TRACE INFO
36 MERRORLOG

Tracking information:
Activity: ASSIGN

PF 1=HELP 2=SPLIT 3=END 4=RETURN
9=SWAP 10=PRINT 12=CURSOR

Figure 31. Example of a Specify Node and Select Resource Type panel

NDMPRBXE \ ASSIGN RESOURCES TO NODES (CMEP SOFTWARE)
Command ==> 

Enter either of the following:

Resource name
==> SOFTW.1234.AAA.BBBBB.CCCC
Blank or partial name followed by * for full-list
Group name . . . _________ Enter * or partial name followed by * for full-list

Automatic definition 2 (1 = Yes, 2 = No)

Partially defined Name 2 (1 = Yes, 2 = No)
Ignored for group name

Tracking information:
Activity: ASSIGN
Node . : CMEP1

PF 1=HELP 2=SPLIT 3=END 4=RETURN
9=SWAP 10=PRINT 12=CURSOR

Figure 32. Example of an Assign Resources to Nodes panel
Changing a Resource Definition

You can change the attributes of a resource that is defined in the DRD. Follow the instructions given below.

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type that the resource is associated with in the Node type field.
2. Select Option 1, Define Resources, and press Enter.

   The Define Resource panel is displayed.
3. Select one of the resource types displayed by typing the corresponding number in the selection field.

   The Specify Operation on Resource panel is displayed.
4. Specify the name of the resource you want to change. There are two ways in which you can do this:
   a. If you know the name of the resource, type the name of the existing resource definition in the Resource name field. Select Option 2, Change, and press Enter.
   b. If you do not know the name of the resource, select the resource from a full-list. Leave the Resource name field empty, or type in a partial name followed by an asterisk (*). Leave the selection field blank, and press Enter. This displays the List of Defined Resources full-list panel. Use the S (Select) line command to select the resource that you want to change, and press Enter.

   The Specify Attributes for Resource panel is displayed.
5. Change the values on this panel, as required.
6. If you specify 1, which means Yes, in the Automatic Assignment field, the Specify Node panel is displayed. This panel is described in “Step 5 (optional): Automatically Assign the Resource to a Node” on page 74.
7. Press Enter when you have finished. The new resource definition is saved in the DRD and GIX returns you to the Specify Operation on Resource panel. You can then define another resource, or press PF4 to return to the GIX main menu.

Deleting a Resource Definition

Follow the instructions below to delete an existing resource definition from the DRD:

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type that the resource is associated with.
2. Select Option 1, Define Resources.

   The Define Resource panel is displayed.
3. Select one of the resource types displayed by typing the corresponding number in the selection field.
4. Specify the name of the resource that you want to delete. There are two ways in which you can do this:
   a. If you know the name of the resource, type the name of the existing resource definition in the Resource name field. Select Option 3, Delete, and press Enter.
b. If you do not know the name of the resource, select the resource from a full-list. Leave the Resource name field empty, or type in a partial name followed by an asterisk (*). Leave the selection field blank, and press Enter. This displays the List of Defined Resources full-list panel. Use the D (Delete) line command to select the resource that you want to delete, and press Enter.

The Confirm the Deletion of Resource panel is displayed.

5. Type the primary command D (Delete) in the command field, then press Enter to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel, or PF4 to return to the GiX main menu.

**Browsing Resource Definitions**

You can look at the attributes defined in the DRD for a specific resource, or for all of the resources defined in the DRD. Follow the instructions given below.

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type that the resource is associated with.

2. Select **Option 1, Define Resources**.

   The Define Resource panel is displayed.

3. Select one of the resource types displayed by typing the corresponding number in the selection field.

   The Specify Operation On Resource panel is displayed.

4. Type the name of the existing resource in the Resource name field, or you can select the resource from the List of Defined Resources full-list panel. If you want to browse all definitions, leave the Resource name field blank.

5. Select **Option 4, Browse**, if you want to browse a specific resource definition. Select **Option 6, Browse-All**, if you want to browse all resource definitions.

   The Browse Selected Information panel is displayed. This panel displays information for each resource defined in the DRD. Use the Scroll field if there is more information than is currently displayed on the screen. The panel shows the following type of information, depending on the resource type:

   - Node type
   - Name of the resource
   - Name of the node
   - Authorization class of the resource
   - Whether the resource is restricted or not
   - Device type

6. Press PF3 when you have finished browsing, to return to the previous panel.

**Printing Resource Definitions**

You can print a definition of a specific resource, or all the resources in the DRD. Follow the instructions given below.

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type that the resource is associated with.

2. Select **Option 1, Define Resources**.

   The Define Resource panel is displayed.
3. Select one of the resource types displayed by typing the corresponding number in the selection field.

    The Specify Operation on Resource panel is displayed.

4. Type in the name of the existing resource in the Resource name field, or select the resource from the List of Defined Resources full-list panel. If you want to print all definitions, leave the Resource name field blank.

5. Select Option 5, Print, if you want to print a specific resource definition. Select Option 7, Print-All, if you want to print all resource definitions.

6. Press Enter, and you will receive a message like this:

    OPERATION COMPLETED. ENTER A NEW REQUEST.

    When you log off from GIX, you can choose to send this job to the printer. Refer to “How to Print Using GIX” on page 15.
Organizing Resources into Groups

You can organize resources into groups, in the same way as you organized nodes into groups in Chapter 3, “Defining Nodes” on page 33. When you create a group of resources you identify it by assigning it a **group name**. As with node groups, you can group resources together when you want to perform the same task on all resources. You can specify this group name in your transmission plan, instead of specifying each resource name.

The following resource types can be grouped:

- AS/400 data object containment structure
- AS/400 object
- CLIST
- Configuration file
- Dataset
- Dump (CMEP)
- Error log (CMEP)
- Flat data
- Member
- Microcode and Microcode customization
- Panel
- Print
- Procedure
- Program
- Relational data
- Software
- Trace

The following considerations apply:

- A group of resources defined for an LU0 node type can also be used in operations with user-defined nodes with NDMT functional capabilities.
- A group of resources defined for an LU 6.2 node type can only be used by LU 6.2 nodes, and not by LU0 node types.

Defining a Resource Group Using GIX

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

**Step 1: Define the Node Type**

1. Select **Option 7, Manage Groups**, from the GIX main menu.
   The Manage Groups panel is displayed.

2. Type the node type, choosing from the list of options displayed, or accept the value shown.

3. Select **Option 2, Define Groups of Resources**.
   The Select Resource Type For Grouping node type panel is displayed, shown in Figure 33 on page 80. The value node type is a variable that indicates the type of node that you are working with.

The data you can enter on the Manage Groups panel is described below.

**Node type:** This defines the type of node. Table 70 on page 561 and Table 71 on page 562 show the possible node types.
Step 2: Define the Resource Type

Figure 33. Example of a Select Resource Type For Grouping CMEP panel. This example is defining a group of software resources for a CMEP node type.

1. Select the number of the resource type, choosing from the list of options displayed, and press Enter.

The Specify Group and Select Operation panel is displayed.

The data you can enter on the Select Resource Type For Grouping panel is described below.

Resource type: GIX displays only those resource types that you can use with this node type. Table 3 on page 63 shows the resource types you can define.
Step 3: Define a Group Name

<table>
<thead>
<tr>
<th>NDMPQACR SPECIFY GROUP AND SELECT OPERATION (CMPE SOFTWARE)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Selection ==&gt;</strong></td>
</tr>
<tr>
<td>Enter:</td>
</tr>
<tr>
<td>Group name ________ Blank or partial name followed by * for full-list Ignored for BROWSE-ALL and PRINT-ALL options</td>
</tr>
<tr>
<td>Group model ________ Allowed for the CREATE option only Enter * to have all defined resources as a group model</td>
</tr>
<tr>
<td>Group creation criteria 1 = Inclusion 2 = Exclusion Allowed only for CREATE option and when group model is specified</td>
</tr>
<tr>
<td>Select one of the following:</td>
</tr>
<tr>
<td>1 CREATE Create a group</td>
</tr>
<tr>
<td>2 CHANGE Change one or more group items</td>
</tr>
<tr>
<td>3 DELETE Delete a group</td>
</tr>
<tr>
<td>4 BROWSE-ALL Browse all resource groups items</td>
</tr>
<tr>
<td>5 PRINT-ALL Print all resource groups items</td>
</tr>
<tr>
<td>PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR</td>
</tr>
</tbody>
</table>

Figure 34. Example of a Specify Group and Select Operation panel

1. Type a symbolic group name in the **Group name** field.

2. You can optionally enter the name of a group model in the **Group model** field, to use an existing group definition as a model for the one you are creating.

   **Note:** If this is the first time you are creating a resource group, it can be useful to type * in this field, as this includes all defined resources in the group. You can then delete the ones you do not want.

3. If you specified a group model, you can specify whether you want to delete all the resources that you want excluded from the group, or select all the resources that you want to include in the group. Enter a 2 in the **Group creation criteria** field for the exclusion option. This is the default value. Enter a 1 in this field for the inclusion option.

4. Select **Option 1, Create**, to create a group.

   The Create/Change a Group panel is displayed.

The fields displayed on the Specify Group and Select Operation panel are described below.

**Group name:** Enter a symbolic name for the group. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, &, and _ (underscore). The first character must be alphabetic or special. Group names must be unique across resource groups of different types.

**Group model:** This is the name of an existing node group definition. The system displays the attributes defined for this definition, and you can overwrite them for the new definition. You can specify * to use all of the defined resources as a group model, and then delete the ones you do not want.
Changing a Resource Group (System Administrator)

Step 4: Insert Resource Names into the Group

1. Use the following line commands to edit the group definition:
   - Use the I (Insert) or R (Replicate) commands to add new lines to the full-list. Once you have created a new line on the full-list, type in the name of the resource you want to include in the group.
   - Use the D (Delete) command to delete lines from the full-list.

2. Type SAVE in the command field when you have finished editing the group, and press Enter to save the new group in the DRD. To quit without saving the new group, press PF3 to return to the previous panel.

When you are editing the group definition, follow the guidelines described below.

**Resource groups:** A resource can belong to more than one group. You can include resources that are not yet defined in the DRD, but they will be ignored in your transmission plans.

**Note:** The names of resources to be included in a resource group must be fully qualified. Only the use of NETLU and NETID tokens is allowed, as these are automatically resolved at transmission time. The logical unit names and network IDs of the target nodes are dynamically substituted, providing that BYDESTID=YES is coded in the function statement. For information about how to include function statements in transmission plans, refer to Chapter 7, “Preparing and Submitting Transmission Plans” on page 119.

**Changing a Resource Group**

You can use GIX to change an existing resource group definition, providing the order of the resources in the group is not significant. Do not change an existing group definition, if the order of resources in the group has to be preserved, for example, if the group is to be used as a corequisite in an Install function (see “INSTALL MICROCODE” on page 359 and “INSTALL RESOURCE” on page 364). Instead, you should delete the existing group and define a new group (see “Deleting a Group of Resources” on page 83 for information about how to delete a resource group).
Follow the instructions below to change an existing group definition:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown.

2. Select **Option 2, Define Groups of Resources**.
   The Select Resource Type for Grouping node type panel is displayed.

3. Select the number of the resource type, choosing from the list of options displayed, and press Enter.
   The Specify Group and Select Operation panel is displayed.

4. Select the name of the group. There are two ways in which you can select the group name:
   a. If you know the name of the group you want to change, type the group name in the **Group name** field. Select **Option 2, Change**, to change one or more resource group items, and press Enter.
   b. If you do not know the name of the group you want to change, leave the **Group name** field empty, or type in a partial name followed by *, and press Enter. This displays the Create/Change Groups full-list panel. Use the S (Select) line command to select the group or groups, and press Enter.

The Create/Change Resource Groups panel is displayed.

5. Follow the instructions described in “Step 4: Insert Resource Names into the Group” on page 82, to change items in the group.

### Deleting a Group of Resources

Follow the instructions below to delete an existing resource group definition:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown.

2. Select **Option 2, Define Groups of Resources**.
   The Select Resource Type for Grouping node type panel is displayed.

3. Select the number of the resource type, choosing from the list of options displayed, and press Enter.
   The Specify Group and Select Operation panel is displayed.

4. Type the group name of the group you want to delete in the **Group name** field, or use the Create/Change Groups full-list panel.

5. Select **Option 3, Delete**, to delete a group of resources, and press Enter.
   The Confirm Deletion of Resource Group panel is displayed.

6. Type the primary command D (Delete) in the command field, to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel.

### Browsing Resource Groups

Follow the instructions below to browse the definitions of all resource groups defined in the DRD:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown.
2. Select **Option 2, Define Groups of Resources.**
   
   The Select Resource Type for Grouping node type panel is displayed.

3. Select the number of the resource type, choosing from the list of options displayed, and press Enter.
   
   The Specify Group and Select Operation panel is displayed.

4. Select **Option 4, Browse-All**, to look at all the resource groups that are defined in the DRD, and press Enter.
   
   The Browse Selected Information panel is displayed.
   
   This panel shows information about each resource group that is defined in the DRD. For each resource group, the panel displays the group name, group owner (the user ID of the person who created the group), the node type, and the name of each resource in the group. Use the **Scroll** field if there is more information than is currently displayed on the screen.

5. Press PF3 when you have finished browsing, to return to the previous panel.

---

**Printing Resource Groups**

Follow the instructions below to print the definitions of all resource groups that are defined in the DRD:

1. On the Manage Groups panel (shown in Figure 23 on page 56), type the node type, choosing from the list of options displayed, or accept the value shown.

2. Select **Option 2, Define Groups of Resources.**
   
   The Select Resource Type for Grouping node type panel is displayed.

3. Select the number of the resource type, choosing from the list of options displayed, and press Enter.
   
   The Specify Group and Select Operation panel is displayed.

4. Select **Option 5, Print-All**, to print all of the definitions of resources groups that are in the DRD, and press Enter. You will receive a message like this:
   
   **OPERATION COMPLETED. ENTER A NEW REQUEST.**

   When you log off from GIIX, you can choose to send this job to the printer. Refer to "How to Print Using GIIX" on page 15.
Chapter 5. Assigning Resources to Nodes

Assigning resources to nodes is the final step in the following sequence of tasks after NetView DM for MVS has been installed:

Step 1. Define profiles for all NetView DM for MVS users
Step 2. Define each node
Step 3. Define each resource
Step 4. Assign resources to nodes

This chapter explains how to assign resources to nodes using GIX. After you create resource assignments, you can also:

- Change the attributes of an assigned resource
- Delete a resource assignment

What Is a Resource Assignment?

Assigning a resource to a particular node, or group of nodes, is a security measure. A resource that is assigned to a particular node can only be transmitted to that node, or deleted from that node. Resources that are assigned to nodes in this way are known as restricted resources. You specify whether or not a resource is restricted when you define the resource. After you have defined the resource, you have to assign the resource to a specific node, or group of nodes. You can also assign groups of resources to groups of nodes.

The following pages show you how to assign a resource to a node.

Assigning Resources Using GIX

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and then follow these instructions:

Step 1: Define the Node Type
1. Select Option 1, Configure Network, from the GIX main menu.
   The Configure Network panel is displayed.
2. Type the node type to which you want to assign a resource. Select from the allowed entries, or accept the one shown. Leave the node type field empty if you want to assign a resource to a node group with different LU 6.2 node types.
3. Select Option 2, Assign Resources to Nodes.
   The Specify Resource Assignment Operation panel is displayed.

The data that you can enter on the Configure Network panel is described below.

Node type: This defines the type of node to which you want to assign the resource. Table 70 on page 561 and Table 71 on page 562 show the possible node types you can define, depending on your installation.
Step 2: Specify the Resource Assignment Operation

Select Option 1, Assign Resources.

The Specify Node and Select Resource Type panel is displayed.

Step 3: Specify the Node and Select the Resource Type

Select Option 2, Specify Node and Select Resource Type.
1. Select the resource type of the resource you are assigning, by typing the corresponding number on the selection line.

2. Specify either the name of a node, or the name of a group of nodes. There are two ways in which you can do this:
   
a. If you know the name of the node, or node group, type the name in either the **Node name** field, or the **Group name** field.
   
b. If you do not know the name, you can select it from a full-list panel. Enter a * or a partial name in either the **Node name** or the **Group name** field, and press Enter. Either the Select the Node panel, or the Select the Node Group panel is displayed.

   If you left both fields blank, the Select the Node panel is displayed. If you are assigning resources to node groups with different LU 6.2 node types, the **Select the Node Group** panel is displayed.

   Use the **S** (Select) line command to select the required node, or group of nodes.

   The **Assign Resources to Nodes** panel is displayed.

The data that you can enter on the Specify Resource Assignment panel is described below.

**Node name:** This is the name of the node to which you want to assign the resource. This is the name at the host that you specified when you defined the node. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore).

**Group name:** This is the name of the group of nodes to which you want to assign the resource. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore). The first character must be alphabetic or special.

**Resource type:** GIX displays only those resource types that you can assign to this node type. Table 3 on page 63 shows the resource types you can define.
Step 4: Assign Resources to Nodes

1. Specify the name of the resource, or group of resources, that you want to assign. You can either type the complete name of the resource, or resource group, in either the Resource name or Group name field. Or, you can select the name from a full-list panel.

2. Specify whether you want to automatically define the resource. If the resource is not already defined in the DRD, you can specify 1, which means Yes, in the Automatic definition field. NetView DM for MVS automatically defines the resource when you assign it.

3. Specify whether the resource name you entered is partially defined or not. Enter 1, which means Yes, or 2, which means No, in the Partially defined Name field.

4. Press Enter when you have finished. You will receive a message like this:

   OPERATION COMPLETED. ENTER A NEW REQUEST.

   You can now either assign another resource, or group of resources, or press PF4 to return to the GIX main menu.

The fields displayed on the Assign Resources to Nodes panel are described below.

**Resource name:** This is the name that you gave to the resource when you defined it. The name can have up to 44 characters, including periods. The periods (.) are delimiters between qualifiers. Each qualifier can have from 1 to 8 characters; the first character must be alphabetic. Refer to Appendix C, “Resource Naming Conventions” on page 569 for more information on the content of each qualifier.

**Note:** The resource name must be unique within the resource type. You cannot define two resources, which are of the same resource type, with the same name.
Changing the Attributes of a Resource Assignment (System Administrator)

**Group name:** This is the name which you gave to the resource group when you defined it. You can specify from 1 to 8 characters, either alphabetic, numeric, or the special characters %, &, $, #, @, and _ (underscore). The first character must be alphabetic or special.

**Note:** Group names must be unique across resource groups of different types. Do not give the same group name to resource groups of different types.

**Automatic definition:** If the resource you want to define is not already defined in the DRD, specify 1, which means Yes, in this field. NetView DM for MVS defines the resource as you assign it.

**Partially defined name:** This field is displayed for LU 6.2 resource types only. The field indicates whether or not the resource name you enter is a partially defined name, or a fully qualified name. You use a percent sign (%) to represent partially defined tokens in a resource name. Partially defined names are explained in “Partial Naming and Matching for LU 6.2 Resource Types” on page 67.

### Changing the Attributes of a Resource Assignment

**Note:** You cannot change the attributes of resource assignments for user-defined node types (nodes with CMEP, CMFP, or NDMT functional capabilities).

Follow the instructions below to change the attributes of an existing resource assignment in the DRD:

1. From the Configure Network panel (shown in Figure 14 on page 35), type the node type to which the resource is assigned. Select from the allowed entries, or accept the one shown.
2. Select **Option 2, Assign Resources to Nodes.**
   The Specify Resource Assignment Operation panel is displayed.
   The Specify Node and Select Resource Type panel is displayed.
4. Type the node name of the node that the resource is assigned to in the **Node name** field.
   **Note:** You cannot specify a group of nodes here.
5. Select the resource type of the resource, by typing the corresponding number in the selection field, and pressing Enter.
   The Specify Resource Name panel is displayed.
6. Enter the resource name of the resource.
   The Change Resource Attributes panel is displayed.
7. Type in the new values and press Enter when you have finished. You are returned to the Specify Resource Name panel and you receive a message like this:
   
   "OPERATION COMPLETED. ENTER A NEW REQUEST."
Deleting a Resource Assignment

Follow the instructions below to delete an existing resource assignment from the DRD:

1. From the Configure Network panel (shown in Figure 14 on page 35), type the node type that the resource is assigned to. Select from the allowed entries, or accept the one shown. Leave the Node Type field empty to choose resource assignments for node groups with different LU 6.2 node types.

2. Select Option 2, Assign Resources to Nodes.

   The Specify Resource Assignment Operation panel is displayed.


   The Specify Node and Select Resource Type panel is displayed.

4. Either type the node name of the node that the resource is assigned to in the Node name field, or type the group name of the group of nodes that you are assigning the resource to in the Group name field.

5. Select the resource type of the resource assignment you want to delete, by typing the corresponding number in the selection field, and press Enter.

   The Delete Resource Assignment panel is displayed.

6. Enter the resource name of the assigned resource, or the group name.

   The Confirm the Deletion of Resource Assignment panel is displayed.

7. Type the primary command D (Delete) in the command field, and press Enter to confirm the deletion. To quit without deleting, press PF3 to return to the previous panel.

Tracking Resources

For some resource types, shown in Table 3 on page 63, you can create a resource history record when you define and assign a resource. These records are stored in the DRD, and record date and time stamps when resources are assigned, stored, deleted, installed, accepted, removed, backed up, uninstalled, executed and cleared, depending on the resource type. For LU 6.2 resource types, you can change the default tracking value at installation time with the NDMRES customization macro.

The operations that are recorded vary according to the resource type. The Status field of Table 5 on page 91 shows which operations are recorded. The time and date are recorded for the following operations:

- **Ac** When a resource was last accepted.
- **As** When a resource was last assigned. This status applies only to assignable resources, as defined at installation time.
- **B** The last backup log.
- **C** The last time resources were cleared.
- **D** The last deletion of a resource.
- **E** The last execution of a CLIST or procedure resource type.
- **I** The last installation of a resource.
- **R** The last time a resource was removed.
- **S** The last time a resource was stored.
- **U** The last time a resource was uninstalled.
<table>
<thead>
<tr>
<th>Resource type</th>
<th>Name at Node</th>
<th>Version or Modific.</th>
<th>Extent</th>
<th>Subcat ID</th>
<th>Key ID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS/400 object (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Sub)category</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/C/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Sub)category update</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/E/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration file</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>Yes</td>
<td></td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dump (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flat data</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library/sublibrary</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microcode</td>
<td></td>
<td></td>
<td></td>
<td>Ac/As/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panel</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td></td>
<td></td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D/E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTF</td>
<td></td>
<td></td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational data structure</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Resource history information for resource types

NetView DM for MVS updates these records whenever a function is successfully completed, if that function implies a change in the status of the resource. This means whenever:

- A resource is successfully assigned to a node.
- A resource is successfully sent, deleted, cleared, executed, installed, accepted, or removed.
- A resource is successfully deleted, reinstalled, or backed up as a side effect of an accept, remove, or install removably function.
- A component is successfully uninstalled from a node. In this case, the node will report all of the associated resource names and NetView DM for MVS will mark these resources as *uninstalled*.

These records are also updated when NetView DM for MVS receives unsolicited reports from nodes for functions that are executed locally.
Defining a Resource History Record Using GIX

NetView DM for MVS automatically creates and updates resource history records at transmission phase completion. However, GIX also gives you the ability to create, modify, or delete these records. You can only do this for those resource types that are trackable by their default value, and not for those resource types for which tracking is specified at phase time.

Use the instructions on the following pages at your terminal. Begin by logging on to the GIX main menu, and follow these instructions:

Step 1: Specify the Node and the Resource
1. Select Option 1, Configure Network, from the GIX main menu.
   The Configure Network panel is displayed.
2. Type the node type of the node that the resource belongs to in the Node type field.
3. Select Option 3, Maintain Resource History at the Nodes.
   The Specify Node and Select Resource Type panel is displayed.
4. Type the node name of the node. This is the name of the node to which the resource you want to create a record for is assigned.
5. Select the resource type from the list shown, and press Enter.
   The Specify Resource and Select Operation panel is displayed.
6. Type the resource name of the resource that you are creating a record for.
7. Select Option 1, Create, to create a new resource history record.
   The Create Resource History at the Node panel is displayed.

Step 2: Create the Resource History Record
The panels that appear here depend for their appearance, content, and options on the node type related to the resource.
Defining a Resource History Record Using GIX (System Administrator)

```
NDMPFXPE CREATE RESOURCE HISTORY AT THE NODE (CMEP SOFTWARE)
Command ==>

Enter desired values:

1 Name at node ==> <==
2 Deletion . . . Date and time (YY/MM/DD HH:MM)
3 Sending
4 Acceptance . . Date and time (YY/MM/DD HH:MM)
5 Removal . . . Date and time (YY/MM/DD HH:MM)
6 Back up . . . Date and time (YY/MM/DD HH:MM)
7 Assignment . . Date and time (YY/MM/DD HH:MM)
8 Uninstallation . Date and time (YY/MM/DD HH:MM)

Tracking information:
Activity . : CREATE
Resource name:

PF 1=HELP 2=SPLIT 3=END 4=RETURN

Figure 39. Example of a Create Resource History at the Node panel

After you enter the values on this panel, another panel appears where you can enter further options.

1. Enter the values that you want to record. The panel may show any of these fields:
   - Name at Node
   - Assignment
   - Installation
   - Deletion
   - Execution
   - Clearance
   - Sending
   - Acceptance
   - Removal
   - Back up
   - Uninstallation

2. When you have finished, press Enter and you are returned to the Specify Resource and Select Operation panel. You receive a message like this:

   OPERATION COMPLETED. ENTER NEW REQUEST.

The Create Resource History at the Nodes panel and the subsequent panel may display any of the following fields, depending on the resource type.

**Name at node:** This is the name of the resource at the node, if it is different to the resource name at the host.

Refer to “Naming LU 0 Resources at the Node” on page 570 for information about how to name LU0 resources at the node.
Changing a Resource History Record (System Administrator)

**Installation:** Enter the date and time that the send operation was completed. The format for the date and time is **YY/MM/DD HH:MM**.

**Deletion:** Enter the date and time that the resource was deleted at the node. The format for the date and time is **YY/MM/DD HH:MM**.

**Sending:** Enter the date and time that the resource was sent to the node. The format for the date and time is **YY/MM/DD HH:MM**.

**Acceptance:** Enter the date and time that the resource was accepted at the node. The format for the date and time is **YY/MM/DD HH:MM**.

**Removal:** Enter the date and time that the resource was removed at the node. The format for the date and time is **YY/MM/DD HH:MM**.

**Backup:** Enter the date and time of the last backup log. The format for the date and time is **YY/MM/DD HH:MM**.

**Assignment:** Enter the date and time that the resource was assigned to the node. The format for the date and time is **YY/MM/DD HH:MM**.

**Uninstallation:** Enter the date and time that the resource was last uninstalled. The format for the date and time is **YY/MM/DD HH:MM**.

**Execution:** Enter the date and time that the resource was executed. The format for the date and time is **YY/MM/DD HH:MM**.

**Clearance:** Enter the date and time that the resource was executed and then cleared. The format for the date and time is **YY/MM/DD HH:MM**.

### Changing a Resource History Record

Follow the instructions below to change an existing resource history record.

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node that the resource belongs to in the **Node type** field.
2. Select **Option 3, Maintain Resource History at the Nodes**.
   The Specify Node and Select Resource Type panel is displayed.
3. Type the node name of the node. This is the name of the node to which the resource is assigned.
4. Select the resource type from the list shown, and press Enter.
   The Specify Resource and Select Operation panel is displayed.
5. Type the resource name of the resource.
6. Select **Option 2, Change**, to change an existing resource history record.
   The Change Resource History at the Node panel is displayed.
7. Change the values on this panel, as required. Refer to “Step 2: Create the Resource History Record” on page 92 for a description of the fields. You must have authorization to change resource history records defined in your user profile.
Deleting a Resource History Record

Follow the instructions below to delete a resource history record.

1. On the Configure Network panel (shown in Figure 14 on page 35), type the node type of the node that the resource belongs to in the **Node type** field.

2. Select **Option 3, Maintain Resource History at the Nodes**.
   
The Specify Node and Select Resource Type panel is displayed.

3. Type the node name of the node. This is the name of the node to which the resource is assigned.

4. Select the resource type from the list shown, and press Enter.
   
The Specify Resource and Select Operation panel is displayed.

5. Type the resource name of the resource, or, if you want a resource full-list, leave the **Resource name** field blank.

6. If you typed in the resource name, select **Option 3, Delete**, to delete a resource history record. If you asked for a full-list, press Enter and type **D** (Delete) in the command field of the displayed panel.
   
The Confirm Deletion of Resource History panel is displayed.

7. Type the primary command **D** (Delete) in the command field to confirm the deletion, and press Enter. If you want to quit without deleting, press PF3 to return to the previous panel.

8. When you have finished, press Enter. You are returned to the Specify Resource and Select Operation panel, and you receive a message like this:
   
   **OPERATION COMPLETED. ENTER A NEW REQUEST.**
Chapter 6. Preparing Resources for Transmission

The system administrator has performed a series of tasks to set up NetView DM for MVS after installation. You can now prepare resources defined in the DRD for transmission. Preparing resources involves loading the resources from external data sets into the resource repository. Loading resources is the first step in the following sequence of tasks after your administrator has set up NetView DM for MVS:

Step 1. Load resources into the resource repository
Step 2. Prepare and submit a transmission plan
Step 3. Work on submitted plans
Step 4. Unload resources from the resource repository

This chapter begins by showing you how to browse the network configuration defined by your system administrator, which shows you the resources that you can load into the resource repository. The chapter then explains how to load the resources using GIX.

Browsing the Network Configuration

Before you begin to load resources into the repository for transmission, it may be useful to look at the nodes and resources that your system administrator has defined to the DRD. These are the nodes and resources that you can work with, referred to as the network configuration.

To browse the network configuration using GIX, you use Option 2, Browse Network, from the GIX main menu. There are various ways in which you can specify the information you want to display. The instructions on the following pages provide you with some examples.
The Browse Network Panel

1. Select Option 2, Browse Network, from the GIX main menu.

The Browse Network panel is displayed.

```
NDMPG2AE BROWSE NETWORK
Selection ==> /zerodot9:/zerodot5
Enter only one of the following:

Node name . . To have the history of assigned resources of the specified node
   (Blank or partial name followed by * for full-list)
Group node name To have the history of assigned resources to nodes belonging to the specified group
Node type . . CMEP To have the history of resources belonging to nodes of specified type
   Allowed entries: CMEP CMFP SERV CLNT AS40

Select one of the following:
1 BROWSE         Browse node with all resource history
2 PRINT           Print node with all resource history
3 FULL-LIST OF RESOURCES Get a list of assigned/tracked resources
4 SELECT          Select resource type and resource name

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 1/zerodot=PRINT 12=CURSOR
```

Figure 40. Example of a Browse Network panel

There are two ways in which you can use this panel:

- Leave all the fields empty, or type *, or a partial name followed by *, in the Node name field. Leave the selection field blank, and press Enter.

   The List of Nodes full-list panel is displayed. On this panel, you can:
   - Use the B (Browse) or P (Print) commands to browse or print all the resources associated with a node. Go to “The Browse Selected Information panel” on page 99.
   - Use the F (Full-list) command to display the List of Resources full-list panel. Go to “The List of Resources Panel” on page 99.
   - Use the S (Select) command to display the Select Resource Type panel. Go to “The Select Resource Type Panel” on page 99.

- Or, you can type a specific name in one of the three node fields. Select one of options 1 through 4 from the Browse Network panel, and press Enter. The options are:

  **Option 1** You can browse nodes and their attributes, and the resource history at the node. Go to “The Browse Selected Information panel” on page 99.

  **Option 2** You can print all resources that have been assigned.

  **Option 3** You can display the List of Resources full-list panel. Go to “The List of Resources Panel” on page 99.

  **Option 4** You can display the Select Resource Type panel. Go to “The Select Resource Type Panel” on page 99.
The Select Resource Type Panel
The Select Resource Type panel list the types of resource that are valid for the type of node you selected. Follow these instructions to select the resource you want to view:

1. Type a number that corresponds to one of the resource types in the selection field and press Enter. The Specify Resource Name panel is displayed. You can use this panel in two ways:

   a. If you know the name of the resource, type the name in the Resource name field. Select Option 1 or Option 2, and press Enter.

      The Browse Selected Information panel is displayed (go to “The Browse Selected Information panel”), or printed.

   b. If you do not know the resource name, leave the Resource Name field empty, or type in *, or a partial name followed by *. Select Option 3 and press Enter.

      The List of Selected Resources panel is displayed. Go to “The List of Resources Panel.”

The List of Resources Panel
This panel lists all resources that are assigned and tracked to the node, node type, or node group that you selected.

1. Type the B (Browse) or P (Print) line commands next to the resource that you want to browse or print, and press Enter.

   The Browse Selected Information panel is displayed (go to “The Browse Selected Information panel”), or printed.

The Browse Selected Information panel
Whatever route you took, the Browse Selected Information panel is the panel that displays the DRD information for nodes and resources. Figure 41 on page 100 shows an example of the type of information displayed on the Browse Selected Information panel.
Loading Resources into the Repository

You have to load resources from an external data set into the resource repository before you can transmit them in a transmission plan. The resource repository is a set of VSAM files which store data resources and software resources.

The resource repository consists of four distinct data sets; the holding file directory and data space, and the NetView DM for MVS library directory and data space. The data spaces contain the actual resources. The directories contain information about how the resource was stored, and about how it should be processed. Each resource type is held either in the holding file, or in the NetView DM for MVS library file. For information about where each resource type is stored, refer to Table 74 on page 566.

The type of data that you are loading into the repository (for example, a job, data set, or print), is stored in a physical sequential (PS) file. Physical sequential files can contain either fixed-length or variable-length records. There are some resource types that you can load into the repository in variable length format.
These are:

- All LU 6.2 resource types
- CLIST
- Job
- Data set
- Print

**Note:** At present, only VSE and user-defined node types can use variable-length records.

### Variable-Length Records

Physical sequential files have a parameter called RECFM which specifies whether the record format is variable (RECFM=V or RECFM=VB), or not (RECFM=F). VSAM files can be organized either as a *key sequenced data set* (KSDS), or as an *entry sequenced data set* (ESDS).

You specify whether or not the file is a variable-length record in the **Variable record length** field of the Specify Attributes panel for selected resources (for example, CLISTS). The values you can enter on this panel are:

- **NO** Depending on the RECFM of the input file, the following takes place:
  - RECFM=F The data is written into the repository.
  - RECFM=V or VB If the value specified in the **Logical record length** field of this panel is greater than the length of the data of the current input record, then the rest of the data part of the input record is padded with blanks up to up to the logical record length. It is then written into the repository.

- **YES** You should specify a variable-length record only if the input file has RECFM=V or RECFM=VB. If you specify it when the input file has RECFM=F, the load operation statement terminates with return code 8 and NetView DM for MVS issues an error message.

Here is an example showing the format of a variable-length record in a physical sequential file:

```
  ┌──┬──┬─────────────┐
  │LL│/zerodot/zerodot│ DATA │
  └──┴──┴─────────────┘
```

Figure 42. Format of variable-length records in user files

In a physical sequential file with record format V, each record begins with a 4-byte record descriptor word (RDW), where **LL** is a 2-byte binary number not exceeding 32 768 that represents the length of the record (including the RDW), and **00** is a 2-byte binary field that is always set to zero. If the input file is a variable-length record file, and VARLEN=YES, the PREPARE statement reads the records from the input file one after the other, including the RDW, and writes them to the repository without changing them in any way, unless you request compression.

In a physical sequential file with record format VB, the record begins with a block descriptor word (BDW) that describes how many variable-length records (logical records, each headed by its own RDW) the file contains. When the files are loaded
Uniform Text Processing (End User)

into the repository, the MAINTDAT utility unblocks the records and writes each logical record into the repository as if it was reading from a file with a record format of V.

VSAM files do not have RDWs. When you load a resource from a VSAM file and VARLEN=YES, NetView DM for MVS builds a RDW for each record, and writes the record to the repository. The RDW contains the actual record length.

The records from a physical sequential file and those from a VSAM file look exactly the same when stored in the repository. Here is an example showing the format of a variable-length record in the resource repository:

```
  ┌───┬───┬──────┬──────┐
  │ LL│/zerodot/zerodot│ data1│LL│/zerodot/zerodot│ data2│
  └───┴───┴──────┴──────┘
  │LL│/SM630000────LL────/SM590000│/SM630000────LL────/SM590000│
  │/SM630000────LL─────/SM590000│/SM630000────LL─────/SM590000│
```

Figure 43. Variable-length records in the repository

The records stored in the repository are simply a stream of continuous bytes, spanning as many repository records as needed. If you requested for the data to be compressed, both the data and the RDW are compressed.

NetView DM for MVS records the fact that the resource is made up of variable-length records, as well as the maximum record length, in the resource repository directory. When the TCP transmits the resource to the node, this information is also sent to the node.

The following pages show you how to load resources into the repository using GIX. The specific fields displayed on each panel may vary, according to the resource type that you are loading.

Uniform Text Processing

A uniform text is a text data object (resource) that contains only characters belonging to an identified graphic character set. The following NetView DM for MVS LU 6.2 resource types can contain a uniform text:

- Flat data
- Dump
- Trace
- Error log
- Procedure

These resources carry the information that tells whether their content is a uniform text. Resources that contain a uniform text must have a defined record structure. Information about the record structure (for example, whether fixed or variable, the logical record length for a fixed length record, or the maximum record length for a variable length record) are associated with the resource.
This information is stored in the directory entry associated with a resource when loading the resource into the resource repository using either a Load or Retrieve function:

**Load** Information is provided as follows:

- Information about the uniform text and record format is provided *directly* by operands of the Load function.
- Logical record length (for fixed records) is provided *indirectly* by the LRECL parameter of the input file.
- The maximum record length (for variable length records) is determined by NetView DM for MVS while loading the data, and corresponds to the longest record read.

Only one code page, identified by the Coded Character Set IDentifier (CCSID) specified when customizing NetView DM for MVS using the NDMCOM macro, is used for all the resources loaded at the host. This CCSID is also stored in the directory record associated with the resource.

All this information is sent to a target node when the resource is transferred to it.

**Retrieve** The node sends the relevant information when transferring the resource to the host. The code page of each resource being retrieved is identified by one of the CCSIDs chosen from Table 7 on page 104.

### Coded Character Set Identifiers (CCSID)

Only one host CCSID is active at NetView DM for MVS at any time. The host CCSID is selected from the set of CCSIDs shown in Table 6.

<table>
<thead>
<tr>
<th>CCSID</th>
<th>CP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0025</td>
<td>00037</td>
<td>CECP: USA, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand</td>
</tr>
<tr>
<td>0111</td>
<td>00273</td>
<td>CECP: Austria, Germany</td>
</tr>
<tr>
<td>0115</td>
<td>00277</td>
<td>CECP: Denmark, Norway</td>
</tr>
<tr>
<td>0116</td>
<td>00278</td>
<td>CECP: Finland, Sweden</td>
</tr>
<tr>
<td>0118</td>
<td>00280</td>
<td>CECP: Italy</td>
</tr>
<tr>
<td>011D</td>
<td>00285</td>
<td>CECP: United Kingdom</td>
</tr>
<tr>
<td>0129</td>
<td>00297</td>
<td>CECP: France</td>
</tr>
<tr>
<td>01F4</td>
<td>00500</td>
<td>CECP: Belgium, Canada (AS/400), Switzerland, international Latin-1</td>
</tr>
<tr>
<td>011C</td>
<td>00284</td>
<td>CECP: Spain, Latin America</td>
</tr>
<tr>
<td>0367</td>
<td>00871</td>
<td>CECP: Iceland</td>
</tr>
</tbody>
</table>

*Table 6. Possible NetView DM for MVS host CCSIDs*

When translating a uniform text received from a node workstation, or from another host, the translation process needs both the *source* CCSID and the *target* (host) CCSID. The source CCSID is received from the node with the resource. The target CCSID is the host CCSID selected by NetView DM for MVS customization (from Table 6). Only a selected set of source CCSIDs, as shown in Table 7 on page 104, can be used by NetView DM for MVS for translation.
Nevertheless, you can use CCSIDs not included in the previous tables by using your own translation table. Be sure that you:

- Define the user translation table in CDRA format.
- Store the table in a data set.
- Run the ADDTABLE utility from the customization dialog to update the index.

Also, if the target CCSID specified in the user translation table is different from the one currently running at the host, you must use the dialog to customize NetView DM for MVS again.

---

<table>
<thead>
<tr>
<th>CCSID</th>
<th>CP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>61B5</td>
<td>00437</td>
<td>PC Display; PC Base; USA, many other countries</td>
</tr>
<tr>
<td>6352</td>
<td>00850</td>
<td>PC Display; Latin-1 countries</td>
</tr>
<tr>
<td>635C</td>
<td>00860</td>
<td>PC Display; Portugal</td>
</tr>
<tr>
<td>635F</td>
<td>00863</td>
<td>PC Display; Canada</td>
</tr>
<tr>
<td>6361</td>
<td>00865</td>
<td>PC Display; Denmark, Norway</td>
</tr>
<tr>
<td>0025</td>
<td>00037</td>
<td>CECP: USA, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand</td>
</tr>
<tr>
<td>0111</td>
<td>00273</td>
<td>CECP: Austria, Germany</td>
</tr>
<tr>
<td>0115</td>
<td>00277</td>
<td>CECP: Denmark, Norway</td>
</tr>
<tr>
<td>0116</td>
<td>00278</td>
<td>CECP: Finland, Sweden</td>
</tr>
<tr>
<td>0118</td>
<td>00280</td>
<td>CECP: Italy</td>
</tr>
<tr>
<td>011D</td>
<td>00285</td>
<td>CECP: United Kingdom</td>
</tr>
<tr>
<td>0129</td>
<td>00297</td>
<td>CECP: France</td>
</tr>
<tr>
<td>01F4</td>
<td>00500</td>
<td>CECP: Belgium, Canada (AS/400), Switzerland, international Latin-1</td>
</tr>
<tr>
<td>011C</td>
<td>00284</td>
<td>CECP: Spain, Latin America</td>
</tr>
<tr>
<td>0367</td>
<td>00871</td>
<td>CECP: Iceland</td>
</tr>
</tbody>
</table>

*Table 7. Possible NetView DM for MVS source CCSIDs*
Restriction on DBCS Code Pages

NetView DM for MVS automatically handles the conversion of text between different code pages. However, NetView DM for MVS does not support double-byte character sets (DBCS). A DBCS is a set of characters in which each character is represented by two bytes. Languages such as Japanese, Chinese, and Korean require double-byte character sets.

In particular, NetView DM for MVS does not support an MVS host running on a DBCS code page and does not allow a uniform text object to be loaded. This has the following impacts when working with workstations running NetView DM/2:

- DBCS EBCDIC objects are not converted to ASCII when downloaded to a workstation running NetView DM/2.
- DBCS ASCII objects are not converted to EBCDIC when sent from NetView DM/2 to a host running NetView DM for MVS.

However, NetView DM for MVS does allow ASCII DBCS objects to be unloaded (see “Unloading or Printing a Uniform Text Resource” on page 106).

Loading a Uniform Text

When loading a uniform text resource into the repository, the record structure of the user input file is handled in the following way:

- If the user input file has a fixed record format, the logical record length of the input file is stored in the holding file directory; the resource is marked as having fixed record length.

- If the user input file has variable record format, set the VARLEN operand of the Load function to Y (Yes). Otherwise, the Load function terminates with return code 8 and an error message is issued.

The Load processor stores records being read from the input file in the data part of the resource repository, with each record headed by an RDW. The RDW conforms with the RDW used by MVS for sequential files with variable-length records. It is a 4-byte field, as follows:

- The first bytes contain the length LL of the logical record, including the 4-byte RDW.
- The length can be from 4 to 32,756, and all bits of the third and fourth byte must be 0.

The resource is marked as having variable-length records.

Note

Batch Utilities, GIX, and the D&CC API Load function can load any LU 6.2 resource type.
Unloading or Printing a Uniform Text Resource

When unloading a resource that contains a uniform text from the resource repository to a user file, or when printing such a resource, the data is translated from the CCSID associated with the resource to the NetView DM for MVS-defined CCSID.

You can also unload double-byte character set (DBCS) uniform text resources. However, no translation is performed and the resources are unloaded as they are.

You cannot unload these resources when they are compressed by a user compression algorithm, and the CCSID of the algorithm name identifies a DBCS code page. Here is a list of the DBCS CCSIDs that are supported:

<table>
<thead>
<tr>
<th>Decimal CCSID</th>
<th>Hex CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0932</td>
<td>03A4</td>
<td>COMBINED_JAPAN_CP1</td>
</tr>
<tr>
<td>0942</td>
<td>03AE</td>
<td>COMBINED_JAPAN_CP2</td>
</tr>
<tr>
<td>0934</td>
<td>63A6</td>
<td>COMBINED_KOREAN_CP1</td>
</tr>
<tr>
<td>0944</td>
<td>03B0</td>
<td>COMBINED_KOREAN_CP2</td>
</tr>
<tr>
<td>0949</td>
<td>03B5</td>
<td>COMBINED_KOREAN_CP3</td>
</tr>
<tr>
<td>0938</td>
<td>03AA</td>
<td>COMBINED_CHINESE_CP1</td>
</tr>
<tr>
<td>0948</td>
<td>03B4</td>
<td>COMBINED_CHINESE_CP2</td>
</tr>
</tbody>
</table>

Table 8. Possible DBCS CCSIDs

Locking Resources in the Resource Repository

User programs can use the NetView DM for MVS D&CC API to lock resources held in the resource repository. Once a resource is locked, you cannot delete or change the resource using GIX, the MAINTDAT Batch Utility, or the TCP. Resources can be unlocked using either the D&CC API Unlock_Catalog_Entry function, or the MAINTDAT control statement called ULOCK RESOURCE, described in "ULOCK RESOURCE" on page 311.

Refer to the Distribution and Change Control Application Programming book for information about how to lock and unlock resources using the D&CC API.
Use the instructions on the following pages at your terminal. After logging on to the GIX main menu, follow these instructions:

**Step 1: Select Resources of a Specific Type**
1. Select **Option 3, Manage Resources**, from the GIX main menu.

   The Manage Resources panel is displayed, as illustrated in Figure 44.

   ![Figure 44. Example of a Manage Resources panel](image)

   **NDMPAOAE**
   **MANAGE RESOURCES**
   **ROW 1 OF 11**
   **Scroll PAGE**

   Do you want specific resources of the type selected? 2
   1 = Yes 2 = No
   (Yes = Additional selection criteria will be displayed
   No = All resources for the selected type will be displayed)

   Select one of the following resource types:

   24 MICROCODE
   26 SOFTWARE
   28 PROCEDURE
   29 FLAT DATA
   30 RELATIONAL DATA
   31 AS/400 OBJECT
   32 AS/400 CTN
   33 CONFIGURATION FILE
   34 MSDUMP
   35 TRACE INFO
   36 MSERRLOG

   **************************** END OF DATA ****************************

   **Note:** You can use the Scroll key to display the remaining resource types that are not in view on the panel.

2. Select the resource type of the resources you want to load into the resource repository. Type the number of the resource type in the selection field. This will list all the resources of this type that are held in the repository. Press Enter and the List of Selected Resources full-list panel is displayed. Go to **“Step 3: Insert Resources into the Repository”** on page 109.

3. Optionally, if you want to use additional selection criteria to build a list, type 1 in the field which asks:

   Do you want specific resources of the type selected?

   Press Enter. The Specify Resource Selection Criteria panel is displayed.

**Step 2 (optional): Specify Resource Selection Criteria**

Use the Specify Resource Selection Criteria panel to narrow down the list of resources you want to display. This panel displays various fields, depending on the resource type you are working with. The fields represent resource identifiers, such as resource name. Resource identifiers identify resources at the host.

1. Type values (or partial values followed by *) in any or all of the fields and press Enter.

   The List of Selected Resources full-list panel is displayed.
The Specify Resource Selection Criteria panel may display any of the fields described below, depending on the resource type.

**Resource name:** This is the symbolic name assigned to the resource when it was defined.

For LU0 resources, the name you assign can have up to 44 characters, including periods. The periods (.) are delimiters between qualifiers. Each qualifier can have from 1 to 8 characters. The specific syntax rules for each resource type are described in Appendix C, “Resource Naming Conventions” on page 569.

For LU 6.2 resources, the name can have up to 10 tokens of up to 64 characters. Each token can have from 1 to 16 characters. The first token must be the corporation ID assigned to your company by IBM. The rules for naming LU 6.2 resources are described in “Naming LU 6.2 Resources” on page 65 and in Appendix C, “Resource Naming Conventions” on page 569.

**File control name:** You can group resources in the host repository under file control names to refer to a queue of resources in the holding file. To identify a resource or resource group, specify this attribute either by itself, or with any other identifier on the panel. You can specify from 1 to 8 alphanumeric characters, the first of which must be alphabetic.

If you do not specify a file control name in a transmission function, the default when the plan is executed is the name of the node, or group of nodes, that a phase addresses.

**Origin:** The origin of a resource is the name of the node from which the resource was retrieved by a transmission function. The origin can either be a specific node (from 1 to 8 alphanumeric characters, the first of which is alphabetic), or the host (DSX). The origin attribute applies only to resources stored in the holding file.

**Group name:** You can define groups of resources in the DRD. Group names can have from 1 to 8 alphanumeric characters. The first character must be alphabetic.

You can specify a combination of the four fields described above, to specify more than one resource.

**Completeness:** When you load a resource into the repository, the load operation may not always be completed successfully. This identifier allows you to specify one of three values:

0 All. List all resources.

1 Incomplete. List only those resources that were not loaded at all, or not loaded completely.

2 Complete. List only those resources that completed successfully.

**Modification:** The modification level is an integer from 0 to 99. This identifier is only used for resources stored in the NetView DM for MVS library. This operand is optional. It takes a default value of 0 from the corresponding batch statement.

**Version:** The version number is an integer from 0 to 99. This identifier is only used for resources stored in the NetView DM for MVS library. This operand is optional. It takes a default value of 0 from the corresponding batch statement.
Note:  VSE resource types may also have two other identifiers. These are described below.

**Library name:**  VSE nodes only. Specify a library name.

**Type:**  Specify the library type. You can specify one of:

- **C** Core image library
- **P** Procedure library
- **R** Relocatable library
- **S** Source library

For member resource types, you can specify from 1 to 8 alphanumeric characters.

**Note:**  If you specify a library type of **S**, you must also specify a sublibrary name in the **Sublibrary Name** field.

**Sublibrary name:**  VSE nodes only. Specify the sublibrary name.

**Note:**  If you specified a library type of **S** in the **Type** field, you must also specify a sublibrary name.

**Step 3: Insert Resources into the Repository**

<table>
<thead>
<tr>
<th>CMD</th>
<th>RESOURCE NAME</th>
<th>LOCK</th>
<th>COMPLETENESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>CORPID.MSG.JUL</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>___</td>
<td>PROVA.01</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Figure 45. Example of a List of Selected Resources panel

This panel lists the resources that are currently held in the resource repository. To load a new resource into the repository, use the **I** (Insert) primary command.

The panel also shows whether a resource was locked in the resource repository by an application program. If the resource is locked, you cannot delete or change the resource.

To insert a resource into the resource repository, follow the instructions below:

1. Enter the primary command **I** (Insert) in the command field, and press Enter.

   The Identify the Resource panel is displayed.
This panel displays various fields, depending on the resource type you are working with. The fields represent resource identifiers, such as resource name.

2. Type the name of the resource in the Resource name field.

You can ignore the Matching indicators field when loading a resource. It is only used when unloading, or converting a resource, from the repository.

3. Press Enter; a second Identify the Resource panel is displayed.

This panel displays various fields, depending on the resource type you are working with. The fields represent resource identifiers, such as resource name.

2. Type the name of the resource in the Resource name field.

You can ignore the Matching indicators field when loading a resource. It is only used when unloading, or converting a resource, from the repository.

3. Press Enter; a second Identify the Resource panel is displayed.
4. You can optionally type over the values displayed for matching attributes, generable attributes, type attributes and class codes. These fields are described below.

5. Press Enter when you have finished.

The Specify Attributes panel is displayed.

The fields shown on the Identify the Resource panels are described below.

**Resource name:** Enter the name of the resource that you want to load. The resource name must be fully qualified. You cannot use partial names to load resources. The format of resource names is described in “Naming Resources” on page 65 and also in Appendix C, “Resource Naming Conventions” on page 569.

**Matching indicators:** This field is ignored when you are loading resources. It only applies when you are unloading resources from the repository, and is described in “Step 3: Convert Resources from the Repository” on page 186.

**Match attributes:** This field specifies the list of matching attributes that define which tokens in the global name must match, and which need not match, when you use partial naming.

If you specify M, the token must match the value you enter. If you specify N, the token need not match.

**Generable attributes:** This field defines whether or not the target server can generate appropriate token values for each token in the global name.

If you specify G, the token is generable. If you specify N, the token is not generable and is the default value.

**Type attributes:** This field defines the type of attribute for each token. The type attribute indicates what each token represents, as follows:

- **UNSPECIFIED** Values assigned to the token cannot be assumed to have any numeric or alphabetic order. This implies that NetView DM for MVS cannot perform any partial matching operations on these tokens.
- **ORDEREDC** The related token is such that an alphanumeric order is established for the values. This implies that this token is eligible for a search using partial matching. This is the default value.
- **ORDEREDN** The related token is such that a decimal numeric order is established for the values. This implies that this token is eligible for a search using partial matching.
- **ORDEREDD** The related token is such that a date format order is established for the values. This implies that this token is eligible for a search using partial matching.
- **ORDEREDT** The related token is such that a time format order is established for the values. This implies that this token is eligible for a search using partial matching.

The following tokens have attributes that are defined by SNA. These tokens are checked for compatibility with the architecture.
SUBTREE: This token indicates the beginning of a subtree structure in the global name.

SYSTYPE: The token contains a value corresponding to a system type.

LUNAME: The token specifies the destination logical unit name (LUNAME).

NETID: The token specifies the destination network ID (NETID).

G00V00: The token contains a value corresponding to an MVS generic data set member.

**Class codes:** These are the level 3 and level 4 SNA/File Services (SNA/FS) data object class codes. Level 1 and level 2 codes are derived from the resource type code. You cannot specify only one value, you must specify both values, or none at all. The values you can specify are hexadecimal codes defined by SNA/FS and are listed in “SNA/File Services Class Codes” on page 584.

If the resource to be loaded is a NetView DM/2 resource containing extended attributes, the level 3 and level 4 class code values must be set to E4.
Step 4: Specify the Job Attributes

Figure 48. Example of a Specify Attributes panel. This is an example for loading flat data into the resource repository. The actual fields shown depend on the specific resource type.

On this panel, you enter information that identifies the batch job that will execute the load operation.

Type in the required values and press Enter. GIX creates the job and submits it to your operating system. You can then use ISPF/PDF to monitor the progress of the job (see “Going to ISPF from GIX” on page 15).

The fields on the Specify Attributes panel can display any of the following fields, depending on the resource type.

**Job name:** This field is mandatory for all resource types. Enter the name of the batch job that you are creating using GIX. The field is automatically filled in with the job name specified in your JCL cards. You can use the job name to track the progress of the batch job (refer to “Going to ISPF from GIX” on page 15).

**Data set name:** Enter the name of the user data set from which you want to load a resource. You can enter from 1 to 54 alphanumeric characters, including periods. This must be a physical sequential data set.

The following attributes are used to specify how and where the data set is stored at the host:

**Unit:** Enter the name of the unit on which the data set is stored at the host. You can specify any disk or tape unit name (such as 3350, 3380), or a user-defined name (such as SYSDA).

**Volume:** Enter the volume serial number that locates the resource in the host data set. You can specify from 1 to 6 alphanumeric characters.
**Label sequence number**: If the unit is a tape, this attribute identifies the sequential number of the resource on the tape. Specify any a number with from 1 to 4 digits.

**Label type**: If the unit is a tape, this attribute specifies the type of label that identifies the resource. The values can be:
- **AL**: American National Standard labels
- **AUL**: American National Standard and user labels
- **BLP**: Bypass label processing
- **LTM**: Leading tapemark
- **NL**: No labels
- **NSL**: Non-standard labels
- **SL**: Standard labels
- **SUL**: IBM standard and user labels

**Filemode number**: Enter one character for the filemode number of the file. This attribute is optional. If you do not enter a number, the file is stored on the disk at the node with a filemode number of 1.

**VSAM catalog name**: Specify the name of the catalog in which the data set is to be cataloged at the node. The name can be up to 44 alphanumeric characters, contained in up to 22 qualifiers.

The following three attributes specify what the record format of the resource should be when it is stored at the node, after it is transmitted.

**Control character**: This attribute must be specified for the Print resource type only. It specifies whether the first byte of each logical record at the node is an ANSI or SCC control character, or whether there is no control character. The default is NO.

For DPPX nodes, GIX generates the following values as input for the MAINTDAT PREPARE DATA statement. Table 9 shows the values assigned to the LBLRIND parameter, depending on the values entered from GIX for LBLRIND.

<table>
<thead>
<tr>
<th>LBLRIND</th>
<th>LBLRIND</th>
<th>LBLRIND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>No</td>
<td>LB</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>LR</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>LR</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Nothing</td>
</tr>
</tbody>
</table>

*Table 9. Values assigned to LBLRIND parameter*

**Logical block length**: The logical block size (from 2 to 32761 bytes) of the resource at the node. The default is 256 bytes.

**Logical Record Length**: The logical record length (from 2 to 32761 bytes) of the resource at the node. The length must be at least two bytes, and no greater than the physical record length.
**To be compressed:** Specify whether you want the input data stream to be compressed when it is loaded into the repository. Data compression helps to reduce data traffic on communication links. You can specify that a resource is to be compressed when it is loaded into the repository, by specifying 1 in the **To be compressed** field. Do not specify Yes for this option if the input data stream is already compressed (specified in the **Data already compressed** field.)

If you specify here that you want the data stream compressed, then you must also specify the compression algorithm to be used, with an entry in either the **FS Compression** or **User Compression Algorithm** field. If you do not make an entry for either of these, the **FS Compression** field defaults to X'01'.

The following resources can be compressed when they are loaded into the repository:

- All LU 6.2 resource types
- Data set (VSE, SSP, and DPPX)
- Dump (DPPX, SSP)
- Print (DPPX, SSP)
- Errorlog (DPPX)

**Attention**

If an LU0 resource is compressed in the repository, you must specify that it is compressed in the transmission plans.

Data sets at VSE nodes are always stored in a non-compressed format. Data sets sent from the host to a VSE node must be compressed, based on the record level. You cannot change the record size of a data set after it was defined at the host, otherwise the VSE node cannot process it. When data sets are printed or unloaded from the repository, they are decompressed.

**Data already compressed:** Specify whether or not the data set is already compressed. If the data set is compressed, you must also enter the size of the decompressed resource in the **Object size** field.

**Object size:** Specify the size of the resource when decompressed. You only have to enter this value if the resource to be loaded is already compressed (if the **Data already compress.** field is set to 1).

**Processing size:** Specify the estimated size in KB of the DASD space required on the target installation drive to install the resource. You can enter an integer from 1 to 999,999. Only specify a value for resource types that support the Install function.

**Note:** This parameter is supported by NetView DM/2 Version 2 Release 1 target workstations only. NetView DM/6000 and NetView DM for NetWare products check the disk space automatically. Specify a processing size only if the resource is to be installed on a NetView DM/2 Version 2 Release 1 workstation.

**FS compression:** Specify the SNA/FS compression technique that is used, if the data set is to be compressed. At present, the only value you can enter is X'01'. Do not enter anything in this field if you specify a value for a user compression algorithm in the **User Compression Algorithm** field.
**FS parms:** Specify the input parameters for the SNA/FS compression. At present, one hexadecimal byte represents the prime control character. This attribute is optional. You can only specify it if you also specify that you want to use SNA/FS compression in the **FS Compression** field.

**User Compression Algorithm:** Specify the name of a user-defined algorithm, from 1 to 12 characters. Do not specify this attribute if you are using an SNA/FS compression technique. This name is always written and transferred to the target node in the EBCDIC character set.

If you want to specify a compression exit based on the LZW technique (used by NetView DM/2), then enter NVDMZFW in the **User comprs algorithm** field. In this situation, do not make enter a value in the **User comprs parameters** field.

**User compression parameters:** Specify the input parameters for the user-defined compression algorithm. The value can be from 1 to 27 bytes long. This attribute is optional. You can only specify a value if you are using a user-defined algorithm. The entire character string must be enclosed within single quotation marks.

**Uniform text:** Specify whether the resource being loaded contains a uniform text. A uniform text is a text data object that contains only characters belonging to an identified character set. If you enter 1, which means Yes, the NetView DM for MVS Coded Character Set defined using the NDMCOM customization macro and the record organization (fixed or variable) of the user input file are stored in the resource directory. The Coded Character Set Identifier and record structure are transferred to the node along with the resource during the Send function.

You can specify Yes only when loading flat data, procedure, dump, error log, or trace resources. If the input data set has a record format of V or VB and you specify Yes for uniform text, you must specify 1, which means Yes, in the **Variable Record Length** field. If the **Variable Record Length** field is set to 2 or not coded at all, the Load function terminates with return code 8.

**Variable record length:** Specifies whether the resource you are loading will be stored as a variable-length record.

If you enter 2, which means No, the following applies:

- **RECFM=F**  
  The resource is loaded as a byte stream if you specified No for the **Uniform Text** field. If you specify Yes for the **Uniform Text** field, the resource is loaded as a fixed-length resource.

- **RECFM=V or VB**  
  The record description words are removed from the records, and the resource is stored as a byte-stream resource in the resource repository. You cannot specify Yes for the **Uniform Text** field.

If you enter 1, which means Yes, the following applies:

- **RECFM=F**  
  The load function terminates with return code 8.

- **RECFM=V or VB**  
  The record description words are stored with the records in the resource repository. The resource is marked as having variable-length records.
Replacing or Appending Resources

These commands are only available for certain resource types. Use the **APP** (Append) primary command on the List of Selected Resources panel to add extra data on to the end of the existing data for a resource. NetView DM for MVS creates the resource, if it does not already exist.

Use the **REP** (Replace) primary command on the List of Selected Resources panel to substitute the existing data for a resource in the resource repository with new data. The resource name and attributes remain the same.

Browsing or Printing the Repository Directory

You can browse or print all of the resources that are currently in the resource repository from the List of Selected Resources full-list panel. This panel displays a list of the resources contained in the repository. You can get more information using the **B** (Browse Directory) primary or line command, and the **P** (Print Directory) primary command. You can use these commands either as primary commands, or as line commands.

To use the **B** (Browse) primary command, follow the instructions below:

1. Enter **B** in the command field. When you press Enter, the Identify Resource panel is displayed.
2. Enter values in any of the fields on this panel. Do not use * wild cards. Press Enter. The Browse Resource Directory panel is displayed.

To use the **P** (Print Directory) primary command, enter **P** in the command field. When you press Enter, the message PRINT REQUEST IS QUEUED is displayed.

Deleting Resources from the Repository

You can delete resources from the repository using the List of Selected Resources panel, shown in Figure 45 on page 109.

**Note:** You cannot delete resources that have been locked in the resource repository.

Follow the instructions below to delete resources using this panel:

1. To delete a single resource, use the **D** (Delete) command in the command field, or next to one of the resources in the full-list. If the resource is locked, NetView DM for MVS displays an error message and the resource is not deleted.

   To delete all of the resources currently displayed on the panel, use the **DAL** (Delete all) command, in the command field only.

   Any resources that are currently locked in the resource repository are not deleted, and NetView DM for MVS displays an error message. NetView DM for MVS only deletes those resources that are not locked.

2. Press Enter. The Confirm Deletion of Resource panel is displayed.

3. Type **D** or **DAL** in the command field, and press Enter to confirm the deletion. If you want to quit without deleting the resources, press PF3 to return to the previous panel.
Printing Resources
You can use GIX to request a printout of any of the following types of resource:

- All LU 6.2 resource types
- ABEND
- CLIST
- Dataset
- Errorlog
- Job

You can do this from the List Selected Resources panel. Use the PDA (Print Data) command in the command field, or next to one of the resources in the full-list.

Querying Space in the Repository
You can query the amount of space left in the resource repository. To do this:

1. Type QSPACE in the command field on any of the GIX panels that you access after selecting Option 3, Manage Resources, from the GIX main menu.
2. If you see that you are running out of space, you can either delete some resources, or refer to “Maintaining the Resource Repository” on page 430.

Editing CLIST and Job Resources Using ISPF/PDF
You can edit CLIST and Job resources using the ISPF/PDF EDIT service. The ISPF/PDF EDIT service requires a dynamically allocated data set with logical fixed record length from 10 to 255 bytes as input.

CLIST resources
You can manage CLIST resources with a record length varying from 2 to 256 bytes. As this range is different from the ISPF/PDF EDIT range, the edit data set record is extended to 10 bytes for a CLIST resource which has 2-byte long records. The resource record is still written to the holding file with only 2 bytes.

Note: A CLIST resource with 256-byte records has only 255 significant bytes. The 256th byte cannot be edited, and so it is set to blank. A warning message is displayed. The real resource record length remains at 256 bytes. No other normalization is necessary. You can handle both variable and fixed-length CLISTs.

Job resources
A Job resource is inserted in the resource repository with a logical record length of 80, but the Job keeps its own record length when it is modified.
Chapter 7. Preparing and Submitting Transmission Plans

Preparing and submitting transmission plans is the second step in the following sequence of tasks after your administrator has set up NetView DM for MVS:

**Step 1.** Load resources into the resource repository
**Step 2.** Prepare and submit a transmission plan
**Step 3.** Work on submitted plans
**Step 4.** Unload resources from the resource repository

This chapter begins by describing the purpose and structure of transmission plans. The chapter then shows you how to use GIX to:

- Define a new transmission plan
- Validate a transmission plan
- Submit a transmission plan to the transmission control file (TCF)

You can also change an existing plan, rename a plan, delete a plan, and query the amount of space left in the plan library.

What Is a Transmission Plan?

Transmission plans contain scheduling information. A plan describes operations you want to perform on resources, and groups these operations together according to nodes.

A transmission plan consists of one or more phases. Phases contain sequences of one or more transmission functions, followed by an optional batch job.

You create a transmission plan in the NetView DM for MVS plan library. You then submit it to the transmission control program (TCP) which transmits the plan. Submitted plans are stored in the TCF. When you create a plan, you define:

- A plan name, which identifies the plan
- Whether the plan is recursive or not
- The number of hours a recursive plan can be delayed by each day
- The execution priority of the plan
- Any notes associated with the plan

**Note:**

When running more than one NetView DM for MVS application on the same system, give unique names to transmission plans from different NetView DM for
What Is a Transmission Plan? (End User)

MVS applications. This applies even if separate plan libraries are used for the applications. Using identically named transmission plans can cause problems.

Recursive Plans

A NetView DM for MVS plan can be recursive. This means that once you have submitted it to the TCF, NetView DM for MVS executes the plan every day at the scheduled time, until it is deleted from the TCF. In effect, NetView DM for MVS generates a new plan every 24 hours. The plan generated in this way is called the plan recursion for that day. You can delete a plan recursion, if transmission is not required on a certain day.

You schedule the plan when you define attributes for a phase. You also define a start-time delay. This means that if the plan cannot be executed on time, NetView DM for MVS starts the plan within the number of hours you specify in the start-time delay, or not at all.

A recursive plan can contain one or more phases. You cannot define transmission windows for phases in recursive plans.

Notes

- If a plan is recursive, the validation of the plan is performed only when the plan is submitted. If you make changes to the plan later, for example, to the password or logon values of the node, the initial values continue to be considered in the next recursions of the plan. To avoid failures in the transmission of such a plan, you should erase the plan and resubmit it.

- As you submit a recursive plan only once, any changes that you make after the plan has been submitted will not come into effect immediately, even if you are running GIX with NETCHNG=IMMEDIATE. For the changes to have effect, you must resubmit the plan.

Phases

A phase is a group of one or more functions that have the same scheduled time and date, the same priority, and address the same node or group of nodes. For example, if you want to send various resources to a single node or single group of nodes, you can group these functions together in a phase. If, however, you want to retrieve a resource from a node and then send the same resource to a different node or group of nodes, you must create two separate phases.

When you define a phase you define:

- A phase name.
- The name of the node or group of nodes the phase is addressing.
- The date and time when the phase will be executed. If the plan is recursive, all of the phases must have the same scheduled date, but they can have different scheduled times.
- A cut-off time or interval within which the phase must be completed.
- Whether the phase is to be held or not.
- Whether the processing of the phase depends on the successful execution of another phase within the same plan.
What Is a Transmission Plan? (End User)

- Whether or not tracking is to be performed for the phase.
- Whether a batch job is to be run at the end of the phase.
- A transmission window within which the phase must be executed (this does not apply to phases within recursive plans).

Within a phase you can also specify a set of logical conditions. See “Conditioning within a Phase” and “Conditioning between Phases” on page 122.

Phases that belong to a plan that is submitted with a specific priority have the same priority. The phases are scheduled and processed according to the priority of the plan and to the scheduling date and time.

If phases that belong to different plans have the same scheduling data and time, the phase that has the higher priority is processed first. If phases have the same priority, they are processed according to the scheduling date and time.

When plans are running, and a higher priority plan is submitted, the system does not interrupt the running phases. The phases belonging to the higher priority plan are processed as soon as the session that addresses the specific logical unit is available.

**Functions**

The basic level of transmission operation is the *function*. A function specifies the transmission activity to be performed on a resource or resources. The transmission functions that you can use with different resource types are automatically displayed on the appropriate GIX panels.

**Conditioning within a Phase**

Within a plan, you can make the execution of a phase or transmission function *conditional* upon the return code from some previous phase or function. When the TCP executes the functions in a plan, the functions terminate with a return code (0 to 8) which indicates whether the function was successful. Table 15 on page 180 shows the transmission function return codes. You can condition the processing of functions in a sequence by assigning a *termination value* to the functions when you create the plan. NetView DM for MVS then compares the termination value you assigned with the return code produced by the transmission.

When the termination value is 0, no termination value is applied. If you specify a value other than 0, and a function executes and returns with a return code that is not 0, greater than or equal to the value you defined, *none* of the remaining functions in the sequence will be executed.

There are four levels at which you can specify the termination value:

1. At the *phase* level. The termination value you define for the phase will apply to *all* the functions in that phase. You can override this value for individual sequences and functions within the phase (see levels 3 and 4 below).
2. At the *phase for node* level. The termination value you specify will apply to all the phases for a specific node, instead of all the phases as above.
3. At the *sequence* level. You can specify a sequence termination value, which will apply to all the functions in a particular sequence. You can override this value for individual functions in the sequence (see level 4 below).
4. At the function level. You can specify a separate function termination value for each function in the sequence. A function termination value always overrides the phase or sequence level values described above.

5. For batch jobs. You can specify a batch job to be executed at the end of each phase, and assign a termination value to the batch job so that it only executes if all the functions in phase were successfully executed, based on the assigned return code value.

Conditioning between Phases

At the phase level, but not at the sequence or functional level, you can specify:

- A conditioning phase
- A conditioning value
- Conditioning criteria

These specify whether the phase is to be executed at all. The phase that must be completed first is called the conditioning phase. The phase whose execution depends on the return code from the conditioning phase is called the conditioned phase. When you define a conditioned phase, you specify a conditioning value and conditioning criteria. The conditioning criteria for a phase can either be:

- **GREATER THAN OR EQUAL TO**: The conditioned phase is executed if the conditioning phase was unsuccessful. In this case, the return code from the conditioning phase is greater than or equal to the termination value assigned to the conditioned phase.

- **LESS THAN OR EQUAL TO**: The conditioned phase is executed if the conditioning phase was successful. In this case, the return code from the conditioning phase is less than or equal to the termination value assigned to the conditioned phase.

The following example illustrates a plan with three phases. Phases 2 and 3 are both conditioned by phase 1:

**Example**

**Phase 1**: Send some new software to a node, and execute it there.

**Phase 2**: If Phase 1 is successful, delete the previous version of the software stored at the node.

**Phase 3**: If Phase 1 is unsuccessful, retrieve diagnostic information from the node, delete the new version of the software, and send some messages.

In this example, you would enter the following values for phase 2:

- **Conditioning phase**: PHASE01
- **Conditioning criteria**: 2
- **Conditioning value**: 04

and these values for phase 3:

- **Conditioning phase**: PHASE01
- **Conditioning criteria**: 1
Browsing or Printing a Plan Using GIX

The first step in preparing a new transmission plan is to browse or print plans which already exist in the plan library, to find a plan with a structure similar to the one you want to create.

If this is the first time you are using GIX to create plans, use your default plan model. Your default plan model is stored in your user profile. Contact your system administrator if you want to change your default plan model permanently. As many of the plans you prepare will be similar, your default plan model should reflect the nodes, resource types, phases, and functions that you are likely to use.

Use the instructions on the following pages at your terminal.

Step 1: Select the Plan Name
1. Select Option 4, Prepare Plans, from the GIX main menu.

The Prepare Plans panel is displayed, shown in Figure 50.

<table>
<thead>
<tr>
<th>NDMPKGAE</th>
<th>PREPARE PLANS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection ====&gt; 7</td>
<td></td>
</tr>
<tr>
<td>Enter:</td>
<td></td>
</tr>
<tr>
<td>Plan name</td>
<td>TEST00</td>
</tr>
<tr>
<td>Plan model</td>
<td>PLMODEL</td>
</tr>
</tbody>
</table>

Select one of the following:

1. CREATE Make a new plan
2. CHANGE Change one of your plans
3. DELETE Delete one of your plans from the plan library
4. VALIDATE Check plan consistency, authorizations, and item availability
5. SUBMIT Send a plan to the TCF for transmission
6. RENAME Change the name of one of your plans
7. PRINT Obtain a printout of or browse a plan

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR

Figure 50. Example of a Prepare Plans panel

2. Select the plan you want to browse or print. There are two ways in which you can do this:

a. Type the name of the plan in the Plan name field. Select Option 7, Print/Browse and press Enter, or

b. Leave the Plan name field blank, and press Enter. The Full-list of Plans panel is displayed. Use the P (Print) line command next to the plan you want to select, and press Enter.

The Browse/Print of Plan panel is displayed, shown in Figure 51 on page 124.
Step 2: Browse or Print the Plan or Phase

```
NDMPKLAEBROWSE/PRINT OF PLAN: TEST00
Selection ===> 09:10
Enter:
Phase name _______ Accepted for options 1 and 3 only
(leave blank and select 1 or 3 for full-list)

Select one of the following:
1 BROWSE PHASE
2 BROWSE PLAN
3 PRINT PHASE
4 PRINT PLAN

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR
```

Figure 51. Example of a Browse/Print of Plan panel

You can use this panel in either of the following ways:

- To browse or print the whole plan
- To browse or print just one phase of the plan

To browse or print the whole plan, follow the instructions below:

1. Do not type anything in the Phase name field.
2. Select **Option 2, Browse Plan**, or **Option 4, Print Plan**, and press Enter.

   If you chose the Browse option, the Browse Plan Information panel is displayed. This panel is shown in Figure 52 on page 125. If you chose the Print option, you receive the message **PRINT REQUEST IS QUEUED**.

To browse or print just one phase of the plan, follow the instructions below:

1. Select the name of the phase. You can do this in one of two ways:
   - Type the name of the phase in the Phase name field. Select **Option 1, Browse Phase**, or **Option 3, Print Phase**, and press Enter.
   - Or, leave the Phase name field blank. Select **Option 1** or **Option 3**, and press Enter. The Full-list of Phases panel is displayed. Use the S (Select) line command to select the phase you want to browse or print and press Enter.

   If you chose the Browse option, the Browse Plan Information panel is displayed. This panel is shown in Figure 52 on page 125. If you chose the Print option, you receive the message **PRINT REQUEST IS QUEUED**.
### Browsing or Printing a Plan Using GIX (End User)

#### Figure 52. Example of the Browse Plan Information panel (top view)

If all of the information is not displayed on the screen, you can use the Scroll key to display the rest.

#### Figure 53. Example of the Browse Plan Information panel (bottom view)
Creating or Changing a Plan Using GIX

This section shows you how to create a transmission plan. You never have to create a plan from scratch. You always use an existing plan as a model. The model provides a sample framework. You create the new plan by editing the model.

Step 1: Define a Plan Name
1. Select Option 4, Prepare Plans, from the GIX main menu. The Prepare Plans panel is displayed.
2. Type the name that you want to give to your new plan in the Plan name field.
3. You can optionally use an existing plan, including the default plan model, as a model for the one you are creating. Type the name of the existing plan in the Plan model field.
4. Select Option 1, Create, and press Enter.
   The Create or Change a Plan panel is displayed.

The data you can enter on the Prepare Plans panel is described below.

Plan name: Specify a symbolic name to identify the plan. You can enter up to eight alphanumeric characters. The first character must be alphabetic.

Step 2: Define the Attributes of a New Plan
Using the S (Select) command on the Create or Change a Plan panel, you can:
- Define the attributes of the new plan, by changing the existing attributes.
- Define the attributes of phases in the plan.
- Define the attributes of functions in the plan (the attributes are specific to the resource type and function).
Figure 54. Example of a Create Or Change a Plan panel

1. Type the S (Select) command on the plan line of the plan whose attributes you want to change, and press Enter.

   The Specify Attributes of Plan panel is displayed.

Figure 55. Example of the Specify Attributes of Plan panel

2. Enter the plan attributes on this panel.

3. When you have finished, press Enter to return to the Create or Change a Plan panel.
Creating or Changing a Plan Using GIX (End User)

The plan attributes you can define are described below.

**Recursive:** Enter 1 in this field, if the plan is recursive. Enter 2, if the plan is not recursive.

**Start time delay field:** If you entered 1 in the Recursive field to indicate that the plan is recursive, you must also enter a value in this field. Enter an integer from 1 to 24 to indicate the number of hours that the plan can be delayed after the scheduled transmission time.

**Plan priority:** Enter an integer from 0 to 9 to indicate the priority with which the plan is executed. 0 represents the highest priority, 9 represents the lowest priority. The default value is 4. The phases of the plan are scheduled first by priority and then by scheduled date and time.

**Plan notes:** You can enter any notes about the plan in this field, up to a maximum of 132 characters.

**Step 3: Specify the Phase Attributes of the Plan**

1. On the Create or Change a Plan panel, type the S (Select) command on the phase line of the phase whose attributes you want to change, then press Enter.

   The Specify Phase Attributes of Plan panel is displayed.

   ![Figure 56. Example of a Specify Phase Attributes of Plan panel](image)

2. Enter the phase attributes on this panel. If you specify 1 (Yes) in the Windows field, to define a transmission window, you cannot enter a cutoff time or interval in the Cut off field.

   **Note:** You cannot define transmission windows for phases in recursive plans.

3. When you press Enter, the Phase Parameters panel is displayed. If you specified 2 (No) in the Windows field. If you specified 1 (Yes) in the Windows field, to define a transmission window, the Specify Windows Attribute of Phase
Creating or Changing a Plan Using GIX (End User)

panel is displayed. Go to “Step 4 (Optional): Define Transmission Windows” on page 131.

<table>
<thead>
<tr>
<th>NDMPK4DE</th>
<th>PHASE PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command =&gt;</td>
<td></td>
</tr>
<tr>
<td>Enter desired value or accept the one shown:</td>
<td></td>
</tr>
<tr>
<td>16 Track option 1 1 = Yes 2 = No</td>
<td></td>
</tr>
</tbody>
</table>

| Tracking information: | |
| Plan: MYPLAN | Phase: PH1 |

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 1/zerodot=PRINT 12=CURSOR

Figure 57. Example of a Phase Parameters panel

You can use the Phase Parameters panel to override the value defined at installation time for the tracking option. You can also override the parameter you define here at the function level (see Figure 60 on page 133).

4. Enter 1 in the **Track option** field if you want all of the resources in this phase to be tracked. Enter 2 if you do not want the resources to be tracked. The value you enter overrides the value defined at installation time in the NDMRES generation macro.

The phase attributes that you can define are described below.

**Phase name:** Specify the name of the phase. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. When you create a phase using GIX by copying or repeating an existing one, the new phase is given a temporary name that begins with the special character à. You can accept this name or change it. If you change it, the new name must follow the naming conventions just given. Otherwise, you will receive an error message when you try to change the phase.

**Node name or group of nodes:** Specify either the name of the node, or the name of the group of nodes, that the phase addresses. The nodes must be defined in the DRD.

**Scheduling:** Specify the date and time when you want the phase to be executed. If the plan is recursive, all the phases in the plan must have the same scheduled date, but they can have different scheduled times. If you do not enter any values, the default is the date and time when the plan is submitted. If the plan has a priority, the phases are submitted first by priority and then by scheduled date and time.

**Cut off:** Specify the cut-off time for the phase. This is the time when execution of the phase should stop. You can either an absolute time or an interval. The interval is added to the time you specify in the **Scheduling** field. When the cut-off time is
reached, the TCP deletes all of the phase by nodes for this phase, unless their status is COMPLETED.

**Hold option:** Enter 1 (Yes) in this field if you want the phase to be held until it is released by an IOF operator or an 8100/DPPX user command.

**Conditioning phase:** Specify the name of a phase whose completion conditions the start of the phase you are defining. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

**Conditioning criteria:** The conditioning criteria for a phase can either be:

- **GREATER THAN OR EQUAL TO:** The conditioned phase is executed if the conditioning phase was unsuccessful. In this case, the return code from the conditioning phase is greater than or equal to the termination value assigned to the conditioned phase.

- **LESS THAN OR EQUAL TO:** The conditioned phase is executed if the conditioning phase was successful. In this case, the return code from the conditioning phase is less than or equal to the termination value assigned to the conditioned phase.

**Conditioning value:** Specify a value greater than zero for the conditioning return code. If a function ends with a return code greater than or equal to this value (depending on what you define for the conditioning criteria), then the phase is terminated.

**Conditioning mode:** Use this field if you specified a node group. Specify the type of conditioning mechanism that you want to use. The default is :1 (All), which means that the phase conditioning is not performed by node. If you specify 2 (Single), the return code of each phase for node in the conditioning phase is used to condition the execution of the phase for node that addresses the same node in the conditioned phase as follows:

- The phases for node for which the conditioning criterion is met are released.
- The phases for node for which the conditioning criterion is not met are marked as deleted by conditioning (return code=12).

You can enter 2 (Single) only if the conditioning phase and the conditioned phases address the same group of nodes.

**Termination value:** Specify the value of a return code to be applied to all of the functions in the phase. If a return code for any function is equal to or greater than this value, any functions that follow it will not be started. Enter any integer from 1 to 99. The default is 0, which means there is no termination value.

**Batch job name:** Use this field to define a batch job to be run when the phase terminates. You can enter from 1 to 8 alphanumeric characters. The first character must be alphabetic.

**Batch job submission conditioning value:** Specify a phase return code value that will condition the submission of the batch job you defined in the **Batch Job Name** field. The batch job will be submitted to the operating system at the end of the phase, if the phase return code is less than or equal to this value.
**Phase notes:** You can enter any type of information about this phase that you want to record for future reference, up to a maximum of 132 characters.

**Windows:** Specify whether or not you want to define transmission windows for this phase. If you enter 1 (Yes), a further panel is displayed once you press Enter (see “Step 4 (Optional): Define Transmission Windows”). The default value is 2 (No).

**Track option:** Specify whether or not you want the resources in this phase to be tracked. The value you enter here at the phase level overrides the value defined at installation time in the NDMRES installation macro.

**Logon/password:** Specify the logon ID and password that authorizes the host to establish operations with the node. These values must match the ones defined at the node. Enter an asterisk (*) to confirm the values specified when the node was defined (see “Step 3: Specify Node Attributes” on page 37).

**Step 4 (Optional): Define Transmission Windows**
If you entered 1 (Yes) in the Windows field, to define a transmission window for the phase, the following panel is displayed.

![Figure 58. Example of a Specify Windows Attributes of Phase panel](image)

You use this panel to specify transmission windows for this phase. A transmission window is a time window within which you want transmission activities for this phase to take place.

You can specify one window per day in each phase. The initial time is the time when you want the phase to start. The initial time applies to the day specified in the window.
The end time is the time when you want transmission activities to stop. The end time could apply to the day after the one specified in the window, because the maximum interval between times is 23 hours and 59 minutes. This means that you can define up to seven windows for a phase, one for each day of the week.

Windows for a phase must not overlap; for example, if you define a window for Monday between 06:00 and 01:00 on Tuesday, you cannot define a window that begins on Tuesday between 00:00 and 01:00.

Here is an example to define three transmission windows for the phase called PH1, using the Specify Windows Attributes of Phase panel:

1. Specify a start and end time in the **Sunday Window** field to define a window starting on Sunday. For example, 13:00 to 08:00 (ending on Monday).
2. Specify a start and end time in the **Monday Window** field to define a window starting on Monday. For example, 17:00 to 20:00.
3. Specify a start and end time in the **Friday Window** field to define a window starting on Friday. For example, 17:00 to 16:59 (ending on Saturday).
4. Specify a cutoff date in the **Cut-off Window Date** field. This is the date when execution of the phase must stop.
5. Specify a cutoff time in the **Cut-off Window Time** field. This is the time when execution of the phase must stop.

If you submit the transmission plan containing this phase inside a window, the TCP puts the phase in READY status. If you submit the plan outside of a window, the TCP holds the phase and puts it in the status HWIN until the next window starts.

In this example, the phase will start at the beginning of a window, such as 13:00 on Sunday. If the phase is not completed by 08:00 on Monday, the status becomes HWIN until the start of the next window.

**Step 5: Define the Attributes of Functions in a Plan**

The attributes that you can change are specific to:

- The type of node that the phase addresses
- The type of function whose attributes you are defining
- The type of resource that the function addresses

**Note:** A Retrieve function in a transmission plan between nodes with CMFP (or CMFP and NDMT combined) functional capabilities always causes a replacement of the resource.

Here is an example to insert an Initiate transmission function:

1. On the Create or Change a Plan panel, type the I (Insert) command on a function line to insert a new function beneath it, and press Enter.

   The Function Resource Selection panel is then displayed.
2. Select **Option 4, Initiate**, in the **Selected function** field.

3. Select **Option 3, Procedure**, in the **Selected resource** field.

4. Press Enter. The Procedure Qualifiers panel is displayed.

5. Enter the resource name parameters on this panel. The actual panel varies according to the function and the resource type you are working with. For example, if you were retrieving software, the panel would look like Figure 61 on page 134.
Creating or Changing a Plan Using GIX (End User)

NDMP7JAE SOFTWARE QUALIFIERS (PLAN: PLAN/zerodot1)
Command ==> 
Enter desired values or accept the ones shown:
1 Resource name
  ==> AAA.BBB.*.DDD.*
2 Group name . . . . . . . . . . Required if field 1 not specified
3 Matching indicators ==> 
L = Low  
H = High  
I = Ignore
4 Send option . . . . . . . . 2  
  1 = Yes  
  2 = No
5 Track option . . . . . . . . 1 
  1 = Yes  
  2 = No
6 Netid. . . . . . . . . . . Any integer from 2 to 999999
7 Netlu. . . . . . . . . . . Any integer from 2 to 10
8 Bydestid . . . . . . . . . 2
  1 = Yes  
  2 = No
9 User def. Processing Size : Any integer from 1 to 999999
10 Force option . . . . . . . 1 
  1 = Yes  
  2 = No  
  3 = Desired
11 Extend File System option. 
2  
  1 = Yes  
  2 = No  
  3 = Desired
Tracking information:
Activity: CREATE
Phase . : PHCMEP
Node: CMEPNODE
PF 1=HELP 2=SPLIT 3=END 4=RETURN
9=SWAP 10=PRINT 12=CURSOR

Figure 61. Example of a Software Qualifiers panel

6. Once you have finished, press Enter. The Initiate Parameters panel is displayed.

NDMP7SAE INITIATE PARAMETERS (PLAN: PLAN01)
Command ==> 
Enter desired values or accept the ones shown:
1 Execution date . . . . . 09/15/90 (MM/DD/YY)
2 Execution time . . . . 10:30 (HH:MM)
3 Timeind. . . . . . . 1 
  1 = Local  
  2 = GMT
4 Procedure Data
==> 

PF 1=HELP 2=SPLIT 3=END 4=RETURN
9=SWAP 10=PRINT 12=CURSOR

Figure 62. Example of an Initiate Parameters panel

7. Enter the function parameters for the Initiate Procedure function on this panel.
   The function parameters vary for each transmission function. The Procedure data field, when displayed, lets you enter up to 60 characters of information to specify user parameters for the procedure.

8. When you have finished, press Enter and you are returned to the Create or Change a Plan panel.

9. On this panel, type SAVE on the command line to save the modified plan.
The plan containing the Initiate Procedure function that you have just defined looks like this:

```
Function . . . . . . . . . . . : INITIATE PROCEDURE AAA.BBB.*.CCC.*
Function termination rc . . . . .:
Track option . . . . . . . . . : YES
Netid . . . . . . . . . . . . . : 3
Netlu . . . . . . . . . . . . . : 5
Send . . . . . . . . . . . . . : NO
Date / Time . . . . . . . . . .: 09/15/90 10:30
Timeind . . . . . . . . . . . : LOCAL
```

Figure 63. Example of an initiate procedure function in a transmission plan

Step 6: Inserting New Phases and Functions into a Plan

Use the following commands to change the existing structure of the plan you are using as a model:

- C  Copy
- M  Move
- R  Repeat
- I  Insert
- B  Before
- A  After
- D  Delete
- G  Get plan
- BS  Begin sequence, VSE nodes only
- ES  End sequence, VSE nodes only
- FT  Change function termination values
- FI  Fast insertion from DRD
- FR  Fast insertion from repository

**Copying an Existing Phase:** One way of inserting a new phase into a transmission plan is to use the C or CC (Copy Block) line commands to specify the section you want to copy, followed by either B or A to specify where you want to place the new section. Figure 64 on page 136 shows you how to create a new phase.
Creating or Changing a Plan Using GIX (End User)

Figure 64. Example of copying an existing phase

1. Type the following, in any order:
   - **CC** on the phase line where phase PH12 starts
   - **CC** where phase PH12 ends
   - **B** on the phase line where phase PH11 starts

2. Press Enter.
   
   Phase PH12 is copied and inserted just before phase PH11.

3. You can accept the automatically generated name or change it. If you change it, follow the naming convention described in “Phase name” on page 129.
   
   Type **S** (Select) on the new phase line. Changing phase attributes is described in “Step 3: Specify the Phase Attributes of the Plan” on page 128.

Creating a New Phase or Function: Instead of copying an existing phase, you can also create a new phase from scratch and then insert functions into it.

1. On the Create or Change a Plan panel, use the **R** or **C** line commands to duplicate just the phase line.

2. Use the **I** (Insert) command to insert and create functions in the phase.

Using Fast Insertion of Functions from the DRD: Once you have created new phases, the quickest way to add new functions is to use the **FI** (Fast Insertion from DRD) command, or the **FR** (Fast Insertion from Repository) command.

The **FI** (Fast Insertion from DRD) command adds a set of functions for a set of resources, or resource groups, defined in the DRD.

1. Type **FI** on any line except the plan line, at the position where you want to add functions, then press Enter.

   The Fast Insertion from DRD panel is displayed.
Creating or Changing a Plan Using GIX (End User)

Figure 65. Example of a Fast Insertion From DRD panel. This example is to display all of
the Microcode resources stored in the DRD.

2. Select the resources (or groups of resources) by entering an asterisk (*), or a
partial name followed by *, in one of these fields:

   Resource name
   Resources group name
   Resources belonging to the group

3. Select the resource type of the resources by typing the number of one of the
allowed resource types shown in the Resource type field.

A Function Resource Selection panel is displayed. This panel displays the
transmission functions that you can select for this node type, and the resources
stored in the DRD for the node type of the phase.

Figure 66. Example of a Function Resource Selection panel
4. Enter line commands corresponding to transmission functions next to the resources in the list.

5. Type the primary command **SAVE** on the command line, then press Enter.

The transmission functions are generated for the selected resources and inserted in your plan. You are returned to the Create or Change a Plan panel.

**Using Fast Insertion of Functions from the Resource Repository:** The FR (Fast Insertion from Repository) command is used to select specific resources from the resource repository to include in a transmission plan.

1. Type **FR** on any line except the plan line, at the position where you want to add functions, and press Enter.

   The Fast Insertion from Repository panel is displayed.

   ![FAST INSERTION FROM REPOSITORY (NODE: AR01, PLAN: FASTIN) ROW 1 OF 9](image)

   **Figure 67. Example of the Fast Insertion From Repository panel**

   2. Select the resources (or groups of resources) by entering a blank, or a partial name followed by an asterisk (*), in the **Resource name** field.

   3. Select the resource type of the resources by typing the number of one of the allowed resource types shown in the **Resource type** field.

   A Function Resource Selection panel is displayed. This panel displays the transmission functions that you can select for this node type, and the resources stored in the resource repository for the node type of the phase.
4. Enter line commands corresponding to transmission functions next to the resources in the list.

5. Type the primary command **SAVE** on the command line, and press Enter. The transmission functions are generated for the selected resources and inserted in your plan. You are returned to the Create or Change a Plan panel.

**Step 7: Moving Elements in a Plan**

To move existing phases and functions around in a plan:

1. Type **M** next to a line that you want to move, or **MM** on the first and last lines of a section that you want to move.

2. Type **B** or **A** on a line to specify where you want to move the line or section to.
   For example, typing **MM** instead of **CC** in Figure 64 on page 136 would move phase PH12 to before phase PH11.

**Step 8: Deleting Elements from a Plan**

To delete phases and functions from a plan, type **D** next to all the lines that you want to delete, and press Enter. Or you can type **DD** on the first and last lines of a section that you want to delete, and press Enter. For example, typing **DD** instead of **CC** in Figure 64 on page 136 would delete the phase PH12 and all the functions contained within it.

**Step 9: Defining Termination Values**

Termination values are described in “Conditioning within a Phase” on page 121. You can define termination values at four levels:

- At the **phase** level. You do this when you define phase attributes, described in “Step 3: Specify the Phase Attributes of the Plan” on page 128.

- At the **sequence** level.

- At the **function** level.

- For **batch jobs**. You do this when you define phase attributes, described in “Step 3: Specify the Phase Attributes of the Plan” on page 128.
Changing a Plan Using GIX (End User)

**At the Sequence or Function Levels:** Follow the instructions below to set termination values at either the sequence or the function levels:

1. On the Create or Change a Plan panel, type the line command **FT** (Function Termination) on the line **BEGIN SEQUENCE**.
2. Press Enter. The Change Termination Values panel is displayed.
   a. To set the sequence termination value, type a number in the command line, and press Enter. The number that you type will appear in the field called **Previous sequence termination value entered**.
   b. To set a function termination value, type a number on the same line as the function whose termination value you want to set.
3. Press PF3 to return to the Create or Change a Plan panel.

**Step 10: Using the Get Plan Command**

You can use the **G** (Get) command to insert phases and functions obtained from the TCF plan that have a termination return code greater than or equal to a specified termination return code. You can also insert related conditioned phases and functions. You can also specify a partial name or a recursive plan name as the TCF plan name.

In the first case, a full-list of the TCF plans matching the specified partial name is displayed. In the second case, a full-list of the specified plan recursions is displayed. You can then select the TCF plan name directly from the list.

**Step 11: Saving Your Plan**

When you have finished making changes, you must save the plan definition by typing the primary command **SAVE** on the command line.

**Changing a Plan Using GIX**

You can change your own transmission plans defined in the plan library. If you have the Manage Other Users’ Plans authorization, you can also change plans created by other users. However, the authority level of the plan owner is used to determine what can be in a plan. This means that you cannot add functions, nodes, or resources to which the actual plan owner is not authorized.

You use the same set of panels as you did when creating the plan for the first time. Follow the instructions below to display the Create or Change a Plan panel.

1. From the Prepare Plans panel, shown in Figure 50 on page 123, specify the name of the existing plan. There are two ways in which you can do this:
   a. If you know the name of the plan, type the name in the **Plan name** field. Select **Option 2, Change**, and press Enter.
   b. If you do not know the name of the plan, you can select the plan from a full-list panel. Leave the selection field blank, and press Enter. The **Full-list of Plans** panel is displayed. Use the **S** (Select) line command to select the plan that you want to change.

   The Create or Change a Plan panel is displayed, shown in Figure 54 on page 127.

2. Follow the instructions described in “Creating or Changing a Plan Using GIX” on page 126, to change an existing plan rather than define a new one.
Deleting a Plan Using GIX (End User)

Querying Space in the Plan Library

You can query the amount of space left in the plan library at any time from any of the panels displayed from the Prepare Plans main menu option. Follow the instructions given below.

1. Type QSPACE in the command field of any of the GIX panels that you access after selecting Option 4, Prepare Plans, from the GIX main menu.

The Browse Plan Library Space panel is displayed.

<table>
<thead>
<tr>
<th>NDMPFQPE</th>
<th>BROWSE PLAN LIBRARY SPACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command =&gt;&gt;</td>
<td>Scroll =&gt;&gt;PAGE</td>
</tr>
<tr>
<td>09:02</td>
<td></td>
</tr>
</tbody>
</table>

Plan Library Space Status

Plan Library

K-Bytes total . . . : 1191
K-Bytes free. . . . : 1163

Figure 69. Example of a Browse Plan Library Space panel

2. If you see that you are running out of space, you can either delete some resources, or refer to “Maintaining the Plan Library” on page 444 for further information about how to maintain the plan library.

Deleting a Plan Using GIX

You can delete your own transmission plans from the plan library. If you are the system administrator or you have the Manage Other Users' Plans authorization, you can also delete plans created by other users.

To delete a transmission plan from the library, follow the instructions below from the Prepare Plans panel.

1. Specify the plan that you want to delete. There are two ways in which you can do this:

   a. If you know the plan name, type in the name of the plan in the Plan name field. Select Option 3, Delete, then press Enter.

   The Confirm the Deletion of Plan panel is displayed.

   b. If you do not know the name, leave the Plan name field blank and press Enter. Select the plan from the Full-list of Plans panel using the D (Delete) line command.

   When you press Enter, the Confirm the Deletion of Plan panel is displayed.

2. Type D (Delete) on the command line and press Enter to delete the plan. To quit without deleting the plan, press PF3 to return to the Prepare Plans panel.
Renaming a Plan Using GIX

You can rename your own transmission plans defined in the plan library. If you have the Manage Other Users' Plans authorization, you can also rename plans created by other users.

To rename a transmission plan in the plan library, follow the instructions below from the Prepare Plans panel.

1. Specify the plan that you want to rename, either by typing the name of the plan in the Plan name field, or by selecting the plan from the Full-list of Plans panel.
2. Select Option 6, Rename, then press Enter.
   The Rename Plan panel is displayed.
3. Type the new name of the plan on this panel, then press Enter. The plan is renamed. To quit without renaming, press PF3 to return to the Prepare Plans panel.

Validating and Submitting Plans

Once you have created a plan in the plan library, the plan does not come into effect until you validate it and submit it to the TCF. The plan is automatically validated when you submit it. You require a special user authorization (called Manage Submitted Plans, described in Chapter 2, “Defining User Profiles” on page 19) to validate and submit plans. This authorization also allows you to manage plans in the TCF after you have submitted them.

Using GIX, you can also validate a plan without submitting it. This is useful if you do not want to submit a plan immediately but want to check it, or if you do not have the Manage Submitted Plans authorization.

The validation process checks for errors in your plan. The errors that the validation process looks for are divided into two categories:

Severe If a plan has any severe errors, you cannot submit it to the TCF for execution.

Warning Warning errors do not prohibit the submission of a plan. However, they warn you that NetView DM for MVS will not be able to execute transmission functions successfully in the way that you have defined them.

Note: When a phase is conditioned by another phase, the validation process does not check that the resources mentioned in the conditioned phase are already contained in the resource repository. This is because the resources may be received into the repository from another node by the conditioning phase.
Validating a Plan Using GIX

Follow the instructions below to validate a plan without submitting it to the TCF.

1. Select Option 4, Prepare Plans, from the GIX main menu.
   The Prepare Plans panel is displayed.

2. Specify the name of the plan to be validated. There are two ways in which you can do this:
   a. If you know the name of the plan, type the name of the plan in the Plan name field. Select Option 4, Validate, and press Enter.
   b. If you do not know the name of the plan, you can select the plan from a full-list panel. Leave the selection field empty. Leave the Plan name field empty, or type in a partial name followed by an asterisk (*), and press Enter. The Full-list of Plans panel is displayed. Use the V (Validate) line command to select the plan you want to validate, and press Enter.

   If the plan has not been validated before, it is validated immediately, and the Validation Errors of Plan panel is displayed, shown in Figure 70 on page 144.

   If you have validated the plan before, another panel is displayed, which asks you whether you want to:
   • Validate the plan again
   • Browse the previous validation errors
   • Print the previous validation errors

The NetView DM for MVS validation process checks to see that:

1. You have defined sequences in the plan correctly.
2. The administrator has defined the nodes and resources in the DRD.
   • See Chapter 3, “Defining Nodes” on page 33
   • See Chapter 4, “Defining Resources” on page 61
3. The resources have been loaded into the resource repository.
   • See Chapter 6, “Preparing Resources for Transmission” on page 97
4. User authorizations to nodes and to resources have been respected.
   • See Chapter 2, “Defining User Profiles” on page 19
5. Groups of resources and nodes have been defined correctly in the DRD.
   • See “Organizing Nodes into Groups” on page 54
   • See “Organizing Resources into Groups” on page 79
6. Conditioning among phases is logically correct.
   • See “Conditioning within a Phase” on page 315
7. The functions and resources in each phase are valid for the types of node.
   • See Table 34 on page 327
   • See Table 37 on page 343

The Messages and Codes book lists all the error messages that can be generated during validation, and describes how to fix them.
3. Select an option. The Validation Errors of Plan panel is either displayed or printed.

```
NDMPKDAE   VALIDATION ERRORS OF PLAN: MYPLAN
Command ==> Scroll ==> PAGE
Line commands: H(Help)

Planner: TMSLEW  Validation date and time: 9/12/17 11:33

PH2 DESTINATION ROMA1SSP
NDMB0981 SEVERE ERROR - PHASE IS EMPTY.
PH6 INITIATE FUNCTION CATLIST
NDMB1071 SEVERE ERROR - RESOURCE RESTRICTED BUT NOT ASSIGNED TO A NODE.
PH6 SEND DATASET DSXCAT.RUNSAMP
NDMB103I WARNING - RESOURCE NOT FOUND IN RESOURCE REPOSITORY.
PH11 SEND MICROCODE MCODE.3174.NA.FUNCTEC.A0500.CONTROL
NDMB103I WARNING - RESOURCE NOT FOUND IN RESOURCE REPOSITORY.
```

Figure 70. Example of a Validation Errors of Plan panel

All the errors found in the plan are presented phase-by-phase on this panel. If there are a lot of errors, you can press PF3 to return to the previous panel and print the errors.

4. Correct all of the errors listed on the Validation Errors of Plan panel. You can use the Messages and Codes book to explain the messages, or you can request online help. Type H (Help) next to an error message and press Enter to display the help text for that message.

5. Once you have changed the plan to correct the errors, you can validate the plan again.

**Submitting a Plan Using GIX**

Follow the instructions below to validate and submit a plan to the TCF. Log on to the GIX main menu and move through a sequence of panels.

1. Select **Option 4, Prepare Plan**, from the GIX main menu.

   The Prepare Plans panel is displayed.

2. Specify the name of the plan to be submitted. There are two ways in which you can do this:

   a. If you know the name of the plan, type the name of the plan in the **Plan name** field. Select **Option 5, Submit**, and press Enter.

   b. If you do not know the name of the plan, you can select the plan from a full-list panel. Leave the selection field empty. Leave the **Plan name** field empty, or type in a partial name followed by an asterisk (*) and press Enter. The Full-list of Plans panel is displayed. Use the **SUB** (Submit) line command to select the plan you want to submit, and press Enter.

   The Submit Plan panel is displayed.
3. Review the plan summary information that is shown on this panel. You can give the plan a different name to be stored in the TCF. Type the new name in the **New name of plan** field, or leave it blank to keep the same name.

4. Press Enter when you are ready to submit the plan. If you want to quit without submitting the plan, press PF3 to return to the previous panel.

5. The plan is validated. If it passes the validation process, it is submitted to the TCF. If errors are found during the validation process, processing is halted and the validation errors are displayed. Press PF3 to continue processing. When the plan has been submitted, you return to the Prepare Plans panel where you receive the message **PLAN SUBMITTED**.

The field you can enter on the Submit Plan panel is described below.

**New name of plan**: You can define a new name for the plan, different from the name in the plan library. The submitted plan is stored in the TCF under the new name. When the IOF operator tracks the progress of the submitted plan, this is the name that will be displayed. You will also see the new name when you use **Option 5, Manage Submitted Plans**, described in Chapter 8, "Managing Submitted Plans" on page 147.
Chapter 8. Managing Submitted Plans

The plans that you submit for execution are stored in the transmission control file (TCF) until they are ready to be processed by the transmission control program (TCP). After you have submitted a plan, you can look at the information in the TCF. Working on submitted plans is the third step in the following sequence of tasks:

**Step 1.** Load resources into the resource repository  
**Step 2.** Prepare and submit a transmission plan  
**Step 3.** Work on submitted plans  
**Step 4.** Unload resources from the resource repository

The TCF contains:
- All transmission plans before they are processed by the TCP.
- Status information about each transmission for plans after they have been processed by the TCP. This includes:
  - The contents and the schedule of the plan, defined when you created the plan  
  - The status of the plan, its phases and phase-by-node records, and functions  
  - The transmission function return codes (listed in Table 15 on page 180).

This chapter explains how to use GIX to look at the information in the TCF.

Looking at TCF Information Using GIX

Use the instructions on the following pages at your terminal. Log on to the GIX main menu, and then use **Option 5, Manage Submitted Plans**, from the GIX main menu to:

- Look at information for a specific plan, using either **Option 1, Browse Plan Status**, or **Option 2, Print Plan**
- Look at information for phases, using **Option 4, List Phases**
- Look at information for phases addressing a specific node, using **Option 5, List Phases With Nodes**
- Look at information for functions, according to specific criteria, using options 6 through 10
- Delete and erase a plan, using either **Option 3, Delete Plan**, or **Option 11, Erase Plan**
- Save a plan, using **Option 12, Save Plan(s)**
- Look at a summary of plan status information, using **Option 13, Plan Status Summary**
- Reset a plan, using **Option 14, Reset Plan**
- List and control phases for nodes, using **Option 15, List Selected Phases With Nodes**
There is a set of primary commands that you can use on each panel. Each panel displays a subset of these commands for you to work with. The commands are:

- **B** Browse. Use this command to browse an item.
- **D** Delete. Use this command to delete an item. This command does not physically remove the item from the TCF, but inhibits transmission.
- **E** Erase. Use this command to physically erase an item from the TCF.
- **LE** List functions by exception. Lists the functions in a plan that have not yet started, or functions that resulted in an error during transmission.
- **LF** List functions. Lists the functions of a plan.
- **LN** List of phases with nodes. Lists the phases-by-node of a specific plan.
- **LP** List Phases. Lists the phases associated with a specific plan.
- **P** Print. Use this command to print an item.
- **PE** Print functions by exception. Prints the functions in a plan that have not yet started, or functions that resulted in an error during transmission.
- **PS** Plan summary. Displays a summary of information about a plan.
- **R** Refresh Listed Items. Refreshes the information displayed about phases with nodes.
- **RP** Reset plan. Resets a plan, plan recursion, or specific phase or phase-by-node record stored in the TCF. You can also remove transmission windows defined for a phase by resetting the phase.
- **SA** Save plan. Saves a plan in the TCF, but does not erase the plan.
- **SE** Save and erase plan. Saves a plan in the TCF, and then erases the plan automatically from the TCF after the plan has been executed.

### Plan Information

Follow the instructions below at your terminal, to browse or print transmission information for a plan, or a particular plan recursion.

#### Step 1: Specify the Plan Name

1. Select **Option 5, Manage Submitted Plans**, from the GIX main menu.

   The Manage Submitted Plans panel is displayed.
Figure 72. Example of a Manage Submitted Plans panel

2. Select the plan that you want to browse or print. There are two ways in which you can do this:

- If you know the plan name, type the name of the plan you want to look at in the Plan name field. If the plan is recursive, type the date of the recursion you want to work with in the Recursion date field. Select Option 1, Browse Plan Status, or Option 2, Print Plan, and press Enter.

- If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the Plan name field blank, or type a partial name followed by an asterisk (*). Leave the Recursion date field blank, and press Enter. The List of Plans full-list panel is displayed.

Type B (Browse) or P (Print) next to the plan that you want to select, and press Enter.

If you chose the Browse option, the Plan Information panel is displayed.

Step 2: Review the Plan Information
1. Look at the information displayed on this panel for the plan you specified. You can also use any of the primary commands on this panel to perform another task, for example, \texttt{LP}, to list the phases in this plan.

2. When you have finished, press PF3 to return to the previous panel.

The information that may be displayed on the Plan Information panel is described below (in alphabetical order).

- **Deleted by:** If the plan has been deleted, the user ID of the person who deleted it.
- **Highest return code:** The highest return code of all phases contained in the plan.
- **Max delay:** If the plan is recursive, the start time delay specified when the plan was created.
- **Number of completed/deleted phases:** The number of phases already transmitted or deleted in the plan.
- **Planner ID:** The user ID of the person who created the plan.
- **Plan notes:** The notes that were entered when the plan was created.
- **Priority:** The priority that was specified when the plan was created.
- **Recursion schedule:** If the plan is recursive, the date of the recursion being displayed.
- **Recursive plan:** Whether the plan is recursive or nonrecursive.
- **Status:** The transmission status of the plan, which can be one of the following:
Phase Information (End User)

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>All the functions in the plan have been processed. Some of the functions may have been skipped, due to the conditioning criteria that were specified between the phases or functions in the plan.</td>
</tr>
<tr>
<td>Deleted</td>
<td>A user has deleted the plan, using either GIX or the TCFMAINT Batch Utility. The TCP also deletes a plan if it does not start within the time interval specified by the plan's start time delay. The plan remains in the TCF but is flagged as deleted.</td>
</tr>
<tr>
<td>Deleting</td>
<td>A user has entered a request to delete the plan. The TCP is going to delete the plan, but the plan is temporarily flagged as “deleting.”</td>
</tr>
<tr>
<td>Submitted</td>
<td>The plan has been submitted to the TCF.</td>
</tr>
</tbody>
</table>

Table 10. Plan status

Submission date and time: When the plan was submitted to the TCF.

Submitted by: The user ID of the person who submitted the plan.

Total number of phases: The number of phases scheduled in the plan.

Phase Information

Follow the instructions below to browse the status of each phase in a particular plan, including any conditioning between phases, and the scheduled date and time of each phase.

Step 1: Select the Plan

1. Select Option 5, Manage Submitted Plans, from the GIX main menu.

   The Manage Submitted Plans panel is displayed. This panel is shown in Figure 72 on page 149.

2. Select the plan that the phase belongs to. There are two ways in which you can do this:

   - If you know the plan name, type the name of the plan you want to look at in the Plan name field. If the plan is recursive, type the date of the recursion you want to work with in the Recursion date field. Select Option 4, List Phases, and press Enter.

   - If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the Plan name field blank, or type a partial name followed by an asterisk (*). Leave the Recursion date field blank, and press Enter. The List of Plans full-list panel is displayed.

     Type LP (List phases) next to the plan that you want to select, and press Enter.

     The List of Phases panel is displayed. This panel shows all the phases in the plan you specified.

3. Type B (Browse) in the CMD column next to the phase that you want to browse, and press Enter.

   The Phase Information panel is displayed. This panel is shown in Figure 74 on page 152.
Step 2: Browse Phase Information

Figure 74. Example of a Phase Information panel

This panel lists all the phases in the plan. You can use any of the primary commands on this panel to perform another task, for example, **LN** to list the functions in each phase. The information on this panel is described below, in alphabetical order. When you have finished, press PF3 to return to the previous panel.

**Completed/deleted phases by node**: The number of completed and deleted phase-by-node records. If the phase addresses a single node, this value can be either 0 or 1.

**Conditioning phase name**: The name of the phase whose execution conditions the execution of this one.

**Conditioning return code**: The return code that the phase is conditional upon. See “Conditioning within a Phase” on page 315.

**Cut-off time interval**: The cut-off time interval is a time interval after the scheduled start date and time. If the phase is not completed within this time, the phase is terminated with return code 24 (see Table 15 on page 180), and NetView DM for MVS issues a message.

**Deleted by**: If the plan has been deleted from the TCF, the user ID of the person who deleted it.
**Phase Information (End User)**

*Highest return code:* If the functions in the phase have been executed, this field shows the highest of all function termination codes (see Table 15 on page 180).

*Highest return code to invoke procedure:* The threshold that determines whether the batch procedure should start.

*Max start date/time:* The latest date and time at which the phase can be started.

*Node name:* The name of the node, or group of nodes, that the phase addresses.

*Node or group of nodes:* Whether the phase addresses a single node or a group of nodes.

*Node type:* The type of node the phase addresses.

*Number of nodes:* If the phase addresses a group of nodes, the number of nodes in the group.

*Phase notes:* The notes entered when the phase was created.

*Phase status:* The status of the phase can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>There are no more functions in the phase to be executed. This can be because they have all been executed, or because the IOF operator has canceled the phase, or because of conditioning within a phase or conditioning between phases.</td>
</tr>
<tr>
<td>Deleted</td>
<td>A user has deleted the phase. Phases are also deleted if the cut-off time or interval was reached before the phase completed. Recursive plans are also deleted when the phase does not start within the time interval specified by the plan's start time delay. The phase remains in the TCF but is flagged as deleted until the plan to which it belongs has been erased.</td>
</tr>
<tr>
<td>Scheduled</td>
<td>The phase belongs to a plan which has been submitted, but not yet completed or deleted.</td>
</tr>
</tbody>
</table>

Table 11. Phase status

*Phase termination return code:* If a function terminates with this return code, the phase will terminate immediately.

*Procedure invoked at end of phase:* The batch procedure invoked at the end of the phase.

*Scheduled start date/time:* This is the date and time that the phase is scheduled to start. The TCP can begin to execute the phase at this time.

*TCP error message:* The number of the message issued by the TCP, if an error occurred during the submission of a job at the end of the phase. If the value is 0, there were no errors. Refer to the *Messages and Codes* book for information about error messages.
Phase by Node Information

You can look at information about phases in a plan which address a specific node, or group of nodes. A phase that addresses a specific node, or group of nodes, is referred to as a phase by node. Follow the instructions in “Phase Information” on page 151, but select Option 5, List Phases With Nodes, from the Manage Submitted Plans panel.

This displays the List of Phases With Nodes panel. You can select the phase by node that you want to look at using the B (Browse) command. This displays the Phase by Node Information panel for the phase by node that you specified.

![Phase by Node Information panel](image)

**Figure 75. Example of a Phase By Node Information panel**

The information displayed on this panel is described below, in alphabetical order.

**Cut-off time interval:** The cut-off time interval is an interval after the scheduled start date and time. If the phase is not completed within this time, the phase is terminated with return code 24 (see Table 15 on page 180), and NetView DM for MVS issues a message.

**DRD updated:** This field shows whether the resource history records in the DRD have been updated. These records are updated when the transmission of the phase produces results that change the status of resources at the nodes.

**End date/time:** The time that the last attempt to transmit the phase ended.
**First start date/time:** The first time the transmission of the phase was attempted.

**Highest phase return code:** The highest function return code of all the functions that were actually executed.

**Last start date/time:** If the phase is now active, this is the time that it started.

**Max start date/time:** The latest date and time the phase can be started.

**Network change effect:** Shows whether the phase by node was deleted by the TCP because of a network change, for example, the definition of the associated node was deleted or the associated logical unit was changed.

**Node type:** The type of node the phase addresses.

**No. of completed/deleted functions:** The number of functions in the phase that have been either completed or deleted, for all nodes.

**Scheduled start date/time:** This is the date and time when the phase is scheduled to start.

**Sequence termination return code:** The return code resulting from the transmission of a function that inhibits the transmission of other functions in the sequence (see Table 15 on page 180).

**Status:** The status of the phase by node record can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>A phase by node is flagged as completed when all of the functions in the phase have been completed, either successfully or unsuccessfully, or when the phase is terminated due to conditioning within a phase.</td>
</tr>
<tr>
<td>Deleted</td>
<td>A phase by node is flagged as deleted because:</td>
</tr>
<tr>
<td></td>
<td>• An IOF operator entered a Delete command.</td>
</tr>
<tr>
<td></td>
<td>• The cut-off time or interval was reached before the phase completed.</td>
</tr>
<tr>
<td></td>
<td>• Conditioning between phases is in effect.</td>
</tr>
<tr>
<td></td>
<td>• An unrecoverable error was encountered while executing the phase (VSE node types only).</td>
</tr>
<tr>
<td></td>
<td>• In the case of recursive plans, the phase did not start within the interval specified by the plan's start time delay.</td>
</tr>
<tr>
<td></td>
<td>• A network change request was issued, for example, a request to delete a node or change a logical unit name.</td>
</tr>
<tr>
<td>Executing</td>
<td>NetView DM for MVS and the remote node are currently exchanging information on the transmission line. If the node is connected indirectly to the host, the TCP communicates with the intermediate node.</td>
</tr>
<tr>
<td>HCOND</td>
<td>Held by Condition. This means that the phase was held because the phase which conditions it is not yet completed.</td>
</tr>
<tr>
<td>HDEST</td>
<td>Held by Destination. This means that the node was held by an IOF Hold or Immediate Hold command.</td>
</tr>
</tbody>
</table>
Table 12. Phase by node status

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSYST</td>
<td>Held by TCP. This means that the phase was held either because of a line problem, the node is down, or because of a logic problem, and the retry counter is exhausted.</td>
</tr>
<tr>
<td>HRETR</td>
<td>Held/retrying. This is either the same as for HSYST, or because an SNA Request Shutdown command was received from the node. The difference is that the retry counter is not yet exhausted.</td>
</tr>
<tr>
<td>HWIN</td>
<td>Held by TCP. This means that a phase with transmission windows defined has been held by the TCP because of one of the following:</td>
</tr>
<tr>
<td></td>
<td>• The plan containing the phase was submitted outside of a defined time window.</td>
</tr>
<tr>
<td></td>
<td>• The phase did not reach pending status by the end of the window.</td>
</tr>
<tr>
<td>Held</td>
<td>Held by the IOF operator, or belongs to a phase defined as held.</td>
</tr>
<tr>
<td>Pending</td>
<td>At least one function has been interrupted, and NetView DM for MVS is waiting for a delayed acknowledgement or data from the node. Execution continues when this has been received from the node.</td>
</tr>
<tr>
<td>Ready</td>
<td>The scheduled date and time have been reached, and no other condition inhibits the phase from starting (neither the phase nor the node has been held, and the line is active). The TCP has not yet started executing functions in the phase.</td>
</tr>
<tr>
<td>Restartable</td>
<td>The transmission of at least one function in the phase was interrupted due to a recoverable error. No other functions or resources are in pending status. An IOF Start command, or a TCP start or restart, causes the execution to be resumed.</td>
</tr>
<tr>
<td>Waiting</td>
<td>Waiting for the scheduled date and time.</td>
</tr>
</tbody>
</table>

Total elapsed time: The time taken to transmit the phase.

Total number of functions: The number of transmission functions defined in the phase.

Function Information for LU0 Resources

Follow the instructions below to browse or print information about the transmission status of functions.

You can select from options 6 through 10 on the Manage Submitted Plans panel:

- Option 6 lists all the functions in a specific plan.
- Option 7 lists all exceptional functions in a specific plan. These are functions that have not yet started, or functions that resulted in an error during transmission.
Function Information for LU0 Resources (End User)

- **Option 8** lists all exceptional functions in the TCF, or in a specific plan. These are functions that have resulted in an error during transmission.

- **Option 9** prints all exception functions in a specific plan. These are functions that have resulted in an error during transmission.

- **Option 10** prints all exception functions in the TCF, or in a specific plan.

1. Select the plan that the function belongs to. There are two ways in which you can do this:
   - If you know the plan name, type the name of the plan you want to look at in the **Plan name** field. If the plan is recursive, type the date of the recursion you want to work with in the **Recursion date** field. Select from options 6 through 10 to list functions, and press Enter.
   - If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the **Plan name** field blank, or type a partial name followed by an asterisk (*). Leave the **Recursion date** field blank, and press Enter. The List of Plans full-list panel is displayed.

   Type **LF** (List functions), **LE** (List functions by exception), or **PE** (Print functions by exception) next to the plan that you want to select, and press Enter.

   The List of Functions panel is displayed.

2. Use the line commands, **B** (Browse) or **P** (Print), against the function that you want to work with.

   The Function Information panel is displayed.
Figure 76. Example of a Function Information panel. This example is for a Dataset resource type at a VSE node.

This panel contains three types of information:

- Information that specifies which function record is being displayed. This includes the plan name, phase name, node name, the transmission function verb and modifier (for example, SEND DATASET), and the resource name (for example, DATA1.MINE).

- Information that was defined when the you defined the function in the plan library, before submitting the plan to the TCF. If the function addresses a group of resources, this information is repeated for each resource in the group. The information varies according to the type of node, of transmission function and of resource, but includes attributes such as share, protection, authorization, compressed, decompressed, disposition, and resynchronization.

- Information that describes what has happened to the function record since the plan was submitted to the TCF.

The Function Information panel may display any of the fields described below (in alphabetical order).

**Bytes transmitted:** The number of bytes transmitted.
**DRD updated:** This field shows whether the resource history records in the DRD have been updated. These records are updated when the transmission of the phase produces results that change the status of resources at the nodes.

**Highest return code:** The highest return code (see Table 15 on page 180).

**Status:** The status of each transmission function record can be:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Completed, either successfully (return code=0), or unsuccessfully.</td>
</tr>
<tr>
<td>Not started</td>
<td>The TCP has not yet started executing the functions, or the function belongs to a phase by node that was deleted, or the function or resource was not executed at the node (VSE nodes only).</td>
</tr>
</tbody>
</table>
| Pending     | This only applies to VSE, DPPX, SSP nodes that deal with compressed resources:  
               - Either the function or resource has been delivered to the node. There is no transmission activity while the TCP is waiting for an acknowledgement from the node. After the TCP receives an acknowledgement, the function status becomes completed (successfully or not).  
               - Or, the TCP may be receiving data from the node. After the TCP receives the data, the function status becomes completed (successfully or not), or the function may be resumed until transmission activity is completed. |
| Restartable | This does not apply to VSE nodes. The function was interrupted by a restartable error. |
| Running     | The host and the remote node are currently exchanging information on the transmission line. |

*Table 13. Function status*

**TCP error message:** The number of the message issued by the TCP, if an error occurred during the submission of a job at the end of the phase. If the value is 0, there were no errors. Refer to the *Messages and Codes* book for information about error messages.

**Primary and secondary return codes:** These are described in “Recovering from Transmission Failures” on page 180.

**Node error message:** These are described in “Recovering from Transmission Failures” on page 180.

**Transmission error identification:** This field identifies transmission errors that have occurred. The information can be:

- The ID of the last error message issued by the TCP.
- The ID of the error messages issued at the node, if applicable. These are either generated at the end node, or at one of the intermediate nodes.
- The primary and secondary return code from the node, if applicable. These are either generated at the end node, or at one of the intermediate nodes.
Function Information for LU 6.2 Resources (End User)

- Return and feedback codes from TCAM.
- TCAM system and user senses, if applicable.

Function Information for LU 6.2 Resources

The Function Information panel for microcode distribution for LU 6.2 nodes displays a different set of fields to the panel described above for LU0 nodes.

Figure 77 shows a sample panel for the ACCEPT MICROCODE function to distribute microcode.

![Sample Function Information panel for Microcode distribution](image)
There may be some extra fields of information for those resource types that you use with LU 6.2 node types. These are described below, in the order in which they appear on the panel.

**Function**: The function name.

**Function termination rc**: See Table 15 on page 180.

**Function rc**: The function return code.

**Function status**: See Table 13 on page 159.

**Resource name**: The name of the resource.

**Matching indicators**: These specify the list of matching indicators used at the host to identify the resource. These can be a hyphen (-), which means *must match*, HIGH, IGNORE, or LOW. These apply only to the Install, Send, Retrieve, Delete, and Initiate functions.

**Transmission return code**: The return code resulting from the transmission.

**Status**: This shows the current condition of a function. It can be one of the following:

- COMPLETED
- STARTED
- RUNNING
- PENDING
- RESTARTABLE
- NOT RUNNING

**Bytes transmitted**: The number of bytes transmitted.

**Node reports**: These are a set of responses from the node, which include the following information:

**Process at the node**: The result of the function, which can be:

- SUCCESSFUL
- NOT SUCCESSFUL
- NOT ATTEMPTED
- WILL NOT BE ATTEMPTED
- For the Activate function, SUCCESSFUL actually means *will be attempted*

**When effective**: The effectiveness of the function, which can be:

- IN USE
- ACTIVATION REQUIRED
- NOT APPLICABLE

**Activation use**: Shows which type of activation causes components to be canceled automatically if the installation or a test are successful, in order to release resources required to maintain removability. These can be:

- TRIAL. The altered components are used only for an activation specifying trial use superseded by production components.
Function Information for LU 6.2 Resources (End User)

- **PRODUCTION.** The altered components are used for any activation, unless superseded by trial components.
- **NOT INSTALLED.**

**Removability:** Shows whether the microcode will be installed in a removable state. This can be YES, NO, or DESIRED.

**Side effects:** This is the effect that this transmission function may have on other resources. This applies to the Accept, Remove, and Install transmission functions only. The effect can be DELETED, REINSTALLED, or BACK-LEVEL.

**Side effect resource names:** The names of the resources effected as a result of the transmission function, in addition to the main resource. See “Side effects.”

**Effect:** The effect of the function, for example, DELETED.

**Snads report code:** Specifies an SNA/Distribution Services (SNA/DS) report code. These codes are listed in *SNA Formats*, GA27-3136.

**Agent report code:** Specifies an SNA report code. These codes are listed in *SNA Formats*, GA27-3136.

**Server report code:** Specifies an SNA report code. These codes are listed in *SNA Formats*, GA27-3136.

**TCP error message number:** If there is an error, this is the ID of the message issued by the TCP. See “Recovering from Transmission Failures” on page 180.

**Details on process performed:** This can include the following items:

- **Pretest:** Shows whether changes are to be tested prior to installation. This field can be YES, NO, or DESIRED.

- **Posttest:** Shows whether changes are to be tested after installation. This field can be YES, NO, or DESIRED.

- **Automatic removal:** Shows whether changes will be removed automatically if the installation or test fails. This field can either be YES, NO, or DESIRED.

- **Automatic acceptance:** Shows whether the removability of changes will be canceled automatically if the installation or test is successful. This releases the resources required to maintain removability. This field can either be YES, NO, or DESIRED.

- **Destruction:** Shows whether existing changes will be explicitly overwritten at the node if storage constraints occur. This field can be ALLOWED or NOT ALLOWED. It applies only to the Install and Send transmission functions.

- **Corequisites:** This applies only to the Install transmission function. It specifies a list of resources to be installed together with the main resource referred to.

- **To be deleted resource name:** The global name of the resource to be deleted at the node. This is shown only if DESTRUCTION=YES. If there is no name, the field displays NONE.
To be deleted matching indicator: Specifies the list of matching indicators to be used at the resource to identify the resource to be deleted at the node. Indicators can be a hyphen (-), which means *must match*, HIGH, IGNORE, or LOW. This is displayed only if the to be deleted resource name was partially specified.

Command execution: The result of the function. This field can be SUCCESSFUL, NOT SUCCESSFUL, NOT ATTEMPTED, or WILL NOT BE ATTEMPTED. For the Activate function, SUCCESSFUL actually means *will be attempted*.

Pre test: The result of the pretest. This field can be SUCCESSFUL, NOT SUCCESSFUL or NOT ATTEMPTED.

Post test: The result of the post-test. This field can be SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

Auto remove status: The result of automatic removal. This field can be SUCCESSFUL, NOT SUCCESSFUL or NOT ATTEMPTED.

Auto accept: The status of the automatic acceptance. This field can be SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

Removability (status): The status of removability. This field can be INSTALLED-REMOVABLE, INSTALLED-NOT REMOVABLE, or NOT INSTALLED.

Resources: The status of resources can be:

- DELETED
- BACK-LEVEL
- REINSTALLED

Activation: The effectiveness of the function. This field can be IN USE, ACTIVATION REQUIRED, or NOT APPLICABLE.

Detailed data: Additional information from the node, describing conditions at the node. This can be up to four lines, each line containing 64 bytes.

Accept if no error: This field can either be YES or NO.

Remove if error: This field can either be YES or NO.

Send: Shows whether the resource has to be sent. This field can either be YES or NO. It applies only to the Install and Initiate functions.

Initiate completion code: This field shows the initiate return code reported by the node.

Deleting and Erasing Plans and Phases

You can delete or erase a plan or phase. *Deleting* an item does not erase it from the TCF. When you delete a plan or phase, its transmission is inhibited, or, if it is already being executed, transmission is interrupted. If the TCP is not actually running when you delete a plan, the plan status is marked as DELETING instead of DELETED.
If the plan contains one or more phases which have been sent to a remote server, but have not yet been reported as successfully completed, you can choose to send a Delete request to the remote server(s). NetView DM for MVS will then carry out the following:

- If the Delete at Server request includes any phases/destinations for which the original request has not yet been transmitted, the original request will be deleted at the host.
- If the Delete at Server request includes any phases/destinations for which the original request has already been successfully executed at the remote server, no action will be taken with respect to these destinations.
- For all other phases/destinations, a Delete at Server request will be transmitted to the server.

If the remote server is running TME 10 Software Distribution Version 3.1.5 or later, it will attempt to delete the original request, and will send a confirmation report to the host if it is successful. If the host receives confirmations from all remote destinations that the requests in PENDING status have been cancelled, the plan at the host will be cancelled automatically. In the event that completed statuses are not received for all of the nodes, you can optionally Force Delete the plan, thereby ignoring the presence of requests still in pending status (note that the Force Delete option is only available from GIX; it is not available from the Batch Utilities).

You can also issue a Delete Pending request for a specific phase. In the same way as for a Plan, NetView DM for MVS deletes transmission requests to individual nodes that have not been started, and issues a Delete Pending request for all those that have started but where completed statuses have not yet been received.

For Delete Pending requests for both plans and phases, if the TME 10 Software Distribution Version 3.1.5 server is processing a function when it receives a Delete Pending request for the same function it will complete the processing and return a completed status to the host for the original function, ignoring the Delete Pending request.

Erasing a plan physically removes the plan from the TCF, and makes space for something else. You can only erase plans that have a status of COMPLETED or DELETED.

**When You Delete a Nonrecursive Plan**

- Functions that have already been executed are not affected.
- Functions that have not yet started will not be started.
- Functions that are currently being executed are interrupted.
- The plan and all its phases (phases by node) are marked as deleted.

**When You Delete a Recursive Plan**

- All the recursions of the plan already completed are not affected.
- The current recursion is deleted.
- No more plan recursions are generated.

**When You Delete a Plan Recursion**

- The recursion with the date you specify is deleted.
- The recursion for the next day is generated as usual.
Using GIX to Delete and Erase Plans (End User)

When You Delete a Specific Phase

- Functions in the phase that have already been executed are not affected.
- Functions in the phases that have not yet started will not be started.
- Functions in the phase currently being executed are interrupted.
- The phase (and related phases by node) are marked as deleted.

When You Delete a Plan or Phase Using the Delete at Server Option

- In addition to the information shown for the various types of delete command above, NetView DM for MVS will send a delete request for any nodes that have functions that have been started, but for which a completed status has not been received.
- The plan or phase will only be marked for deletion at the host if confirmations are received that all pending functions for all nodes were successfully deleted.

Using GIX to Delete and Erase Plans

Use the instructions on the following pages at your terminal. From the Manage Submitted Plans panel, illustrated in Figure 72 on page 149, follow these instructions to delete or erase a plan.

To Delete a Nonrecursive Plan

Select the plan that you want to delete or erase. There are two ways in which you can do this:

- If you know the plan name, from the Manage Submitted Plans panel type the name of the plan in the Plan name field. Select Option 3, Delete Plan, or Option 11, Erase Plan, and press Enter.
- If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the Plan name field blank, or type a partial name followed by an asterisk (*). The List of Plans full-list panel is displayed. Type D (Delete) or E (Erase) next to the plan that you want to delete or erase.

Either the Confirm the Deletion of Plan or the Confirm the Erasure of Plan panel is displayed, depending on whether you entered the D (Delete) or E (Erase) command.

Confirm the Deletion of Plan: Type D (Delete), F (Force) or S (Delete at Server) in the command field and press Enter.

D Deletes all phases in the plan for all nodes for which transmission has not started

F Deletes all phases in the plan for all nodes even if there are functions that have started transmission and for which a completed status has not been received; the deletion of those functions at the nodes will not be requested

S Deletes all functions of all phases in the plan that have not been started, and sends a delete request to the servers controlling all nodes where the transmission has started but a completed status has not been received.

When NetView DM for MVS receives a confirmation from a TME 10 Software Distribution Version 3.1.5 server that a function has been successfully deleted, it will check its records for the plan to see if there are still outstanding functions not yet deleted; if there are none it will delete the plan.

To quit without deleting or forcing, press PF3.
Using GIX to Delete and Erase Plans (End User)

**Confirm the Erasure of Plan:** Type E (Erase) in the command field and press Enter. You can only erase plans that have a status of COMPLETED or DELETED. To quit without erasing, press PF3.

**To Delete a Recursive Plan**
Use this procedure to delete a recursive plan so that no further recursions of the plan occur.

From the Manage Submitted Plans panel, type the name of the recursive plan in the Plan name field. Do not enter a date in the Recursion date field. Select Option 3, Delete Plan, or Option 11, Erase Plan, and press Enter.

Either the Confirm the Deletion of Plan or the Confirm the Erasure of Plan panel is displayed, depending on whether you entered the D (Delete) or E (Erase) command.

**Confirm the Deletion of Plan:** Type D (Delete), F (Force) or S (Delete at Server) in the command field and press Enter.

- **D** Deletes all phases in the plan for all nodes for which transmission has not started
- **F** Deletes all phases in the plan for all nodes even if there are functions that have started transmission and for which a completed status has not been received from the node; the deletion of those functions at the nodes will not be requested
- **S** Deletes all functions of all phases in the plan that have not been started, and sends a delete request for all nodes where the transmission has started but a completed status has not been received. When NetView DM for MVS receives a confirmation from a TME 10 Software Distribution Version 3.1.5 node that a function has been successfully deleted, it will check its records for the plan to see if there are still outstanding functions not yet deleted; if there are none it will delete the plan.

To quit without deleting or forcing, press PF3.

**Confirm the Erasure of Plan:** Type E (Erase) in the command field and press Enter. You can only erase plans that have a status of COMPLETED or DELETED. To quit without erasing, press PF3.

**To Delete One Recursion of a Recursive Plan**
Use this procedure to delete just today's recursion.

Select the plan that you want to delete or erase. There are two ways in which you can do this:

- If you know the recursive plan name, from the Manage Submitted Plans panel type the name of the plan in the Plan name field. Type today’s date in the Recursion date field. Select Option 3, Delete Plan, or Option 11, Erase Plan, then press Enter.

- If you do not know the recursive plan name, you can select a plan from the List of Plans full-list panel. Leave the Plan name field blank, or type a partial name followed by an asterisk (*). Leave the Recursion date field blank and press Enter. The List of Plans full-list panel is displayed.
Type **D** (Delete) or **E** (Erase) next to the plan that you want to select, then press Enter.

Either the Confirm the Deletion of Plan or the Confirm the Erasure of Plan panel is displayed, depending on whether you entered the **D** (Delete) or **E** (Erase) command.

**Confirm the Deletion of Plan:**

- **D** (Delete), **F** (Force) or **S** (Delete at Server) in the command field, then press Enter.

- **D** Deletes all phases in the plan for all nodes for which transmission has not started
- **F** Deletes all phases in the plan for all nodes even if there are functions that have started transmission and for which a *completed* status has not been received; the deletion of those functions at the nodes will *not* be requested
- **S** Deletes all functions of all phases in the plan that have not been started, and sends a delete request for all nodes where the transmission has started but a *completed* status has not been received. When NetView DM for MVS receives a confirmation from a TME 10 Software Distribution Version 3.1.5 server that a function has been successfully deleted, it will check its records for the plan to see if there are still outstanding functions not yet deleted; if there are none it will delete the plan.

To quit without deleting or forcing, press PF3.

**Confirm the Erasure of Plan:**

- Type **E** (Erase) in the command field and press Enter. You can only erase plans that have a status of COMPLETED or DELETED.
  
To quit without erasing, press PF3.

**To Delete a Phase**

You can delete a phase from a non-recursive or a recursive plan, or from a single recursion. The steps are as follows:

1. Select the plan from which you want to delete a phase using one of the methods given above
2. Type **LP** (List Phases) against the phase to be deleted and press Enter. The List of Phases (Plan: <plannname>) panel is displayed.
3. Next to the phase to be deleted type **D** (Delete) and press Enter. The Confirm the Phase Deletion of Plan: <plannname> panel is displayed.

**Confirm the Deletion of a Phase:**

- **D** (Delete), **F** (Force) or **S** (Delete at Server) in the command field and press Enter.

- **D** Deletes the phase for all nodes for which transmission of the phase has not started
- **F** Deletes the phase for all nodes even if there are functions that have started transmission and for which a *completed* status has not been received; the deletion of those functions at the nodes will *not* be requested
- **S** Deletes all functions of the phase that have not been started and sends a delete request to all nodes where the transmission has started but a *completed* status has not been received. When NetView DM for MVS receives a confirmation from a TME 10 Software Distribution Version 3.1.5 server that a function has been successfully deleted, it will check its records
for the phase to see if there are still outstanding functions not yet deleted; if there are none it will delete the phase.

To quit without deleting or forcing, press PF3.

---

**Saving Plans Using GIX**

You can save the contents of plans stored in the TCF into user data sets at the host.

From the Manage Submitted Plans panel, illustrated in Figure 72 on page 149, follow these instructions to save a plan or plan recursion.

1. Leave the **Plan name** field empty, or type a partial name followed by an asterisk (*). Leave the **Recursion date** field empty.

2. Select **Option 12, Save Plan**, then press Enter.

   The List of Plans to be Saved full-list panel is displayed.

3. Type **SA** next to each plan that you want to save, but not erase from the TCF.

4. Type **SE** next to each plan that you want to save, and to erase automatically from the TCF after saving.

5. Type the primary command **SAVE** in the command field and press Enter.

   The Specify Job Attributes panel is displayed. This panel asks you to specify the output data set into which you want to save the plan or plans. An example of this panel is shown in Figure 48 on page 113. There is one additional field that is not defined in this example. This is the **Erase option** field, which is described below.

6. Type either 1 (Yes) or 2 (No) in the **Erase option** field.

7. After you press Enter on the Specify Job Attributes panel, a batch job is submitted to the operating system on your host, and the plans are saved and optionally erased from the TCF.

The additional field on the Specify Job Attributes panel is described below.

**Erase option:** This specifies whether the plan or plan recursion is erased automatically from the TCF when the plan has been saved. The default value for this field depends on whether you entered **SA** or **SE** on the List of Plans to be Saved full-list panel.

---

**Browsing or Printing a Plan Summary Using GIX**

You can browse or print a summary of the status information relating to a specific plan. Follow the instructions below from the Manage Submitted Plans panel, illustrated in Figure 72 on page 149.

1. Select the plan that you want to look at. There are two ways in which you can do this:

   - If you know the plan name, type the name of the plan in the **Plan name** field. If the plan is recursive, type the date of the recursion in the **Recursion date** field. Select **Option 13, Plan Status Summary**, and press Enter.
If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the Plan name field blank, or type a partial name followed by an asterisk (*). Leave the Recursion date field blank, and press Enter. The List of Plans full-list panel is displayed.

Type PS (Plan summary) next to the plan whose summary that you want, then press Enter.

The Plan Summary panel is displayed, shown in Figure 78.

![Figure 78. Example of a Plan Summary panel](image)

This panel shows the current status of the plan and the phases, phase-by-node records, and functions within it. You can refresh the information displayed on this panel at any time, by typing R in the command field and pressing Enter.

2. To print the plan summary, use PF10 to print the information currently displayed on the panel.

### Resetting Plans

You can reset a plan or plan recursion, or you can reset specific phases or phase-by-node records within the plan. To reset phases or phase-by-node records, the plan must contain the following, or you will receive a warning message:

- One or more unsuccessful functions. A function is defined as unsuccessful if it completes with return code 8 and one of the following is true:
  - The function status is COMPLETED.
  - The phase-by-node status is COMPLETED.
  - The phase status is COMPLETED or SCHEDULED.
  - The plan status is SUBMITTED or COMPLETED.

Refer to “Plan Information” on page 148 to find out the status and return codes of plans, phases, phase-by-node records, and function records in the TCF.
• One or more phase-by-node records with a status of HSYST or HRETR.

**Attention**

Your system administrator may have assigned the same logical unit name to more than one node. All of the phase-by-node records that are in HSYST or HRETR status, and which address nodes with the same logical unit name, are released, even if they do not belong to the plan that you are resetting but to a different plan.

When you perform a reset, you can also specify a new scheduling time and date. The new time and date apply only to phase-by-node records with a status of COMPLETED, not HSYST or HRETR.

You can also remove transmission windows that have been defined for a phase by resetting the plan. In this case, you can reset the plan even if the phase is not in HSYST or HRETR status and there are no unsuccessful functions.

For recursive plans, you can reset either the current recursion, or the last completed recursion, if the scheduled date and time of the current plan recursion have not yet been reached.

When you reset the last completed recursion and specify a new scheduling date and time, the date and time you specify must be less than that of the current recursion.

If the plan is recursive, the scheduling date you specify must either be the date of the current recursion, or of the previous recursion. If the date is that of the previous recursion, you must specify a scheduling time that is less than the current recursion. If you do not specify a date and time as described, you will receive an error message.

**Note:** The System/36 fanout capability is not used when you reset a plan.

## Using GIX to Reset Plans

Use the instructions on the following pages at your terminal. From the Manage Submitted Plans panel, illustrated in Figure 72 on page 149, follow these instructions to reset a plan, or plan recursion.

1. Select the plan that you want to reset. There are two ways to do this:
   
   • If you know the plan name, type the name of the plan in the **Plan name** field. If the plan is recursive, type the date of the recursion in the **Recursion date** field. If you do not specify a date for a recursive plan, the current recursion will be reset. Select **Option 14, Reset Plan**, and press Enter.

   • If you do not know the plan name, you can select a plan from the List of Plans full-list panel. Leave the **Plan name** field blank, or type a partial name followed by an asterisk (*). Leave the **Recursion date** field blank, and press Enter. The List of Plans full-list panel is displayed.

   Type **RP** (Reset plan) next to the plan that you want to reset, and press Enter.

   The Reset Plan panel is displayed.
After Resetting a Plan

After you perform a reset, the following occurs before re-execution:

**Functions**

The return code of each unsuccessful function is reset to 0, and the function status becomes NOT STARTED. The plans and phases to which unsuccessful functions belong have their status and return code updated. If the function addresses more than one resource, the function status can be:

- **NOT STARTED**, if all of the function-by-resource records were unsuccessfully completed and must be re-executed. Unsuccessful completion is when the phase status is COMPLETED with return code 12, the phase-by-node status is DELETED with return code 12, and the plan status is SUBMITTED or COMPLETED. In this case, the return code of the function is reset to 0.

- **RESTARTABLE**, if some function-by-resource records were successfully completed (with return code 0 or 4), and do not have to be re-executed. In this
case, the function return code is set to the highest return code (0 or 4) from among the completed function-by-resource records.

**Phase-by-Node Records**
Phase-by-node records which contain unsuccessful functions or which are in HSYST or HRETR status, are given a status of either WAITING or READY (depending on the scheduling time and date). If you specified a new scheduling time and date, this is changed accordingly. On the GIX List of Phases with Nodes panel and the Phase by Node Information panel, the phase-by-node record is marked as either reset in progress, or reset done, and the TCP issues the message NDM0414S.

**Phases with Transmission Windows Defined**
Even if a phase for which transmission windows are defined is not in HSYST or HRETR status and does not contain unsuccessful functions, the phase is reset. The windows are removed from the phase definition for all the destinations involved in the phase. The phase that has been reset changes to READY status and is ready to be executed.

**Conditioned Phases**
If you reset an entire plan, this can affect conditioned phases within the plan. For example, if a plan contains two phases, where Phase 2 is conditioned by Phase 1, and Phase 1 is unsuccessful, Phase 2 and its phase-by-node records complete with return code 12.

- If the entire plan is reset, the unsuccessful functions in Phase 1 are re-executed, and the status of Phase 2 and its return code is also reset. Phase 2 is executed if Phase 1 now completes successfully.
- However, if you only reset Phase 1, and not the entire plan, then only the unsuccessful functions in Phase 1 are re-executed. Phase 2 is left unchanged and none of its functions are executed.

---

**Listing and Controlling Phases By Nodes Using GIX**
You can use GIX to display a selected list of phases by nodes, and then control and recover normal and abnormal situations that occur during the transmission. For example, you can enter commands to control the following TCP transmission activities:

- Release selected nodes
- Change the scheduling date and time for one or more phases by nodes

You do this by first creating a full-list of phases by nodes, built according to a set of selection criteria. After you create the full-list, you can perform functions using line commands.

**Note:** Before you can list or control phases by nodes, the TCP must be active. Refer to Chapter 19, “Starting the Transmission Control Program (TCP)” on page 471 to find out how to start the TCP.
Listing Selected Phases With Nodes

From the Manage Submitted Plans panel, illustrated in Figure 72 on page 149, follow these instructions to build a full-list of phases by nodes.

1. Select **Option 15, List Selected Phases By Nodes**, then press Enter.

   The List Selected Phases By Nodes panel is displayed, illustrated in Figure 80.

   **Note:** When you select **Option 15**, any entries in the **Plan name** or **Recursion date** fields are ignored.

   NDMPJPAE

   LIST SELECTED PHASES BY NODES

   Command ==>

   Enter full-list selection criteria:
   1 Plan name . . . . . . ________
   2 Phase name . . . . . . ________ (if field 1 is specified)
   3 Node name . . . . . . ________ (not together with field 4)
   4 Group name . . . . . . ________ (not together with field 3)
   5 Scheduled date from . ______ (Date format is: YY/MM/DD)
   6 Scheduled time from . ____ (Time format is: HH:MM)
   7 Scheduled date to . . ______ (Date format is: YY/MM/DD)
   8 Scheduled time to . . ____ (Time format is: HH:MM)
   9 Executing status . . Y (“N” to select no display)
   10 Ready status . . . . Y (“N” to select no display)
   11 Waiting status . . . . Y (“N” to select no display)
   12 Held status . . . . . Y (“N” to select no display)
   13 Restartable status . . Y (“N” to select no display)
   14 Pending status . . . . Y (“N” to select no display)

   PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR

   **Figure 80. Example of the List Selected Phases By Nodes panel**

2. Specify the selection criteria that you want to use to build the full-list of phases by nodes. To display all transmission requests in the TCF, press Enter without specifying any selection criteria. You can tailor the list with the selection criteria shown on the panel. The selection criteria you can use on this panel are described below. They are divided into three sets of fields:

   **Fields 1 to 4** Use these fields to specify the plan name, phase name, and the name of the node or the name of a group of nodes (not both), that the phase refers to.

   **Fields 5 to 8** Use these fields to select phases by nodes scheduled within a specific date or time range. You cannot use this date or time range option if you specify a plan name, phase name, or node name.

   **Fields 9 to 14** Use these fields to select phases by nodes that have a particular status.

   a. Specify the plan name and phase name in fields 1 and 2, respectively. Specify **either** a node name **or** a group name in fields 3 and 4, respectively. This will give you a restricted list. Use Table 14 on page 174 to guide your selection.
b. Use fields 4 to 8 to obtain a list of transmission requests scheduled at the specified range of date or time. You can use these fields for all types of requests. Each field is optional. You can enter just one field, or several fields in combination.

- To display all phases by nodes that are scheduled within a specific date range, enter the scheduled dates in the **Scheduled date from** and **Scheduled date to** fields.
- To display all phases by nodes that are scheduled within a specific time range, enter the scheduled times in the **Scheduled time from** and the **Scheduled time to** fields.

c. To select phases by nodes which have a specific status, enter an **N** in the entry field for any status you do not want to display. This suppresses the display.

3. When you have finished, press Enter. GIX passes your request to the TCP, if it is active, using the request queue file. If the TCP is not active, GIX rejects the request and displays a message.

The TCP gathers the information you requested and writes it to a VSAM ESDS file called the *NetView DM for MVS exchange information file* (EIF). This file is referred to by the DD statement called NDMEIF when you start the TCP, and by the ALLOC command in the TSO CLIST for GIX. Refer to Chapter 19, “Starting the Transmission Control Program (TCP)” on page 471 to find out about starting the TCP.

**Note:** GIX may wait up to five minutes for the TCP to gather the information. If this time interval expires, GIX displays a message and considers the request to be unsuccessful.

If the request is successful, the List of Selected Phases With Nodes panel is displayed, illustrated in Figure 81 on page 176.

The fields on the List Selected Phases By Nodes panel are described below.

**Plan name:** The name of the transmission plan that the phase by node belongs to.

**Phase name:** The name of the phase.

**Node name:** The name of the node that the phase is associated with.

**Group name:** The name of the group of nodes that the phase is associated with. You cannot specify a group name if you previously specified a single node name.

<table>
<thead>
<tr>
<th>To Display This Information</th>
<th>Enter These Selections</th>
</tr>
</thead>
<tbody>
<tr>
<td>A single plan</td>
<td>Valid plan name (Field 1)</td>
</tr>
<tr>
<td>A single phase of a plan</td>
<td>Valid plan name (Field 1) and phase name (Field 2)</td>
</tr>
<tr>
<td>All phases for a specific node</td>
<td>Node name (Field 3)</td>
</tr>
<tr>
<td>All phases for a specific group of nodes</td>
<td>Group name (Field 4)</td>
</tr>
</tbody>
</table>

*Table 14. Selecting transmission requests by plans, phases, node, or group name*
Scheduled date from/Scheduled date to: Enter the scheduled dates in the form YY/MM/DD.

If you do not enter a date or time, all the transmission requests will be displayed. If you enter a time but no date, the date defaults to the current date.

Scheduled time from/Scheduled time to: Enter the scheduled times. You can either enter NOW to select the current time, or you can specify a time in the form HH:MM.

The default values are:

- TO-TIME: 23:59
- FROM-TIME: 00:00

If you specify NOW, you do not have to enter a date. Specifying NOW means the current system time and the current date.

Executing status: Phases with nodes have an executing status when the TCP is executing the phase.

Ready status: The ready status applies to phases by nodes that the TCP cannot execute, even though they have passed their scheduled date and time. The TCP cannot execute them because either the maximum number of concurrent SNA sessions, or the involved logical unit is already active.

Waiting status: Phases with nodes that have a waiting status are phases that the TCP will start to process as soon as they reach the scheduled date and time. Waiting phases do not include phases that have passed their scheduled date and time and that are now ready for initiation.

Held status: This status applies either to phases that you held, or that the system held for one of the reasons described below. The status shown for the phase on this panel may be one of the following:

- HELD: Appears if you held phases explicitly using IOF. Sometimes, users may have defined the phase as held when creating the transmission plan.
- HDEST: Appears if you held the node for the transmission request. Nodes are held using the IOF.
- HRETR: The node for the phase has the HRETR status between retries when there are line problems or a node shutdown.
- HSYST: The status becomes HSYST when the TCP reaches the maximum number of retries and still cannot establish transmission.
- HCOND: Appears if the TCP holds a phase because the start of transmission is conditioned by the processing of another phase.
- HWIN: Appears if the TCP holds a phase with transmission windows defined, either because the plan containing the phase was submitted outside of a defined time window, or the phase did not reach pending status by the end of the window.
Controlling Phases By Nodes (End User)

**Restartable status:** This can apply to either of the following:
- Phases that have ended with a temporary error that might disappear if the phase is run again
- Phases that have been interrupted by an Immediate Quiesce command from the IOF

**Pending status:** This status applies to phases that contain at least one function in pending status. A function is in a pending status when NetView DM for MVS has delivered it to a node and is waiting for a delayed acknowledgment or data from the node.

### Controlling Phases By Nodes

You can change the transmission schedule for your selected phases by nodes using the List Of Selected Phases By Nodes By Selection panel. When you have completed the selection criteria, press Enter to display the panel shown in Figure 81.

<table>
<thead>
<tr>
<th>CMD</th>
<th>PLAN</th>
<th>PHASE</th>
<th>NODE</th>
<th>SCHEDULE</th>
<th>STATUS</th>
<th>HRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>___</td>
<td>PLANGOOD</td>
<td>PHAS0001</td>
<td>POSC0001</td>
<td>900822 11:31</td>
<td>HELD</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>PLANGOOD</td>
<td>PHAS0001</td>
<td>POSC0002</td>
<td>900822 11:31</td>
<td>WAITING</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>PLANTEST</td>
<td>PHTEST01</td>
<td>NODE01</td>
<td>900822 11:45</td>
<td>HSYST</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>PLANTEST</td>
<td>PHTEST01</td>
<td>NODE02</td>
<td>900822 11:45</td>
<td>READY</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>PLANTEST</td>
<td>PHTEST02</td>
<td>NODE01</td>
<td>900822 12:02</td>
<td>HSYST</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>PLANTEST</td>
<td>PHTEST02</td>
<td>NODE02</td>
<td>900822 12:02</td>
<td>READY</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>SENDPLAN</td>
<td>SEND01</td>
<td>VMNODE01</td>
<td>900822 08:30</td>
<td>READY</td>
<td>0</td>
</tr>
<tr>
<td>___</td>
<td>SENDPLAN</td>
<td>SEND02</td>
<td>VMNODE02</td>
<td>900822 08:30</td>
<td>HRETR</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 81. Example of the List of Selected Phases By Nodes By Selection panel

The top of the List of Selected Phases By Nodes By Selection panel shows you a list of line commands that you can use. These commands are:

**B (Browse)**  
Browse information about a particular phase by node.

**P (Print)**  
Print information about a phase by node.

**LF (List Functions)**  
List the functions of a phase by node.

**C (Change schedule date and time)**  
Change the start date and time of a phase by node.

**RN (Release node)**  
Release the transmissions for a node.

**RP (Reset phase by node)**  
Reset a phase by node.
You can also issue the following primary command:

**R (Refresh listed items)**

Updates the information about the phases by nodes that are currently displayed.

The following pages describe these commands:

- C (Change schedule date and time)
- RN (Release node)
- RP (Reset phase by node)
- R (Refresh listed items)

The Browse and Print commands for phases by nodes are described in “Phase by Node Information” on page 154. The List Functions command is described in “Function Information for LU0 Resources” on page 156.

**C–Change Schedule Date and Time**

This command changes the scheduled start date and time, or both, of phases by nodes.

From the List Of Selected Phases By Nodes By Selection panel, shown in Figure 81 on page 176, follow the instructions below.

1. Enter the line command C in the CMD column next to the phase by node.
2. Press Enter. The Change Schedule Date And Time panel is displayed, providing that GIX accepts the command. GIX accepts the command if:
   - The phase by node does not contain send or initiate functions to multiple end nodes connected via an intermediate node
   - The current phase by node status has one of the following statuses:
     - HELD
     - HDEST
     - HSYST and not Pending
     - HRETR and not Pending
     - HCOND
     - RESTARTABLE
     - WAITING
     - READY
Figure 82. Example of the Change Schedule Date and Time panel

3. Enter the new values for the scheduled date and time, then press Enter. GIX performs the following checks on the scheduled date and time values:

- For recursive plans only, whether the date and time you entered are beyond the expiration date and time of the phase
- Whether the date and time you enter are beyond the cut-off date and time

If the scheduled date or time is invalid, GIX displays an error message. If your command is accepted, the contents of the line referred to by the command is updated.

The actual change is not performed by GIX, but by the TCP. The TCP does not acknowledge the result of the execution of the request. The result is shown in a message that also includes the ID of the requester. The TCP writes this message to the SYSPRINT file.

RN–Release Node

This command releases the transmissions for the node associated with the phase by node that you specify, providing the current status of the node is either one of the following:

- HSYST
- HRETR

From the List Of Selected Phases With Nodes By Selection panel, shown in Figure 81 on page 176, follow the instructions below.

1. Enter the line command RN in the CMD column next to the phase by node.

2. Press Enter. A response to your command appears at the end of the full-list row. If your command is accepted, the contents of the line referred to by the command are updated.
The actual change is not performed by GIX, but by the TCP. The TCP does not acknowledge the result of the execution of the request. The result is shown in a message that also includes the ID of the requester. The TCP writes this message to the SYSPRINT file.

**RP–Reset Phase By Node**

This line command resets a phase by node. It works in the same way as the Reset Plan command, except that the command acts on a single phase by node instead of a complete plan.

There are only two situations in which you can effectively issue this command:

- When the status of the phase by node is COMPLETED with a return code 8. Before this situation can occur, you must issue a Refresh Listed Items command (described in “R–Refresh Listed Items” on page 180).
- When the status of the phase by node is HSYST or HRETR. In this situation, this command releases the transmissions for the node associated with the phase by node you select, just like the Release node command described in “RN–Release Node” on page 178.

From the List Of Selected Phases With Nodes panel, shown in Figure 81 on page 176, follow the instructions below to reset a phase by node.

1. Enter the line command **RP** in the CMD column next to the phase by node.
2. Press Enter. The Reset Phase By Node panel is displayed.

   **RESET PHASE BY NODE**

   Command:

   Enter new date/time or blank:

   Date : ________ Schedule date (optional)
   YY/MM/DD
   Time : _____ Schedule time (optional)
   HH:MM

   Tracking information:
   Plan name . . . ________
   Phase name . . ________
   Destination . . ________
   Recursion date. ________

   PF 1=HELP 2=SPLIT 3=END 4=RETURN
   9=SWAP 10=PRINT 12=CURSOR

   Figure 83. Example of the Reset Phase By Node panel

3. Enter the new values for the schedule date and time.
4. Press Enter. The contents of the line referred to by the command is updated.
R–Refresh Listed Items

This primary command requests GIX to update the information about the status, the schedule date and time, and the return code for the items (phases by nodes) of the list currently displayed.

Phases by nodes that have a status of COMPLETED or DELETED may be displayed after you have entered a Refresh Listed Items command, even if the initial list does not contain phases by nodes with a status of COMPLETED or DELETED.

To refresh the listed items on the List of Selected Phases By Nodes panel, enter the primary command R on the command line, then press Enter. The information shown for the phases by nodes is updated.

Recovering from Transmission Failures

A transmission error occurred if the status of a plan, phase, or function is COMPLETED but with a return code other than zero, or if the status is RESTARTABLE.

Follow this sequence of steps to correct a transmission error:

Step 1. Identify the functions that did not execute successfully. You can use GIX to browse or print plan status information, described in “Looking at TCF Information Using GIX” on page 147. You can either check the functions of a single plan, or you can list all the functions of all plans that were not executed successfully.

Step 2. Correct whatever the cause of the transmission failure was. The rest of this section shows how to correct the three main types of error, which are host errors, node errors, and VTAM errors.

Step 3. Resubmit the transmission plan to the TCF, to be executed again. You can either resubmit the entire plan (you will probably need to specify a new phase scheduling date and time). Or, if most of the functions in the plan or phase executed successfully and you do not want to execute them again, you can use the old plan in the library to build a new plan. Delete all the functions that executed successfully from the plan, and then submit the new plan to the TCF.

There are three main types of transmission errors:

- NetView DM for MVS host errors
- Node errors
- VTAM errors

Table 15 shows the possible transmission function return codes for these errors.

<table>
<thead>
<tr>
<th>RC</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Function completed successfully.</td>
</tr>
<tr>
<td>04</td>
<td>Function completed, but warning messages were issued (for example, a request to delete a data set was issued for a data set that does not exist, or a request to send an existing resource to the node). The requested operation was performed.</td>
</tr>
</tbody>
</table>
Table 15. Transmission function termination return codes

<table>
<thead>
<tr>
<th>RC</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| 06 | Function was not completed due to a temporary error. For example:  
  • An LU0 resource was not found in the holding file. If an LU 6.2 resource is not found in the holding file, it causes transmission to end with return code 08.  
  • A VSAM error occurred in the holding file.  
  • A printer was not available at a node for a Send Print function.  
  The function is restartable. If the phase by node that the function belongs to is marked RESTARTABLE, an IOF restart command applied to this phase by node will cause the re-execution of the failing function.  
  If the phase by node that the function belongs to is not marked COMPLETED or DELETED, then the failing function will be re-executed the next time the TCP is started, and the function then would be completed successfully. For example:  
  • Data set was not found in the holding file.  
  • A VSAM error occurred in the holding file.  
  • Printer was not available at a node for a Send Print function. |
| 08 | Function completed with an error that is not temporary. The function will not be run again, and is therefore not restartable, for example, a logical error exists in the plan. This return code is also issued when an LU 6.2 resource is not found in the holding file. |
| 12 | This return code is for phase by node functions. It means that the phase by node was deleted because the phase conditioning criteria were not met. |
| 20 | This return code is for phases only. It means that the phase was deleted, either using GIX or with an IOF command. This return code is also issued when NETCHNG=IMMEDIATE and the phase is addressing a node that has been deleted, or the logical unit or connection profile associated with the node has been changed. |
| 24 | This return code is for phases only. It means that the phase was deleted because the cut-off time or interval has passed. |
| VSE only | For phases which access VSE nodes, the function return codes have the following meanings:  
  • 00 Function completed successfully.  
  • 08 Function was not completed successfully. The various condition codes produced at the node are replaced with return code 08 by NetView DM for MVS. |

Table 15. Transmission function termination return codes

NetView DM for MVS Host Errors

The Function Information panel shows you the error message returned by the TCP. Consult the Messages and Codes book to see if the information there helps you to correct the error at the host. For example, if the TCP error message is 26, you will find this description in the Messages and Codes book:

NDM0026E SEND-WITHOUT-REPLACE WAS SPECIFIED FOR RESOURCE resource name  
BUT THE RESOURCE ALREADY EXISTS AT NODE node identifier

In this case, the book tells you to either delete the resource that already exists at the node, or specify the replace option for the resource you want to send.
Node Errors (End User)

VTAM Errors

The Function Information panel shows how errors caused by problems with VTAM, or at the node, are identified by return codes as well as error messages. The TCP error message identifies whether the primary and secondary return codes are from the node, or from VTAM. For example, this feedback on the panel is for a VTAM error:

<table>
<thead>
<tr>
<th>TCP error message</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return codes:</td>
<td></td>
</tr>
<tr>
<td>Primary . . . . . : 000C  Secondary . . . : 000B</td>
<td></td>
</tr>
<tr>
<td>Sense codes:</td>
<td></td>
</tr>
<tr>
<td>User . . . . . . : 0000  System . . . . : 0000</td>
<td></td>
</tr>
</tbody>
</table>

Figure 84. Example of VTAM error feedback

The TCP message, NDM0020E, provides the following information:

NDM0020E CONNECTION LOST WITH LOGICAL UNIT unit name, TP ACCESS METHOD RPL RETURN CODE 12, RPL FEEDBACK 11.

From the message text, you know that the return codes refer to VTAM and not the node. Consult the VTAM manual Advanced Communication Function for VTAM Programming, SC27-0449 and SC23-0115, for more information. The NetView DM for MVS operator can use this information to restart transmission, and resend the phases that were not completed.

Node Errors

Nodes send different types of information when an error occurs, depending on the type of node:

- All nodes send primary and secondary return codes.
- DPPX nodes also issue error messages.
- Other node types also issue sense codes.

This is an example of information about an error at a node:

<table>
<thead>
<tr>
<th>TCP error message</th>
<th>23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return codes:</td>
<td></td>
</tr>
<tr>
<td>Primary . . . . . : 020C  Secondary . . . : 0F56</td>
<td></td>
</tr>
<tr>
<td>Node error message</td>
<td>HDKF0039</td>
</tr>
</tbody>
</table>

Figure 85. Example of node error feedback

The TCP message is as follows:

NDM0099E WHILE PROCESSING function-type FOR resource, THE TCP RECEIVED FROM LOGICAL UNIT unit-name, A PRIMARY RETURN CODE 020C, SECONDARY RETURN CODE 0F56, MESSAGE NUMBER HDKF0039

This tells you that the return codes refer to a node error. For more information about return codes and messages from nodes, refer to the node publications listed in the bibliography of this manual.
After resources have been transmitted and received, you must unload the resources from the resource repository into external data sets. Unloading resources is the final step in the following sequence of tasks:

**Step 1.** Load resources into the resource repository
**Step 2.** Prepare and submit a transmission plan
**Step 3.** Work on submitted plans
**Step 4.** Unload resources from the resource repository

This chapter shows you how to unload resources after they have been transmitted and received.

### Unloading Resources from the Repository

Your transmission plans can retrieve data sets or dumps from nodes, so that they can be processed at the host. NetView DM for MVS stores the resources that it retrieves in the resource repository. You must unload (or convert) the resources from the repository into an external data set, before the resources can be processed using user application programs on the host. You can use GIX to unload resources.

You unload a resource stored in the repository into an output file. The output file can either be a physical sequential file (PS), or a VSAM KSDS or ESDS file (for LU0 resources only). LU 6.2 resources can be converted into a sequential file.

**Converting Fixed-Length Resources:** If a resource is stored with fixed-length records, the output file will consist of fixed-length records, if the output file has RECFM=F, or variable-length records if the output file has RECFM=V. The format of the records depends on the environment.

**Converting Variable-Length Resources:** You can also convert resources that have variable-length records. The repository directory stores information about whether a resource has variable-length, or fixed-length, records. You must specify RECFM=V for the output user data set when it is allocated, and also in the DD statement.

If these conditions are met, the output user file will consist of records having one of the formats shown in Figure 42 on page 101. If you do not meet these conditions and the output file was defined with a fixed record format, processing is terminated with return code 8, and an error message is issued.

**Converting Multiple Resources:** You can specify groups of data sets or groups of dumps to be converted, and concatenated into a single output file on the host. This happens if you specify one FCTLNAME and/or GROUPID, or if you specify ID=(*name1,name2,...*). If you want converted data sets to be kept separate, you should use separate control statements, with a different output user data set for each one.
When you are converting multiple resources and the output file has RECFM=F, if a variable-length resource is found, the processing of the control statement terminates with a return code 8, and an error message is issued.

Unloading into Physical Sequential Files
A physical sequential output file can be a member of a partitioned data set, a GDG data set, or a sequential data set of any device type. It can have any record format: fixed, variable or undefined, blocked or unblocked. The maximum record length is 32768 bytes. It can have ANSI control characters, but does not support machine code control characters.

You must specify the characteristics of the file, either in the DCB parameter of its DD statement, or in the label of an existing file. The characteristics include RECFM, LRECL, and BLKSIZE. If you omit any required DCB parameter, the job will abend with return code 013. See the Messages and Codes book.

If you are unloading a resource stored as a variable-length record, each record is unloaded into the output file with its RDW, if the output file is physical sequential with RECFM=V or RECFM=VB. You cannot unload variable-length resources into physical sequential files that do not have a record format of V or VB.

You must also specify a DISPOSITION parameter. This parameter determines whether the output will replace an existing data set at the host. If the data set is sequential, new records are either appended to the end of the file, or they replace existing records. If the data set is partitioned, the existing member is replaced. You should always specify DISP=(OLD) for partitioned data sets, even if the member already exists. DISP=(MOD) causes an abend if the member exists.

When NetView DM for MVS converts a resource, it discards the header. For variable-length output records (for PS files only), the first four bytes of the record are a *record descriptor word*. Each record of the output file corresponds to one logical input record, with the following exceptions. A warning message is issued if any of these conditions occur:

- If the output file has ANSI control characters, but the input data set does not. In this case, a blank (ANSI control character for a single space) is appended to each logical record.
- Logical input records that are longer than the maximum output record length are truncated on the right. A warning message is issued, and the function ends with return code 4.
- Logical input records that are shorter than the fixed-length output records are padded on the right. The padding character is a blank, if either the input data set or the output file has ANSI control characters. Otherwise, the padding character is a binary zero.

Unloading into VSAM Files
You can unload into a VSAM entry-sequenced (ESDS), or key-sequenced (KSDS) file. If the output file is a VSAM KSDS or ESDS file, the record is unloaded into the output file without a RDW.

The characteristics of a VSAM output data set are specified when the data set is defined at the host. These include the maximum record length, and, for KSDS files, the key length and its offset. When the output file is KSDS, the input records
must be in ascending key sequence. If the output file already contains records, new records are appended to the end of the file (ESDS), or inserted in key sequence (KSDS).

**Unloading Resources with Uniform Text (LU 6.2)**

NetView DM for MVS handles resources that have a uniform text format in a special way. This is known as *data translation*. A uniform text is a text data object (resource) that contains only characters belonging to an identified graphic character set. For complete information, refer to “Uniform Text Processing” on page 263.

When unloading the resource from the NetView DM for MVS resource repository to a user file, the data is translated from the Coded Character Set Identifier (CCSID) associated with the resource to the NetView DM for MVS CCSID (defined by the NDMCOM customization macro when customizing NetView DM for MVS).

Besides the source and target CCSID shown in “Coded Character Set Identifier (CCSID)” on page 263, you can manage SBCS CCSIDs by using the ADDTABLE utility at installation time. For more information on ADDTABLE, see the section on using utilities in the *Installation and Customization* book.

You can also unload the double-byte character set (DBCS) and mixed uniform text resources supported by DFSMS 1.2 with APAR AW14950 or higher.

**Unloading Resources Using GIX**

When you use GIX to unload resources, you can only convert one data set at a time. For LU0 resources, you can only convert data set resources, not dumps. You can unload any LU 6.2 resource.

The DCB of the output data set will not have the same logical record length and block size as the data set in the repository. The LRECL is always 80. The BLKSIZE is always 800.

To unload resources from the repository using GIX, follow these instructions:

**Step 1: Select Resources of a Specific Type**

1. Select **Option 3, Manage Resources**, from the GIX main menu.

   The Manage Resources panel is displayed, as illustrated in Figure 44 on page 107.

2. Select the resource type of the resources you want to unload from the resource repository. Type in the number of the resource type in the selection field. This lists all the resources of this type that are held in the repository. Press Enter. The List of Selected Resources full-list panel is displayed. Go to “Step 3: Convert Resources from the Repository” on page 186.

3. Optionally, if you want to use additional selection criteria to build a list, type 1 in the field which asks:

   Do you want specific resources of the type selected?

   Press Enter. The Specify Resource Selection Criteria panel is displayed.
Step 2 (optional): Specify Resource Selection Criteria

Use the Specify Resource Selection Criteria panel to narrow down the list of resources you want to display. This panel displays various fields, depending on the resource type you are working with. The fields represent resource identifiers, such as resource name. Resource identifiers identify resources at the host. The fields which may be displayed on this panel are described in “Step 2 (optional): Specify Resource Selection Criteria” on page 107.

Type values (or partial values followed by an *) in any or all of the fields, then press Enter. The List of Selected Resources full-list panel is displayed.

Step 3: Convert Resources from the Repository

```
NDPA3NE   LIST OF SELECTED RESOURCES (SOFTWARE)  Scroll ==>PAGE
Command ==>  17:21

Primary commands: D(Delete)  P(Print directory)
B(Browse directory)  DAL(Delete all)
I(Insert)  CNV(Convert)

Line commands: D(Delete)  B(Browse directory)  CNV(Convert)

CMD RESOURCE NAME        LOCK   COMPLETENESS
__  PROVA.DEL            N     Y
__  PROVA.01             N     Y

*********************************************************************************

Figure 86. Example of a List of Selected Resources panel (LU0)

This panel lists the resources that are currently in the resource repository. To unload a resource from the resource repository into a user data set, use the CNV (Convert) primary command.

1. Enter the primary command CNV (Convert) in the command field of the List of Selected Resources panel, then press Enter.

   The Identify the Resource panel is displayed. This panel displays various fields, depending on the resource type you are working with.
Unloading Resources Using GIX (End User)

Figure 87. Example of an Identify the Resource panel (LU 6.2)

2. Type values in any or all of the fields, then press Enter.

The Specify Attributes panel is displayed.

The fields on the Identify the Resource panel are described below.

**Resource name:** Enter the name of the resource that you want to unload. For LU 6.2 resources, you can specify a partial name. The format of resource names is described in “Naming Resources” on page 65 and also in Appendix C, “Resource Naming Conventions” on page 569.

**Matching indicators:** You can use this field when you are unloading resources from the repository to identify the resource to be unloaded. For example, suppose you specify a partial name A.B.C.D.* and there are two resources in the repository with similar names:

- A.B.C.D.E
- A.B.C.D.X

If you set the matching indicator to the *Low* value (by entering L), the MAINTDAT utility unloads the resource called A.B.C.D.E.

If you set the matching indicator to the *High* value (by entering H), the MAINTDAT utility unloads the resource called A.B.C.D.X.

The *Ignored* value (I), means that the * token can have any value.
Step 4: Specify the Job Attributes

<table>
<thead>
<tr>
<th>NDMPAX2E</th>
<th>SPECIFY ATTRIBUTES (SOFTWARE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>==&gt;</td>
</tr>
<tr>
<td>14:09</td>
<td></td>
</tr>
</tbody>
</table>

Enter desired values or accept the ones shown:

1. Job name . . . . . . . GIXJOB__
2. Data set name . . ==> ===
3. Unit . . . . . . . . ____  Enter if data set name is not cataloged
4. Volume . . . . . . . . ____  Enter if data set name is not cataloged
5. Label sequence number ____  Enter if data set name is on tape
6. Label type . . . . . . _ Entries: AL AUL BLP LTM NL NSL SL SUL
7. Disposition . . . . NEW  Entries: NEW OLD
8. Scratch . . . . 2 1 = Yes 2 = No

Tracking information:
- Activity . . . : CONVERT
- Resource name .:

PF 1=HELP 2=SPLIT 3=END 4=RETURN 9=SWAP 10=PRINT 12=CURSOR

Figure 88. Example of a Specify Attributes panel (LU 6.2). This is an example for unloading Microcode from the resource repository. The actual fields shown depend on the specific resource type.

On this panel, you enter information that identifies the batch job that will execute the convert operation.

Type in the required values and press Enter. GIX will create the job, and submit it to your operating system. You can then use ISPF/PDF to monitor the progress of the job (see “Going to ISPF from GIX” on page 15).

The fields on the Specify Attributes panel may display any of the following fields for unloading resources, depending on the resource type.

**Job name:** This field is mandatory for all resource types. Enter the name of the batch job that you are creating using GIX. The field is automatically filled in with the job name specified in your JCL cards. You can use the job name to track the progress of the batch job (refer to “Going to ISPF from GIX” on page 15).

**Data set name:** Enter the name of the user data set into which you want to unload the resource. You can enter from 1 to 54 alphanumeric characters, including periods. This must be a physical sequential data set. If the convert process cannot be started (for example, if no resource is found in the resource repository), NetView DM for MVS does not open and close the output dataset.

The following attributes are used to specify how and where the data set is stored at the host:

**Unit:** Enter the type of unit that the data set is stored on at the host. You can specify TAPE or DISK.

**Volume:** Enter the volume serial number that locates the resource in the host data set. You can specify from 1 to 6 alphanumeric characters.
**Label sequence number:** If the unit is a tape, this attribute identifies the sequential number of the resource on the tape. Specify any one- to four-digit number.

**Label type:** If the unit is a tape, this attribute specifies the type of label that identifies the resource. The values can be:

- **AL** American National Standard labels
- **AUL** American National Standard and user labels
- **BLP** Bypass label processing
- **LTM** Leading tapemark
- **NL** No labels
- **NSL** Nonstandard labels
- **SL** Standard labels
- **SUL** IBM standard and user labels

**Disposition:** This attribute tells NetView DM for MVS whether the resource is replacing an existing data set, or creating a new data set in the host file. The disposition values can be:

- **OLD** The data set is to replace an existing data set of the same name.
- **NEW** The data set is a new one to be inserted in the host file.

**Scratch:** Use this field to specify whether you want the resource to be deleted from the repository after it has been unloaded. You can specify 1, which means Yes, or 2, which means No (the default value).
Chapter 10. Preparing Messages

There is one type of message that you can send and receive using GIX. These messages are prepared at the host to send to nodes, except user-defined nodes with CMFP and CMEP functional capabilities.

You can send messages to all types of node. Messages are stored in queues in the holding file. Each queue is identified by a file control name. When you send messages to nodes, you identify the queue that they belong to by specifying the file control name. If there is more than one message in the queue, all the messages are sent.

Preparing Messages Using GIX

You can use GIX to:

- Queue a message, either by adding a new message to an existing queue or by creating a new queue and inserting the message into it
- Browse a list of all the messages in a queue along with the message text
- Delete a queue and all the messages in it

Queueing a Message

Queueing a message means to define the message in the holding file, by giving it a file control name.

Use the following instructions at your terminal, after logging on to the GIX main menu.

1. Select Option 6, Handle Messages, from the GIX Main Menu.

   The Message Type Selection panel is displayed.

   **Note:** This option does not apply to nodes with CMFP and CMEP functional capabilities.

2. Select Option 3, and press Enter.

   The Message Management panel is displayed. There are three ways in which you can use this panel:

   a. You can add the message to an existing queue. Type the file control name of the queue in the File Control Name field. Select Option 1, and press Enter.

   b. You can add the message to an existing queue, without knowing the queue's file control name. Leave the File Control Name field and the selection field blank, and press Enter. The List of Message Specifications full-list panel is displayed.

      Use the Q (Queue) command to specify the queue to which you want to add the message.

   c. You can create a new queue for the message. Type in a new file control name for the queue. Select Option 1, then press Enter.

   The Enter Text of Output Message panel is displayed.
Preparing Messages Using GIX (GIX)

3. Type in the message text, then press Enter to create the message.
   To quit without creating the message, press PF3.

Browsing Messages
To browse the messages that are in the resource repository, follow these steps:
1. Select **Option 6, Handle Messages**, from the GIX main menu.
   
   The Message Type Selection panel is displayed. On this panel, you specify the messages that have been prepared and can be sent to the nodes

2. Select **Option 2**, and press Enter, or use the **B** (Browse) command.
   
   The Message List panel is displayed. This shows the contents of all the messages in the queue.

Deleting Messages
To delete one or more messages from the resource repository, follow these steps:
1. Select **Option 6, Handle Messages**, from the GIX main menu.
   
   The Message Type Selection panel is displayed.

   **Note**: This option does not apply to nodes with CMFP and CMEP functional capabilities.

2. Select **Option 3**, then press Enter.
   
   The Message Management panel is displayed.

3. Select **Option 3**, then press Enter, or use the **D** (Delete) command on the full-list panel.
   
   The Confirm Deletion of Message panel is displayed.

4. Type **D** (Delete) in the command field, then press Enter to confirm the deletion.
   To quit without deleting, press PF3.
Part 2. Using the Batch Utilities

Note: Not all of the attached nodes support the parameters that NetView DM for MVS can generate. For specific limitations, please refer to the appropriate node documentation.

Chapter 11. Getting Started

What Are the Batch Utilities? ............................ 197
How to Define Control Statements
  Control Statement Label ........................................ 198
  Verb and Modifier ............................................ 199
  Operands ....................................................... 199
  END Statement ................................................. 200
General Rules for Using JCL with Control Statements for Batch Utilities .... 200
What Could Go Wrong .......................................... 200
How to Run the Batch Utilities
  How to Run MAINTDAT ..................................... 203
  How to Run SUBMIT ........................................ 204
  How to Run TCFMAINT .................................... 206
  How to Run PLCOPY ....................................... 208
  How to Run DSXBUBK and DSXBURS ........................ 208
General Purpose Control Statements ................................ 209
COPY .............................................................. 209
DUMP .............................................................. 209
EJECT .............................................................. 209
END ............................................................... 210
SPACE ............................................................ 210
TITLE ............................................................. 210

Chapter 12. Defining User Profiles

What Is a User Profile? ..................................... 211
SUBMIT Control Statements for User Profiles ................. 211
  CHANGE USER/DEFINE USER ............................ 212
  DELETE USER/PRINT USER ............................... 214

Chapter 13. Defining Nodes

What Is a Node Definition? ................................ 215
Node Types .................................................... 215
SUBMIT Control Statements for Nodes ......................... 217
  CHANGE NODE/DEFINE NODE ............................ 217
  DELETE NODE ................................................. 223
  PRINT NODE ................................................. 224
Organizing Nodes and Resources into Groups ................. 225
  When to Use Node Groups of the Same Type .............. 225
  Group of LU 6.2 Nodes with Different Node Types ....... 225
  When to Use Server Node Groups ........................ 225
  When to Use Resource Groups ............................ 227
SUBMIT Control Statements for Groups ....................... 227
  DEFINE GROUP/DELETE GROUP ........................... 228
  PRINT GROUP ................................................. 231
Using Transmission Profiles ................................ 232
Using Connection Profiles .................................. 232
### Chapter 14. Defining and Assigning Resources

**What Is a Resource Definition?**  
- Resource Types .......................................................... 240
- Naming Resources .......................................................... 243
  - Naming LU0 Resources .................................................. 243
  - Naming LU 6.2 Resources .............................................. 243
  - Partial Naming and Matching for LU 6.2 Resource Types ...... 244
- Tracking Resources ....................................................... 247

**SUBMIT Control Statements for LU0 Resource Types** .................. 248
- ADD Statements ............................................................ 249
- DEASSIGN Statements ................................................... 251
- DELETE DEFINITION ....................................................... 252
- PRINT DEFINITION .......................................................... 253

**SUBMIT Control Statements for LU 6.2 Resource Types** .............. 254
- ADD RESOURCE ............................................................. 254
- DEASSIGN RESOURCE ..................................................... 255
- DEFINE RESOURCE ....................................................... 256

**SUBMIT Control Statements for Tracking Resources** .................. 257
- DELETE HISTORY ........................................................... 257
- PRINT HISTORY ............................................................ 260

### Chapter 15. Managing Resources in the Repository

**Loading Resources into the Resource Repository** ..................... 261
- Loading Resources from Variable-Length Records .................. 262
- Uniform Text Processing ................................................. 263
- Using Primary and Embedded Statements .............................. 265
- Locking Resources in the Resource Repository ..................... 266

**Unloading Resources from the Repository** ................................ 267
- Using the DISP Operand ................................................... 268

**MAINTDAT Control Statements for All Resource Types** ............... 269
- CONVERT RESOURCE ....................................................... 269
- ERASE RESOURCE ........................................................... 273
- PREPARE RESOURCE ....................................................... 277
- PRINT DIRECTORY .......................................................... 280
- PRINT RESOURCE ............................................................ 281

**MAINTDAT Control Statements for Specific Resources** ............... 284
- CONVERT DATASET/DUMP .................................................. 284
- INSERT/REPLACE CLIST ................................................... 286
- INSERT/REPLACE JOB ..................................................... 287
- LOAD RESOURCE ............................................................ 288
- PREPARE CLIST/JOB ....................................................... 294
- PREPARE DATASET/PRINT ................................................ 296
- PREPARE MESSAGE .......................................................... 303
- PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB ......................... 304
- QUERY SPACE ............................................................... 305
- UNLOAD RESOURCE .......................................................... 306
- ULOCK RESOURCE ........................................................... 311
Chapter 16. Preparing and Submitting Transmission Plans .......................... 313
What Is a Transmission Plan? .............................................................. 313
Recursive Plans ................................................................................. 313
Phases ................................................................................................. 314
Functions ............................................................................................. 314
Conditioning within a Phase ............................................................... 315
Conditioning between Phases ............................................................. 315
Defining a Transmission Plan ............................................................... 317
Submitting a Transmission Plan .......................................................... 318
SUBMIT Control Statements for Transmission Plans .......................... 319
DEFINE PLAN ....................................................................................... 319
DEFINE PHASE .................................................................................... 321
DELETE PLAN ....................................................................................... 324
QUERY SPACE ...................................................................................... 325
SUBMIT PLAN ....................................................................................... 326
SUBMIT Transmission Functions for LU0 Node Types ......................... 327
DELETE ................................................................................................. 328
INITIATE ................................................................................................. 330
RECEIVE/RETRIEVE ........................................................................... 332
SEND/REPLACE ................................................................................... 337
SUBMIT Transmission Functions for LU 6.2 Node Types ....................... 343
ACCEPT MICROCODE ........................................................................... 346
ACCEPT RESOURCE ............................................................................ 347
ACTIVATE NODE .................................................................................. 349
DELETE MICROCODE .......................................................................... 351
DELETE RESOURCE (from node) .......................................................... 353
INSTALL MICROCODE .......................................................................... 359
INSTALL RESOURCE ............................................................................ 364
REMOVE MICROCODE ......................................................................... 371
REMOVE RESOURCE ............................................................................. 373
RETRIEVE MICROCODE ....................................................................... 376
RETRIEVE RESOURCE .......................................................................... 378
SEND MICROCODE ............................................................................... 383
SEND RESOURCE .................................................................................. 385
UNINSTALL COMPONENT ...................................................................... 392

Chapter 17. Managing Submitted Plans .................................................. 395
Printing the Contents of a Plan ............................................................. 395
Plan Status Information ........................................................................ 395
Phase Information ................................................................................ 396
Phase by Node Information .................................................................... 397
Function Information for LU0 Resources ............................................... 400
Function Information for LU 6.2 Resources ........................................... 402
Recovering from Transmission Failures ................................................ 405
NetView DM for MVS Host Errors ......................................................... 406
VTAM Errors ......................................................................................... 406
Node Information .................................................................................. 407
Managing Node Solicited Requests and Node Unsolicited Reports .......... 408
TCFMAINT Control Statements ............................................................ 409
DELETE SPLAN ...................................................................................... 409
When You Delete a Recursive Plan ....................................................... 410
When You Delete a Plan Recursion ....................................................... 410
When You Delete a Specific Phase ....................................................... 410
<table>
<thead>
<tr>
<th>When You Delete a Plan or Phase Using the Delete at Server Option</th>
<th>410</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERASE APIQENT</td>
<td>412</td>
</tr>
<tr>
<td>ERASE PLAN</td>
<td>413</td>
</tr>
<tr>
<td>ERASE REQUEST</td>
<td>414</td>
</tr>
<tr>
<td>PRINT APIQENT</td>
<td>415</td>
</tr>
<tr>
<td>PRINT PLAN</td>
<td>416</td>
</tr>
<tr>
<td>PRINT REQUEST</td>
<td>417</td>
</tr>
<tr>
<td>RESET PLAN</td>
<td>418</td>
</tr>
<tr>
<td>SAVE APIQENT</td>
<td>420</td>
</tr>
<tr>
<td>SAVE PLAN</td>
<td>421</td>
</tr>
<tr>
<td>SAVE REQUEST</td>
<td>422</td>
</tr>
</tbody>
</table>

Erasing Plans Using the DB2 Environment .................................. 423

Chapter 18. Maintaining NetView DM for MVS Files ........................ 429
Procedure for Maintaining NetView DM for MVS Files ...................... 429
Maintaining the Resource Repository ....................................... 430
  Space Requirements .................................................... 430
  Using Extended Addressability ........................................ 431
  What to Do If the File Is Full ....................................... 431
  Reorganizing the Library or Holding File Without Using Extended
  Addressability .......................................................... 432
  Reorganizing the Library or Holding File Using Extended Addressability 437
  Fast Reinitialization Using DSXBIHLF ................................ 440
  DSXBUBK and DSXBURS .................................................. 442
Maintaining the Plan Library .................................................. 444
  Space Requirements .................................................... 444
  PLCOPY ........................................................................... 447
  DUPLICAtE PLANS ......................................................... 448
Maintaining the Distributed Resource Directory ................................ 449
  Space Requirements .................................................... 450
Maintaining the TCF ................................................................ 452
  Space Requirements .................................................... 453
Maintaining the Unsolicited Message File .................................... 457
  Space Requirements .................................................... 457
Maintaining the NetView DM for MVS Work File ............................... 458
  Space Requirements .................................................... 458
Maintaining the Exchange Information File (EIF) ............................ 459
  Space Requirements .................................................... 459
Maintaining the TCP Work File ................................................... 460
  Space Requirements .................................................... 460
Maintaining the Request Queue File ............................................ 462
  Space Requirements .................................................... 462
Maintaining the GIX Data Sets .................................................. 464
  Skeletons Library ........................................................ 464
  CLISTs Library ........................................................... 464
  GIX Edit Data Sets ........................................................ 464
  Snap Data Sets ............................................................ 464
  Commands and Validation Errors Data Set ................................ 464
  Log Data Sets .................................................................. 465
  Reports Data Set ........................................................... 465
Maintaining the DSXJOBS File ..................................................... 466
Chapter 11. Getting Started

This chapter provides you with the information you need to begin using the NetView DM for MVS Batch Utilities.

What Are the Batch Utilities?

The Batch Utilities are a set of programs provided by NetView DM for MVS and invoked using job control language (JCL). You can use them to perform most of the same functions as GIX, and to maintain the NetView DM for MVS files. These utilities form an interface with NetView DM for MVS, allowing host application programs to use NetView DM for MVS facilities.

The Batch Utilities that you use to carry out the main NetView DM for MVS functions are:

- MAINTDAT
- SUBMIT
- TCFMAINT

There are also some general utilities used to maintain the NetView DM for MVS files. These are described in Chapter 18, “Maintaining NetView DM for MVS Files” on page 429. These are:

- PLCOPY
- DSXBUBK
- DSXBURS

The MAINTDAT, SUBMIT, and TCFMAINT utilities each have a set of control statements, which you use to pass information to the batch utility. Control statements define the operation NetView DM for MVS is to perform, the resource it is to act upon, and the target node for the operation. The main uses of each type of batch utility are described below.

**MAINTDAT**  
Use this utility to work with resources in the resource repository. MAINTDAT control statements are described in:

- Chapter 15, “Managing Resources in the Repository” on page 261.

How to run the MAINTDAT utility is described in “How to Run MAINTDAT” on page 203.

**SUBMIT**  
Use this utility to define and work with:

- User profiles
- Nodes and node groups
- Resources and resource groups
- Resource assignments
- Transmission plans

The SUBMIT control statements are described in:

- Chapter 12, “Defining User Profiles” on page 211
- Chapter 13, “Defining Nodes” on page 215
- Chapter 14, “Defining and Assigning Resources” on page 239
- Chapter 16, “Preparing and Submitting Transmission Plans” on page 313

The instructions for running the SUBMIT utility are described in “How to Run SUBMIT” on page 204.
TCFMAINT

Use this utility to work with plans that have been submitted to the transmission control file (TCF) for processing. The TCFMAINT control statements are described in:

- Chapter 17, “Managing Submitted Plans” on page 395.

The instructions for running the TCFMAINT utility are described in “How to Run TCFMAINT” on page 206.

To run the Batch Utilities on an MVS host, you insert control statements into a job file and then submit the job file to your host operating system.

**How to Define Control Statements**

Each batch utility control statement consists of:

- A control statement label (optional)
- A verb and modifier
- A set of operands
- An END statement (optional)

This book uses boxes like this to illustrate the syntax of control statements:

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
</table>
| VERb MODifier | KEYWORDA=(KEYWORD1|KEYWORD2), {KEYWORDB=value|KEYWORDC=value} [.KEYWORDD=[KEYWORD1|variable 
| | [.KEYWORDE=(variable,...)] | DPPX, SSP only |
| | [.KEYWORDE=(value1,value2,value3,value4)] | SSP, PDOS only |
| | [.KEYWORDF='literal with embedded blanks'] | CMEP only |
| | | Except VSE |

Any additional information about the control statement.

The first column (Statements) contains the verb and its modifier, for example, PREPARE DATASET.

The second column (Operands) shows the operands in the operand field. The order of operands in the field is not significant.

The third column (Node Types) shows which operands are valid for which types of node, and in a few cases, which operands are valid for which types of resource. In this example, you can use KEYWORDB with only the DPPX and SSP node types. When an operand is valid for all node types, this column is left blank. When all operands in a table have the same node restrictions, this is specified once in bold type.

**Control Statement Label**

Although the labels are not marked in the control statement definitions, each statement can optionally have a label. When you use a label, the following rules apply. The label must:

- Begin in the first position of the statement (column one)
- Be followed by one or more blanks
- Contain from 1 to 8 alphanumeric characters, the first character of which must be alphabetic
Verb and Modifier

The beginning of each statement is a verb and modifier. If you do not use a label, you must insert at least one blank before the verb (it can begin in columns 2 through 35 inclusive). The verb and modifier must end, and the first operand must begin, on the same line.

You can enter either the complete verb and modifier, or the first three letters of the verb and the first four letters of the modifier, for example PRE DATA.

Operands

You must enter one or more blanks immediately before and after the operand field. You should separate successive operands by commas, with no space before or after.

Here are the syntax conventions that are used to describe the operands. Enter uppercase, non-italicized letters and words, parentheses ",," commas, and the equal sign "=" exactly as specified in the definition.

KEYWords

Words that appear in upright (non-italicized) text in tables are keywords. When these keywords are referred to in text, they appear in bold uppercase typecase, as in RESTYPE.

variables

Words that are italicized are words that require an appropriate value. For example, replace file-control-name with a specific file control name, for example, FCTLNAME=QUEUE014. Replace nn with a two digit decimal number, for example, LINES=48.

The syntax you use to name resources varies according to the node type that you are working with. Operands such as origin or file control name have a fixed syntax of from 1 to 8 alphanumeric characters, the first character of which must be alphabetic.

(a,b,c,d)

Multiple variables, in parentheses and separated by commas, are a positional operand set. This means that when you omit an entry and code a later one, you must enter a positional comma for the omitted entry. You can omit entries from the end of the list. If you enter only the first operand, you can omit the parentheses, but if there are nested parentheses, you must include all of the parentheses.

Note: The following symbols are used in the boxes illustrating control statements throughout this book simply to explain the format of the operand field. Never include them as part of the control statement.

[optimal]

Square brackets show that an operand, or set of operands, is optional. The text that describes each operand states what happens if you omit the optional operand. Usually, NetView DM for MVS assumes a default. The default value is shown underlined in the definitions.

{either|or}

Curly braces show that you must specify only one operand out of a set of operands separated by | (the “or” sign).

(name,...)

Ellipsis points "..." show that you can repeat the variable operand in this position. If you enter a list of similar operands, you must:

- Separate the values with commas
- Enclose the list in parentheses

The letters A through Z and the special characters %, &, $, #, @, and _ (underscore) are considered alphabetic.

Note: Avoid using _ (underscore) in the names of nodes, plans, and groups of nodes or resources. If your definition is used with the generalized interactive executive (GIX), GIX may interpret the underscore character as a blank.
How to Define Control Statements

If an operand is followed by a blank, not preceded by a comma, the control statement is ended. Embedded blanks are only allowed if the operand is a literal enclosed in single quotation marks ‘“’. If the literal character string includes a single quotation mark or apostrophe, you must substitute a pair of single quotation marks for each such character. Restrictions on the length of literals do not include leading, trailing, and embedded single quotation marks.

END Statement

You should code an optional END statement to indicate the end of the control statement set being processed. If you do not include an END statement, NetView DM for MVS will provide one and issue a warning message. Refer to “END” on page 210.

General Rules for Using JCL with Control Statements for Batch Utilities

This book assumes that you are familiar with using JCL. Here are some guidelines on using JCL with NetView DM for MVS:

- Lines cannot go beyond column 71. In order to continue a line, enter any nonblank character in column 72. This book uses “X” as the continuation character. If the current line is a comment, its continuation line can begin in any column. If the current line is a control statement, its continuation line must begin in column 16. There is no limit on the number of continuation lines, or on the total number of characters in the operand field.

- You can write comments in a control statement, but they must be separated from the last operand of the operand field by one or more blanks. A control statement with comments but no operands must have a comma followed by at least one blank separating the comment from the verb. A comment beginning on a new line begins with an asterisk in column one.

What Could Go Wrong

If the verb and modifier do not exactly match an expected combination, NetView DM for MVS ignores the control statement, issues an error message and processes the next statement.

The input/output preprocessor (called DSXPREP) does, however, try to recover from syntax errors that occur in the operands of control statements. To do this, DSXPREP makes assumptions about what is expected when it encounters an incorrect item, that is, on what should normally follow the preceding item. There is no guarantee that these assumptions made in the recovery are correct.

In general, if any operand is coded twice by mistake, NetView DM for MVS uses the first occurrence and does not issue an error message. Any exceptions to this rule are mentioned in the individual control statement definitions.

During error recovery, DSXPREP issues diagnostic messages. Refer to the Messages and Codes book, which explains the notation of diagnostic messages. Table 16 on page 201 summarizes the possible return codes.
How to Define Control Statements

Table 16. Batch Utility return codes

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC 0</td>
<td>Normal completion</td>
</tr>
<tr>
<td>RC 4</td>
<td>Normal completion, but results may not be as expected (for example, a default was assumed, or an embedded control statement expected, or a replace specified when the resource does not exist to replace).</td>
</tr>
<tr>
<td>RC 8</td>
<td>Error. The issued request is rejected (for example, due to a syntax error, or an authorization class with the wrong operand value). Processing continues with the next statement.</td>
</tr>
<tr>
<td>RC 12</td>
<td>Error. The issued request is rejected, and processing terminates (for example, due to an error on a NetView DM for MVS file, or a file being full).</td>
</tr>
<tr>
<td>RC 16 - 20</td>
<td>Abnormal end (for example, due to an unrecoverable error).</td>
</tr>
</tbody>
</table>

Batch Utilities will return the highest return code from processing. You can use this to decide whether or not you want to start another job.
How to Run the Batch Utilities

To run the Batch Utilities on an MVS host, you insert batch control statements in a job file, and submit the job file to your host operating system.

The name of the NetView DM for MVS batch preprocessor is DSXPREP. When you code a batch utility, include the following statements as the first statements in the job file:

```plaintext
EXEC PGM=DSXPREP
   PARM='FUNCTION=utility,LINECNT=nn,INST=n,
       USERID=userid,PASSWORD=yy,PROFILE=zz'
```

Figure 89. Example of DSXPREP operands

**PGM**

The DSXPREP program is required. DSXPREP checks the **PARM** operand string to make sure that the user is authorized to run the batch utility specified by the **FUNCTION** parameter. It also checks the syntax of the control statements in the job file. Afterwards, DSXPREP passes control to the batch utility specified in the **FUNCTION** operand.

**FUNCTION**

Specify the name of the batch utility that you want to run; for example, MAINTDAT, SUBMIT, TCFMAINT, or PLCOPY.

**Note:** FORMAT and LIBMAINT are also accepted, for compatibility with Release 2 of NetView DM for MVS.

**LINECNT**

This operand is optional. Specify the number of lines per page in the printed output. If you do not specify a value, the default is taken from your user profile. If your user profile does not specify a default, the default value is 55. You can specify any number from 21 to 99.

**INST**

Use this operand only if the SPMF feature is installed. Specify the instance number to refer to the correct set of DB2 tables. Because the DRD and TCF VSAM clusters are no longer referenced by NetView DM for MVS programs, the following JCL DD statements are no longer required and must be removed:

```plaintext
//DSXDRD //DSXTCF
```

**USERID**

You must specify a **userid** for which a NetView DM for MVS user profile has been defined.

**PASSWORD**

The password is optional, depending on how your user profile and the NDMBATCH customization macro were defined.

If your user profile was defined with a password, and the PSWD operand of the NDMBATCH customization macro has been set to **REQ** (required), you must specify this password here in one of the following ways:

- By inserting the **PASSWORD** operand in the **PARM** card of the JCL
- By inserting the first eight characters of the first record of a file with DD NAME=BATCHPW in the JCL.

If your user profile was defined with a password but the PSWD operand of the NDMBATCH customization macro has been set to **OPT** (optional), then you can omit the password here.

If your user profile was defined without a password but you want the user exit routine to check your RACF password, then you must specify the password here (in the batch
utility invocation). The password is passed through the Interface Control Block (ICB) to the user exit.

**PROFILE**

This specifies the suffix number of the profile that NetView DM for MVS will use to run the batch utility. This is a 2-digit number generated at installation time as SUFFIX=nn, using the generation macro called NDMBATCH. If you do not specify a profile number, the default value is 01.

After specifying these initial statements, you must put DD statements in the job files to define the various data sets that the Batch Utilities access.

### How to Run MAINTDAT

You can run the MAINTDAT batch utility by submitting JCL statements. The JCL DD statements that you can include in the JCL are shown in Table 17. Figure 90 on page 204 is an example of JCL you use to run the MAINTDAT utility in a VSAM environment.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLIB</td>
<td>Partitioned data set for COPY members</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>MAINTDAT message file</td>
</tr>
<tr>
<td>SNAP</td>
<td>SNAP file</td>
</tr>
<tr>
<td>DSXDRD</td>
<td>Distributed resource directory</td>
</tr>
<tr>
<td>DSXLIB</td>
<td>NetView DM for MVS library directory</td>
</tr>
<tr>
<td>DSXLIBT</td>
<td>NetView DM for MVS library data repository</td>
</tr>
<tr>
<td>BQBLIBI (DPCX only)</td>
<td>Contains DPCX resources to be loaded</td>
</tr>
<tr>
<td>DSXHFDI</td>
<td>Holding file directory</td>
</tr>
<tr>
<td>DSXHFDA</td>
<td>Holding file data repository</td>
</tr>
<tr>
<td>DSXPRINT</td>
<td>MAINTDAT output file</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Control statements file</td>
</tr>
<tr>
<td>BATCHPW</td>
<td>Contains user password if not specified in PARM stream</td>
</tr>
</tbody>
</table>
How to Run the Batch Utilities

Figure 90. Sample JCL to run MAINTDAT in the VSAM environment

To take advantage of the enhancements for the code page translation, add the language environment run library data set:

CEE.V1R4M0.SCEERUN

to the JOBLIB DD card of your batch job.

For compatibility with Release 2, you can still use the FORMAT utility to call up the following MAINTDAT functions:

- CONVERT DATASET/DUMP
- FORMAT FILE/PRINT
- PREPARE CLIST/JOB/DATASET/PRINT

You can also use the LIBMAINT utility to call up these MAINTDAT functions:

INSERT/REPLACE CATEGORY/CATUPDATE/CLIST/DSCB/FP/FSP/JOB/KEY

How to Run SUBMIT

You can run the SUBMIT batch utility by submitting JCL statements. The JCL DD statements that you can include in the JCL are shown in Table 18 on page 205.

Note: In the DD statement SYSPRINT, the LRECL value has to be set to 133 (hard-coded value). Figure 91 on page 206 is an example of JCL to run the SUBMIT utility.
### Table 18. DD statements to run SUBMIT

<table>
<thead>
<tr>
<th>DD name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSLIB</td>
<td>Partitioned data set for COPY members</td>
</tr>
<tr>
<td>DSXPRINT</td>
<td>Utility output file</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>SUBMIT message file</td>
</tr>
<tr>
<td>SNAP</td>
<td>SNAP file</td>
</tr>
<tr>
<td>DSXDRD</td>
<td>Distributed resource directory</td>
</tr>
<tr>
<td>DSXGIX</td>
<td>Plan library directory</td>
</tr>
<tr>
<td>DSXGIXD</td>
<td>Plan library data repository</td>
</tr>
<tr>
<td>NDMRQFDA</td>
<td>Request queue file data</td>
</tr>
<tr>
<td>NDMRQF</td>
<td>Request queue file directory</td>
</tr>
<tr>
<td>DSXTCF</td>
<td>Transmission control file</td>
</tr>
<tr>
<td>DSXLIB</td>
<td>DSX library directory</td>
</tr>
<tr>
<td>DSXLIBT</td>
<td>DSX library data repository</td>
</tr>
<tr>
<td>DSXHFDI</td>
<td>Holding file directory</td>
</tr>
<tr>
<td>DSXHFDA</td>
<td>Holding file data repository</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Control statements file</td>
</tr>
<tr>
<td>BATCHPW</td>
<td>Contains user password if not specified in PARM stream</td>
</tr>
</tbody>
</table>
How to Run the Batch Utilities

//SUBPLA JOB (,CIARALL,A), 'CIARALL',
// MSGLEVEL=(1,1),
// MSGCLASS=A,
// REGION=4096K,
// CLASS=M,
// USER=CIARALL, PASSWORD=XXXXXXX
//JOBLIB DD DSN=NDM15.NDMLOAD, DISP=SHR
//JOBCL DD DSN=CATALOG.MVSICF1.VNDM003, DISP=SHR
//STEP EXEC PGM=DSXPREP,
// PARM='FUNCTION=SUBMIT, USERID=CIARALL, PASSWORD=KATIA'
//SNAP DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//DSXPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DSXHFDI DD DSN=NDM15.NDMHFDI, DISP=SHR
//DSXHFDA DD DSN=NDM15.NDMHFDA, DISP=SHR
//DSXLIB DD DSN=NDM15.NDMLIB, DISP=SHR
//DSXLIBT DD DSN=NDM15.NDMLIBT, DISP=SHR
//DSXDRD DD DSN=NDM15.NDMDRD, DISP=SHR
//DSXTCF DD DSN=NDM15.NDMTCF, DISP=SHR
//NDMRQF DD DSN=NDM15.NDMRQF, DISP=SHR
//NDMRQFDA DD DSN=NDM15.NDMRQFDA, DISP=SHR
//DSXGIX DD DSN=NDM15.NDMGIX, DISP=SHR
//DSXGIXD DD DSN=NDM15.NDMGIXD, DISP=SHR
//SYSIN DD *
  DEF PLAN NAME=TTTPLAN7
  DEF PHAS NAME=TTTPHASE, NODE=T12PS113,
  STATUS=HOLD
  SEND RESOURCE NAME=FLAT.13.*,.*.UPD.01.02,
  NETLU=3, X
  NETID=4, X
  RESTYPE=0100
END

Figure 91. Sample JCL to run SUBMIT in the VSAM environment

Note:

To run the SUBMIT Batch Utility in a DB2 environment (with SPMF feature installed only), modify the JCL as follows:

1. Add the Inst=n parameter in the PARM statement to specify the SPMF instance number to point to the correct set of DB2 tables.
2. Remove the JCL DD statement //DSXDRD because the distributed resource directory (DRD) is no longer used.
3. Remove the JCL DD statement //DSXTCF because the transmission control file (TCF) is no longer used.

How to Run TCFMAINT

You can run the TCFMAINT batch utility by submitting JCL statements. The JCL DD statements that you can include in the JCL are shown in Table 19 on page 207.

Note: In the DD statement SYSPRINT, the LRECL value has to be set to 133 (hard-coded value). Figure 92 on page 208 is an example of JCL to run the TCFMAINT utility.
### Table 19. DD statements to run TCFMAINT

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPRINT</td>
<td>TCFMAINT message file</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Print file for dumps</td>
</tr>
<tr>
<td>DSXPRINT</td>
<td>TCFMAINT utility print file</td>
</tr>
<tr>
<td>SNAP</td>
<td>SNAP file</td>
</tr>
<tr>
<td>SYSLIB</td>
<td>Partitioned data set for COPY members</td>
</tr>
<tr>
<td>DSXDRD</td>
<td>Distributed resource directory</td>
</tr>
<tr>
<td>DSXTCF</td>
<td>Transmission control file</td>
</tr>
<tr>
<td>DSXSTAT</td>
<td>The output sequential file where plan information is stored</td>
</tr>
<tr>
<td>SYSIN</td>
<td>TCFMAINT control statements file</td>
</tr>
<tr>
<td>NDMRQFDA</td>
<td>Request queue file data</td>
</tr>
<tr>
<td>NDMRQF</td>
<td>Request queue file directory</td>
</tr>
<tr>
<td>BATCHPW</td>
<td>Contains user password if not specified in PARM stream</td>
</tr>
</tbody>
</table>
How to Run the Batch Utilities

```
//PRIPLA JOB ('CIARALL','CIARALL',
//  MSGLEVEL=(1,1),
//  MSGCLASS=A,
//  REGION=2404K,
//  CLASS=A,
//  USER=FABB,PASSWORD=XXXXXXXX
//JOBLIB DD DSN=NDM15.NDMLOAD,DISP=SHR
//STEP EXEC PGM=DSXPREP,
//  PARM='FUNCTION=TCFMAINT,USERID=FABB,PASSWORD=MARINA'
//SNAP DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//DSXPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DSXHFDI DD DSN=NDM15.NDMHFDI,DISP=SHR
//DSXHFDA DD DSN=NDM15.NDMHFDA,DISP=SHR
//DSXLIB DD DSN=NDM15.NDMLIB,DISP=SHR
//DSXLIBT DD DSN=NDM15.NDMLIBT,DISP=SHR
//DSXDRD DD DSN=NDM15.NDMDRD,DISP=SHR
//DSXTCF DD DSN=NDM15.NDMTCF,DISP=SHR
//NDMRQF DD DSN=NDM15.NDMRQF,DISP=SHR
//NDMRQFDA DD DSN=NDM15.NDMRQFDA,DISP=SHR
//DSXGIX DD DSN=NDM15.NDMGIX,DISP=SHR
//DSXGIXD DD DSN=NDM15.NDMGIXD,DISP=SHR
//SYSIN DD /
PRI PLAN
END
```

Figure 92. Sample JCL to run TCFMAINT in the VSAM environment

Note:

To run the TCFMAINT Batch Utility in a DB2 environment (with the SPMF feature installed only), modify the JCL as follows:

1. Add the Inst=n parameter in the PARM statement to specify the SPMF instance number to point to the correct set of DB2 tables.
2. Remove the JCL DD statement //DSXDRD because the distributed resource directory (DRD) is no longer used.
3. Remove the JCL DD statement //DSXTCF because the transmission control file (TCF) is no longer used.

How to Run PLCOPY

Refer to “PLCOPY” on page 447 for the method of running the PLCOPY batch utility.

How to Run DSXBUBK and DSXBURS

Refer to “DSXBUBK and DSXBURS” on page 442 for the methods of running the DSXBUBK and DSXBURS Batch Utilities.
General Purpose Control Statements

You can use the following control statements with any of the NetView DM for MVS Batch Utilities, unless otherwise stated. You insert them into the job stream of control statements.

- COPY
- DUMP
- EJECT
- SPACE
- TITLE

COPY

Use the COPY statement to copy the member of a partitioned data set that you specify in a SYSLIB DD job control statement. You must insert it at the point in the input stream where the COPY statement appears. The member of the partitioned data set can contain any of the NetView DM for MVS batch utility control statements. COPY statements can be mixed with normal control statements.

membername

The name of the partitioned data set member that you want to copy into the input stream. The name can consist of from 1 to 8 alphanumeric characters, the first of which must be alphabetic.

You can also nest COPY statements. The nesting level available for COPY is limited only by the amount of storage available to NetView DM for MVS. The level of the COPY statement is indicated to the right of each printed COPY record.

For example, a MAINTDAT job stream using COPY statements can look like this:

```cobol
//.... EXEC PGM=DSXPREP,PARM='FUNCTION=MAINTDAT,USERID=XXXX,PASSWORD=YYYY'
//SYSLIB DD DISP=SHR,DSN=TOM.MAINTDAT.SYSLIB01
// DD DISP=SHR,DSN=GENERAL.NDM.SYSLIB
//SYSSIN DD *
COPY ALPHA
   INSERT CLIST NAME=CLISXX, X <== col 71
   CMD='.............'
```

Figure 93. Sample JCL using COPY statements

DUMP

You can only use the DUMP control statement with the FORMAT batch utility. It produces an abend dump, which you can use to help correct severe errors (RC=16) in a job stream. You can place DUMP ON at any point in the SYSIN stream, to apply to all of the control statements that follow, until a DUMP OFF statement is encountered.

EJECT

Use the EJECT statement to eject a page in an output listing. You can use this statement as many times as required in the input stream.
END

Use the END statement to indicate the end of the input stream. Any statements that appear after the END statement are not processed. If you do not include an END statement, NetView DM for MVS provides one and gives you a warning message.

SPACE

Use the SPACE statement to specify the number of lines to skip when output is printed. You can use this statement as many times as required in the input stream.

[number]

The number of lines to skip. You can specify a number from 1 to 55. The default is 1.

TITLE

Use the TITLE statement to specify a title for each page of the input log. You can use this statement as many times as required in the input stream.
Chapter 12. Defining User Profiles

This chapter explains how to define, change, delete and print user profiles using the SUBMIT batch utility. You use the following SUBMIT control statements to work with user profiles:

- DEFINE USER, to create a new user profile
- CHANGE USER, to change an existing user profile
- DELETE USER, to delete a user profile
- PRINT USER, to print a user profile

This chapter describes these control statements in alphabetical order. But first, read the following information to find out about user profiles.

What Is a User Profile?

A user profile is a file which contains a description of each user of NetView DM for MVS. Each user must be assigned a user profile before that user can access either GIX, Batch Utilities, or the IOF. The profile describes:

- Who the user is, in terms of a user identification and password. The user identification corresponds to the TSO user ID; the password is specific to the NetView DM for MVS environment.
- Which GIX tasks the user is authorized to perform. The different functions are listed on page 212, in the FUNCAUTH parameter.
- Which nodes, resources, and functions the user is authorized to access.
- The default values that will appear on GIX panels, customized for the user.

If you are the system administrator using NetView DM for MVS for the first time after installation, you must create a user profile for each user before that user can access NetView DM for MVS. Your users will probably need different authorizations, depending on their role in your organization and the tasks they will have to perform. The user profiles are stored in the distributed resource directory (DRD).

Note: If you are a new user, but not the first person to use NetView DM for MVS, contact your system administrator, or whoever is authorized to define new user profiles.

SUBMIT Control Statements for User Profiles

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.
CHANGE USER/DEFINE USER

Use the DEFINE USER control statement to define the authorizations and password of a new user, creating a user profile. You can then change these authorizations or the password at any time using the CHANGE USER control statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAnge USER</td>
<td>NAME=user-name [.FUNCAUTH=function-authorized][ALL] <a href="node-class,...">.NODECLASS=A0</a>[ALL] [.NODETYPE=(node-type,...))[ALL] [.PASSWORD=password] <a href="resource-class,...">.RESCLASS=A0</a>[ALL] [.RESTYPE=(resource-type,...))ALL]</td>
<td>All node types</td>
</tr>
<tr>
<td>DEFine USER</td>
<td>NAME=user-name [.FUNCAUTH=function-authorized][ALL] <a href="node-class,...">.NODECLASS=A0</a>[ALL] [.NODETYPE=(node-type,...))[ALL] [.PASSWORD=password] <a href="resource-class,...">.RESCLASS=A0</a>[ALL] [.RESTYPE=(resource-type,...))ALL]</td>
<td></td>
</tr>
</tbody>
</table>

NAME

Specify the name of the user. This is the user identification that must be entered by GIX users when logging on to GIX, batch utility users when creating job files, and IOF users when logging on to IOF. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

FUNCAUTH

Specify an optional string of ten characters. Each character has a positional significance, as follows:

- Y in a specified position grants authority to the user to carry out specific tasks
- Any other character prohibits the user from carrying out the tasks.

The user tasks are as follows:

Char 1  Manage user profiles  
Char 2  Prepare plans and delete plans from the plan library  
Char 3  Manage submitted plans in the TCF  
Char 4  Manage network configurations  
Char 5  Manage resources in the resource repository  
Char 6  Maintain the repository and plan library utilities  
Char 7  Use the interactive operator facility (IOF)  
Char 8  Delete phases, D&CC API requests, and node-solicited requests using the IOF  
Char 9  Change node profile authorizations  
Char 10 Manage other users’ plans.

You can only specify Y for the Manage other user’ plans option (char 10) if you also specify Y for the Prepare plans and delete plans from the plan library option.

The default is ALL of the tasks listed above.

NODECLASS

Optional. You can specify the list of node classes that the user is authorized to access. The node class is a way of authorizing only certain users to work with specific nodes. You assign a node class to a node when you define the node. See “CHANGE NODE/DEFINE NODE” on page 217.

A node class consists of two alphanumeric characters. The first character must be alphabetic. You can specify up to 14 node classes. The default is A0. All users can use class A0 nodes.
**NODETYPE**  
Optional. You can specify the node types that the user is authorized to access. Refer to Table 70 on page 561 and Table 71 on page 562 for a list of node types. If you do not specify a value, the default is ALL node types.

**PASSWORD**  
Optional. You can specify a password that you, the system administrator, allocate to each NetView DM for MVS user.

This password must be entered by GIX and IOF users when they log on, batch users when they submit jobs, and D&CC API users. Even if this password is specified, you do not have to code a password in the job file to be submitted if the operand PSWD of the NDMBATCH installation macro has been set to OPT (optional). You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

**RESCLASS**  
Specify the list of resource classes that the user is authorized to access. The resource class is a way of authorizing only certain users to work with specific resources.

A resource class consists of two alphanumeric characters. The first character must be alphabetic. You can specify up to 14 classes. If you do not specify a value, the default is A0. All users can use class A0 resources.

**RESTYPE**  
Specify the resource type names or resource type codes that the user is authorized to access. If you do not specify a value, the default is ALL. Refer to “NetView DM for MVS Resource Types” on page 566 for a complete list of the resource type names and codes that you can use.

**Example:**  
Here is an example of a DEFINE USER statement to create a new user profile:
```
DEF USER NAME=OCTOPUS,NODETYPE=DPPX,NODECLASS=A/zerodot,PASSWORD=XXXX
```

This is a CHANGE USER statement to change the definition of an existing user:
```
CHA USER NAME=GIRAFFE,NODETYPE=VM,PASSWORD=MMMM
```
Use the DELETE USER control statement to delete an existing user profile from the DRD. Use the PRINT USER statement to print a user profile.

**Attention**

If you are the system administrator, do not delete your own user profile if it is the only profile remaining in the DRD. There must always be at least one user profile defined.

### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete USER</td>
<td>NAME=user-name</td>
<td>All node types</td>
</tr>
<tr>
<td>PRInt USER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NAME**

Specify the name of the user. This is the user identification that must be entered by GIX users when logging on to GIX, batch utility users when creating job files, and IOF users when logging on to IOF. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

**Example:** Here is a DELETE USER statement to delete the definition of an existing user from the DRD:

```
DEL USER NAME=ZEBRA
```

Here is a PRINT USER statement to print the definition of an existing user:

```
PRI USER NAME=SEALION
```
Chapter 13. Defining Nodes

This chapter explains how to define, change, delete, and print node definitions using the Batch Utilities. It also shows you how to organize nodes and resources into groups that are easier to work with. Finally, the chapter shows you how to work with logical units.

You use the following SUBMIT control statements to work with node definitions:

- DEFINE NODE, to create a new node definition
- CHANGE NODE, to change an existing node definition
- DELETE NODE, to delete a node definition
- PRINT NODE, to print a node definition

This chapter describes these control statements in alphabetical order. But first, read the following pages to find out about node definitions and node types. If you want to work with node or resource groups, turn to “Organizing Nodes and Resources into Groups” on page 225. If you want to work with the assignment of logical units to transmission profiles or connection profiles, turn to “SUBMIT Control Statements for Logical Units” on page 233.

What Is a Node Definition?

A node definition describes the characteristics of a node to NetView DM for MVS. You must define each node in your network to NetView DM for MVS before end users can refer to the nodes in transmission plans.

For each node, you must define:

- A symbolic name that identifies the node
- The type of node
- A set of node attributes

The following pages explain the node types you can define.

Node Types

NetView DM for MVS can use a variety of different types of node. The node type identifies which hardware and software the node comprises. The types of node that your host is able to access are defined at installation time, as described in the Installation and Customization book, which also describes the hardware, operating system, and other software required at the nodes.

Nine types of nodes have names predefined in NetView DM for MVS. These nodes use the LU0 protocol for communication connections. The node type determines which resources and functions you can use.

There are also node types which do not have predefined names. Each of these node types has a user-defined name. These nodes use LU 6.2 protocols for communications connections. Again, the node type determines which resources and functions you can use.

Nodes can be connected to the host either directly or through an intermediate node. The intermediate node queues resources and distribution requests from the host and sends them to the nodes. It also queues resources and requests received from the nodes and forwards to the host. The end point nodes are called entry points. The host, where NetView DM for MVS usually runs, is called the focal point.
What Is a Node Definition? (System Administrator)

LU0 Node Types: Table 70 on page 561 contains a complete list of the LU0 node types you can define.

When you define one of these node types, you are also defining its capabilities in relation to the functions and resources it can use. The node types differ in the functions and resources they can handle. Appendix A contains a set of tables which map out the relationship between node types, functions and resource types.

LU 6.2 Node Types (User-defined): Table 71 on page 562 contains a complete list of the user-defined LU 6.2 node types you can define.

The functional capabilities of these nodes are defined at installation time in the following ways:

CMEP Change management entry point

Nodes which receive and implement changes that are managed and distributed by the focal point are called entry points. An entry point is any distributed processor or subsystem whose software, microcode, user data, updates for each of them, and maintenance information are controlled and tracked by the focal point. An entry point can also initiate distribution, retrieval, and deletion requests to other entry points and to the focal point, but only a focal point can initiate remote change management functions. An entry point is also referred to simply as a “node” throughout this book.

Each CMEP node type has a node authorization profile. This profile defines the functions that the CMEP node can request in a node solicited request (NSR), the resource types that the node can work with, and the classes of resource. NetView DM for MVS provides defaults for these three parameters, but they can also be changed at installation time in the NDMNODE generation macro. You can also change these parameters using GIX, when you define each node to NetView DM for MVS. Refer to “CHANGE NODE/DEFINE NODE” on page 217.

The control statement definitions in “SUBMIT Control Statements for Nodes” on page 217 indicate which node types are valid for each function.

Note: CMEP node types support only LU 6.2 resource types, and not LU0 resource types.

CMFP Change management focal point

You can define a node as a CMFP node, which means that it has the functional capability to administer changes to other nodes. A focal point is any host which implements centrally controlled resource distribution, remote change management, and tracking. For example, you could define any S/390 host where NetView DM for MVS is running and controlling the retrieval, distribution, and remote installation of resources to and from distributed processors or subsystems. You can define several nodes in the same NetView DM for MVS network as CMFP nodes.

The control statement definitions in “SUBMIT Control Statements for Nodes” on page 217 indicate which node types are valid for each function.

Note: CMFP node types support only LU 6.2 resource types, and not LU0 resource types.

NDMT NetView DM for MVS transfer

You can define another NetView DM for MVS node in the network to your DRD (at the host) as an NDMT node. NDMT is the functional capability of one NetView DM for MVS focal point to exchange data objects from its resource repository with another NetView DM for MVS focal point resource repository.
NDMT nodes support only LU0 resource types.

You can define an LU 6.2 node's capabilities in any one of these ways:

- CMFP capabilities only
- CMEP capabilities only
- NDMT capabilities only
- Both NDMT and CMFP capabilities.

SUBMIT Control Statements for Nodes

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

CHANGE NODE/DEFINE NODE

Use the DEFINE NODE control statement to define a node to the DRD. You can use the CHANGE NODE statement to change an existing node definition. In order for this change to have an immediate effect when the TCP is running with the NETCHNG startup parameter set to IMMEDIATE, you must also run Batch Utilities with NETCHNG=IMMEDIATE in the batch profile. The specified node type must already have been defined when the TCP was started.

If you are not running with NETCHNG=IMMEDIATE, the change will only come into effect when the TCP is cold-started.

However, if you change either the associated logical unit, the connection type, or the RGN or REN definitions, the change always comes into effect immediately and the TCP deletes all existing units of work (phases for node, API requests and NSRs) for that node.
## CHANGE NODE/DEFINE NODE

### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE NODE</td>
<td>NAME=node-name .NODETYPE=nodetype .LUNAME=logical-unit-name .CONNECTION=DIRECT .DIRECTORY=directory-name .LOGMOD=logon mode .RGN=netid .REN=netlu .CLASS=node-authorization-class .LINETYPE=L</td>
<td>See note</td>
</tr>
<tr>
<td>DEFINE NODE</td>
<td>.LUNAME=logical-unit-name .CONNECTION=DIRECT .DIRECTORY=directory-name .LOGMOD=logon mode .RGN=netid .REN=netlu .CLASS=node-authorization-class .LINETYPE=L</td>
<td>SSP, PDOS only PDOS only Except VSE Except user-defined DPPX, SSP only CMEP only CMEP only CMEP only CMEP or CMFP</td>
</tr>
</tbody>
</table>

You cannot use the NODETYPE operand with CHANGE NODE. You can change any attributes of an existing node definition, except for the node name and type.

### NAME

Specify the name of the node. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. Do not use _ (underscore) in a node name, because GIX may interpret this character as a blank, with unreliable results.

For PDOS and SSP entry points connected via intermediate node support, the name must match the system name assigned to this node in the Communication Routes definition for the front-end SSP.

**Note:** When the node is defined as having CMFP, CMEP, or NDMT functional capabilities, you can only define one end node for each target destination distribution service unit (DSU). More than one of these types can share the same LUNAME.

### NODETYPE

Specify the type of node. Refer to Table 70 on page 561 and Table 71 on page 562 for a list of node types.

### LUNAME

Specify the mandatory logical unit (LU) name assigned to the node. You must specify the LU name that was assigned to the node when NetView DM for MVS was installed, that is, the name defined in the VTAM/TCAM tables for the same node. If you are distributing resources via intermediate nodes, the logical unit name must be the name of the next (intermediate) node. Nodes of the same type, which have the same type of connection, can share the same LU name. Different node types must share the same LU name in case of PDOS and SSP end nodes connected via intermediate node support through the same front-end node. You must specify the LU name that was defined to VTAM/TCAM for the front-end node.

If the node is a user-defined type node, with NDMT, CMFP, CMEP, or both NDMT and CMFP functional capabilities, the LUNAME value is the logical unit name of the directly connected node.
If the node you are defining or changing is associated with a logical unit that NetView DM for MVS does not know, the logical unit is assigned to one of the default transmission profiles as follows:

- To NDMTPL, if you specify LINETYPE=L
- To NDMTPL, if you do not specify a LINETYPE
- To NDMTPS, if you specify LINETYPE=S.

If the node you are defining is associated with a logical unit that NetView DM for MVS already knows, the LINETYPE information is checked for consistency with the type of transmission profile that is already associated with the logical unit. If the information is inconsistent, NetView DM for MVS stops processing the statement and issues an error message.

If you did not specify a LINETYPE value, the definition of the new node is accepted.

Refer to the Installation and Customization book for information about the default transmission profiles.

**CONNECTION**
Specify whether an SSP or PDOS entry point is connected to the host via an intermediate SSP or AS/400 node (CONNECTION=INTERMEDIATE), or not (CONNECTION=DIRECT).

**DIRECTORY**
You must always specify a directory for PDOS and SSP nodes with CONNECTION=INTERMEDIATE. It must match the name in the Communication Routes definition to the SSP node that connects this node to the host. It can be from 1 to 8 alphanumeric characters. The first character must be alphabetic.

*Note:* Do not specify a directory for SSP nodes.

**LOGMOD**
Specify the logon mode assigned to the node.

If both Batch Utilities and the TCP are running with NETCHNG=IMMEDIATE and you change the logon mode value of an LU 6.2 node that is directly connected to the host, NetView DM for MVS automatically changes the logon mode to this value for all of the nodes that share this logical unit.

Changing the logon mode value for an LU 6.2 end node, however, does not affect the logon mode values of other nodes that share the same logical unit. Only changes to the logon mode of a node that is directly connected to the host, and that shares the same logical unit as the end node, affects transmission.

**RGN**
**REN**
These two operands apply to CMEP and CMFP user-defined node types only.

RGN is a SNA/DS term that means routing group name. REN is a SNA/DS term that means routing element name. Together these two operands specify the distribution services unit (DSU) identifier of the target node. You must specify the RGN and REN values that are defined at the node.

When defining NetView DM/6000 nodes, you must specify the RGN and REN values as follows:

- For a NetView DM/6000 node that is directly connected to the NetView DM for MVS host, the RGN and REN values are the same. The RGN must be the same as the node's REN (or short name, as it is referred to in NetView DM/6000).
- For NetView DM/6000 nodes that are indirectly connected to the host as CC clients, RGN corresponds to the REN defined for the CC server to which the CC client is connected.
connected locally. REN corresponds to the REN defined for the CC client when it was configured.

- If the DRD contains node definitions for NetView DM/6000 nodes with NetView DM/6000 Release 1.0 installed and the CC server is upgraded to NetView DM/6000 Release 1.1, you must change the definitions for the CC server and all of the CC clients controlled by the server.

You can enter up to eight characters for each operand. Each character must belong to the subset of character set 1134 that is defined in “Character Sets” on page 585.

If you do not specify the target name, the default is the name of the node directly connected to the host (this node is an intermediate node). When using intermediate nodes, the LUNAME (logical unit name) you specify for the node you are defining will be the LUNAME of the directly connected node (an intermediate node). Note how this applies in the example shown in Figure 94 on page 222.

CLASS Specify the node authorization class. You can enter two alphanumeric characters. The first character must be alphabetic. If you specify class A0, all users can access the node. If you do not specify a value, the default is taken from the node class field in the existing user segment.

LINETYPE Specify whether the node is connected to the host via S (switched) or L (leased) line.

PASSWORD Optional. Specify the logon ID and password that authorizes the host to establish operations with DPPX, and SSP nodes. The logon ID and password must match the ones defined at the node. The logon ID can be from 1 to 8 alphanumeric characters. The password can be from 1 to 4 characters for SSP node types, and from 1 to 6 characters for DPPX node types.

STATUS Specify the node operation status. This can be:

- P (production), which means that no testing is carried out at the node
- T (testing), which means that no production work is carried out at the node
- B (both), which means that both production and testing are carried out at the node.

XMFUNC CMEP nodes only. A user-defined node with CMEP functional capabilities must be authorized to perform a specific transmission function. This is defined by you, the system administrator. Specify a 3-character string. Each character has a positional significance in the string, as follows:

1. SEND
2. RETR
3. DELE

Specifying Y in a position authorizes the transmission function. Any other character prohibits the transmission function.

If you do not specify any characters, the default is taken from the NDMNODE generation macro. Refer to the Installation and Customization book for information about this macro.

RESTYPE CMEP nodes only. Specify the LU 6.2 resource type names or resource type codes that the node is authorized to access. These are listed in Table 74 on page 566. This parameter is only valid for user-defined nodes with CMEP functional capabilities. You can only specify LU 6.2 resource types.

If you do not specify any characters, the default is taken from the RESTYPE parameter of the NDMNODE generation macro. Refer to the Installation and Customization book for information about this macro.
RESCLASS  CMEP nodes only. Specify the resource classes which the node is authorized to access. A resource class consists of two characters. You can authorize a node to access up to 14 classes.

If you do not specify this parameter, the default is taken from the CLASS parameter of the NDMNODE generation macro. Refer to the Installation and Customization book for information about this macro.

SRVNAME  CMEP nodes only. Specifies the Distribution and Change Control Server name (from 1 to 8 alphanumeric characters, the first of which is alphabetic) to which the workstation is attached. If no name is specified, SRVNAME will be set to the same value as for the NAME operand.

The server name does not have to have been previously defined in the DRD.

TIMZOFFS  Time Zone Offset. Use this optional attribute to specify the offset between the time zones where the host and the node are located. This will allow polling to begin at the right time for the INSTALL RESOURCE, ACTIVATE NODE and INITIATE PROCEDURE functions, because the execution date and time are expressed in local time.

The value you specify is a whole number, ranging from +12 to −12 (the + sign can be omitted). The default value is 0 (that is, the host and the node are in the same time zone).

Note: The time zone offset function only applies to nodes with CMEP and CMFP functional capabilities.

NOTE  You can enter any type of information about the node you are defining here. You can specify this operand more than once, and each string can contain up to 256 characters. The total number of characters entered must not exceed 639. You must group together all of the NOTE operands, and they must be the last operands in the operand field.

Examples: Here are examples of two DEFINE NODE statements. The first is defining an intermediate node, the second is defining the corresponding end node:

```
DEF NODE NAME=CMEP4,
   NODETYPE=CMEP,
   LUNAME=T04PS261,
   LOGMOD=LU62,
   NOTE='THIS IS THE INTERMEDIATE NODE'
DEF NODE NAME=CMEP2,
   NODETYPE=CMEP,
   LUNAME=T04PS261,
   LOGMOD=LU62,
   RGN=ITIBM/zerodotPC,
   REN=T/zerodot4PS262,
   NOTE='THIS IS THE END NODE'
```
Figure 94. Example of defining a node connected to the host via an intermediate node
DELETE NODE

Use the DELETE NODE statement to delete a node definition, and its associated resource history record, from the DRD.

In order for this change to have an immediate effect when the TCP is running with the NETCHNG startup parameter set to IMMEDIATE, you must also run Batch Utilities with NETCHNG=IMMEDIATE in the batch profile. The TCP will:

- Immediately cancel all existing requests of transmission activities that involve the node being deleted
- Reject any further requests for transmission activities involving this node.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete NODE</td>
<td>NAME=node-name [,FORCE={Y[ES]</td>
<td>NO}]</td>
</tr>
</tbody>
</table>

NAME

Specify the name of the node. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. Do not use _ (underscore) in a node name, because GIX may interpret this character as a blank, producing undesirable side effects.

FORCE

VSE nodes only. If you specify FORCE=YES then the VSE node will be deleted even when there are phases that are in a PENDING state for the node. This prevents the TCP from continually trying to execute phases for a VSE node that you have deleted, and that no longer exists.

Once you have deleted the node, the PENDING phases remain in the TCF until the TCF is reinitialized. Until the TCF is reinitialized, the following takes place:

- Before a TCP cold start, NetView DM for MVS tries to execute these phases at every TCP startup. The phases are displayed as PENDING in GIX or IOF.
- After a TCP cold start is done, the TCP no longer tries to execute the phases, although they are still in the TCF. The phases are not displayed by IOF, but are still displayed by GIX.

The default value is FORCE=NO. If you specify NO, or you do not enter a value, and there are phases in a PENDING state, then NetView DM for MVS will not delete the node. NetView DM for MVS issues an error message (return code 8).

Example:

Here is an example of a DELETE NODE statement to delete a VSE node definition from the DRD:

```
DEL NODE NAME=MYVSE,FORCE=Y
```
PRINT NODE

Use the PRINT NODE control statement to print a single node definition, or all of the node definitions for a specific node type.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
</table>
| PRINT NODE | {NODETYPE=node-type
| NAME= name} | All node types |

NODETYPE Specify the type of node. Refer to Table 70 on page 561 and Table 71 on page 562 for a list of node types.

NAME Specify the name of the node. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. Do not use _ (underscore) in a node name, because GIX may interpret this character as a blank, with unreliable results.

Example: Here is an example of a PRINT NODE statement to print a node definition:

PRI NODE NAME=MY3174
Organizing Nodes and Resources into Groups

You can organize nodes into node groups, and resources into resource groups. The control statements you use to do this are the same for both nodes and resources. When you create a group you assign it a group name. Using groups saves you time when you are defining transmission plans. It can also reduce the volume of data sent between the host and intermediate nodes. You can specify this group name in your transmission plan, instead of entering each node or resource name. This means that you only have to perform the same task once, for different nodes and resources. For example, you can create a node group that logically corresponds to the task you are performing, such as, “send the same report to all sales nodes.”

A node can belong to more than one group. You can include nodes that are not yet defined in the DRD, but you must define them before you submit plans that address the nodes. If you do not define them, you receive a validation error when you submit the plan. If you define an empty group, you will receive a validation error when you try to use the group in a plan.

When you define a group, your user ID becomes the owner of the group definition. Other users can use the definition, but only you can change or delete it. If you are a system administrator with the User profiles authorization, you can also delete any group.

When to Use Node Groups of the Same Type

Note: With these groups you can only put nodes of the same node type into a group together.

You usually define node groups for all the entry points that are connected to an intermediate node in a multi-tiered network. If you want to send the same resource to all the entry points, the best way to do this is to define the entry point as a group, so you only need to transmit the resource once, between the host and the intermediate node. You can also define groups that specify a subset of the entry points connected to the intermediate node. When you specify the subset in a plan, the intermediate node will only send the resource to the entry points in the subset group.

Group of LU 6.2 Nodes with Different Node Types

You can define a group of nodes where the nodes have different node types. This only applies to user-defined node types (LU 6.2 nodes) with either CMEP or CMFP functional capabilities. This means you can omit the node type when defining a group of nodes.

When the node type of a group of nodes is not specified, you can include in this group LU 6.2 nodes that have:

- Different node types
- Functional capabilities of the node type to which they belong, either CMEP or CMFP.

You can define a node group without defining its node type only if you are authorized to use at least one LU 6.2 node type with either CMEP or CMFP functional capability.

When to Use Server Node Groups

When defining a node group, you can also have NetView DM for MVS automatically create another node group that contains the server nodes of those nodes in the initial group. You can define the:

- Group name of the automatically-created server group
- Disposition of the server group: New or Old
Define Group: Use the DEFINE GROUP function to define the server group’s name and disposition. To create a server node group, the initial node group must meet these conditions:

- All the nodes in the initial group must be defined in the DRD.
- Each of the nodes in the initial group must have the same node type as the group’s node type (if defined). For any initial group with an undefined node type, all nodes in the group must be LU 6.2 nodes with either CMEP or CMFP functional capabilities. If the initial group has an undefined node type:
  - For CMEP nodes, NetView DM for MVS adds the CMEP node server to the server group.
  - For CMFP nodes (where no server node exists), in order to address the server group, NetView DM for MVS adds the CMFP node name to the server group.

A server group is always created with an undefined node type (mixed group).

Changing a Server Group: Use the batch DEFINE GROUP statement to create or update a server group starting from an already defined group of nodes. You have these options:

- If the node group is already defined, you can add nodes to this group and create or update a corresponding server group by using the SRVGRP operand of DEFINE GROUP.
  
  The original group of nodes is changed and the server group is created/updated according to the disposition.

- If the node group is already defined and you do not want to add any nodes to it, but you want to create or update a corresponding server group, use the SRVGRP operand of DEFINE GROUP.
  
  The original group of nodes is unchanged. The server group is created or updated.

- If the node group is already defined but empty, and you do not want to add any nodes to the group, but you do want to create or update a server group, use the SRVGRP operand of DEFINE GROUP.
  
  The original group of nodes is unchanged. An empty server group is created.

Delete Group: When deleting a node group or some nodes in a node group with the DELETE GROUP batch statement, you can specify the name of the server group and the disposition with the SRVGRP operand.

When the node group is deleted and a corresponding server group name has been specified, both groups are deleted from the DRD (regardless of the disposition).

When some or all nodes are deleted from the group (but the group itself is not deleted), and a corresponding server group name has been specified with a disposition value, the server group is updated/created (with some servers perhaps deleted from the group) according to the disposition.
When to Use Resource Groups

You can organize resources into groups, in the same way as you organized nodes into groups. When you create a group of resources you identify it by assigning it a *group name*. As with node groups, you can group resources together when you want to perform the same task on all resources. You can specify this group name in your transmission plan, instead of specifying each resource name.

The following resource types can be grouped:

- AS/400 data object containment structure
- AS/400 object
- CLIST
- Configuration file
- Dataset
- Dump (CMEP)
- Error log (CMEP)
- Flat data
- Member
- Microcode and Microcode customization
- Panel
- Print
- Procedure
- Program
- Relational data
- Software
- Trace

The following considerations apply:

- A group of resources defined for an LU0 node type can also be used in operations with user-defined nodes with NDMT functional capabilities.
- A group of resources defined for an LU 6.2 node type can only be used by LU 6.2 nodes, and not by LU0 node types.

SUBMIT Control Statements for Groups

You can use the following SUBMIT control statements to work with groups of nodes, and groups of resources:

- DEFINE GROUP, to define a group of nodes or resources
- DELETE GROUP, to delete the definition of a group of nodes or resources
- PRINT GROUP, to print all node or resource groups of a specific type

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.
DEFINE GROUP/DELETE GROUP

Use the DEFINE GROUP statement to define a group of nodes or resources to the DRD, or to add a new member to an existing group.

Use the DELETE GROUP statement to delete the definition of a group of nodes or resources from the DRD, or to delete a member from an existing group. The DELETE GROUP statement has the same syntax as DEFINE GROUP, except that you are also allowed to specify NODE=ALL (or, for example, FP=ALL) to delete an entire group.

To Work with Groups of Nodes: You can work with a group defined for LU0 node types and with user-defined nodes with NetView DM for MVS Transfer (NDMT) capabilities without redefining the group. Specify the name of the group and the type of nodes in the group, followed by the names of the nodes.

To Work with Groups of Resources: Specify the name of the group, and the type of nodes to which the resources belong, followed by the resource type, the resource names and their version (if applicable), modification levels and extents.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node/Resource Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE GROUP</td>
<td>NAME=group-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NODETYPE=node-type]</td>
<td></td>
</tr>
<tr>
<td>DELETE GROUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NODE=(node-name1,node-name2,...,node-name2000)]</td>
<td>User-defined types only</td>
</tr>
<tr>
<td></td>
<td>[.RESOURCE=(name1,name2,...,name50)]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.RESTYPE=[resource-typeALL]]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.CATE=((name[, ver, mod]),...)]</td>
<td>Except VSE, CMEP, CMFP</td>
</tr>
<tr>
<td></td>
<td>[.CATU=((name[, ver, mod]),...)]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.CLIST=((name[, ver, mod]),...)]</td>
<td>Except CMEP, CMFP</td>
</tr>
<tr>
<td></td>
<td>[.DSCB=((name[, ver, mod,ext]),...)]</td>
<td>Except SSP, PDOS, CMEP, CMFP</td>
</tr>
<tr>
<td></td>
<td>[.DSN=(name,...)]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.DUMP=(name,...)]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.FP=((name[, ver, mod]),...)]</td>
<td>NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.FSP=((name[, ver, mod]),...)]</td>
<td>VSE, NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.MEMBER=((name[, ver, mod]),...)]</td>
<td>Except CMEP, CMFP</td>
</tr>
<tr>
<td></td>
<td>[.PANEL=((name[, ver, mod]),...)]</td>
<td>Except CMEP, CMFP</td>
</tr>
<tr>
<td></td>
<td>[.PROGRAM=((name[, ver, mod]),...)]</td>
<td>DPPX, RPS, NDMT only</td>
</tr>
<tr>
<td></td>
<td>[.PRINT=(name,...)]</td>
<td>VSE MEMBER, PANEL, or PROGRAM resource types only</td>
</tr>
<tr>
<td></td>
<td>[.LIBNAME=library-name]</td>
<td>VSE MEMBER, PANEL, or PROGRAM resource types only</td>
</tr>
<tr>
<td></td>
<td>[.LMTYPE=library/member type]</td>
<td>VSE MEMBER, PANEL, or PROGRAM resource types only</td>
</tr>
<tr>
<td></td>
<td>[.SUBLIB=sublibrary-name]</td>
<td>VSE MEMBER, PANEL, or PROGRAM resource types only</td>
</tr>
<tr>
<td></td>
<td>[.SRVGRP=(group_name(disposition))]</td>
<td>User-defined types only</td>
</tr>
</tbody>
</table>

NAME

Specify the name of the group. You can specify 1-8 alphanumeric characters. The first character must be alphabetic.

Note: Do not use these special characters in a group name: $, #, @, %, _. GIIX may
DEFINE GROUP/DELETE GROUP

interpret the underscore character as a blank and IOF cannot interpret the other special characters, producing unreliable results.

**NODETYPE** Specify the node types in a group of nodes, or the type of node to which the resources in a group of resources belong. You can omit this operand if you want to define a group of LU 6.2 nodes that have different node types, as long as each LU 6.2 node type has either CMIF or CMFP functional capabilities.

**NODE** Specify the name of one or more nodes that you are adding to, or deleting from, a group of nodes. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. All nodes must be of the same type for LU0 node types. You can add nodes to a group before you define them in the DRD. NetView DM for MVS does not verify the node authorization class or the consistency of the node types of the nodes in the group except when you define the SRVGRP parameter. The maximum number of node names you can specify is 2000.

**RESOURCE, CATE, CATU, CLIST, DSCB, DSN, DUMP, FP, FSP, MEMBER, PANEL, PROGRAM, PRINT**
These operands refer to different resource types. Use one of these operands instead of the NODE operand, to add or delete the specified resource names from a group of resources. The maximum number of resource names you can specify is 50.

**RESTYPE** Specify the resource name or type code (see Table 74 on page 566). You do not have to specify this operand if the resource type is microcode and the first token of the resource name is MCODE or MCUST. In this case, you cannot specify RESOURCE=ALL when you are using the DELETE GROUP statement.

**LIBNAME** This operand is only used with member, panel, or program resource types. It specifies the name of the library.

**LMTYPE** This operand is only used with member, panel, or program resource types. It specifies the type of library or member. You can specify one of these libraries:

- C  Core image library
- P  Procedure library
- R  Relocatable library
- S  Source library

For member resource types, you can specify up to eight alphanumeric characters.

**Note:** If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

**SUBLIB** This operand is only used with member, panel, or program resource types. It specifies the name of the sublibrary.

**Note:** If the library type is S (LMTYPE=S), you must specify a sublibrary name.

**SRVGRP** This operand is optional and divided into two fields. The first field specifies the server group name (from 1 to 8 alphanumeric characters, the first of which is alphabetic). There is no default for this field; you must specify a name if you are using the SRVGRP operand. The second field (disposition) specifies the group characteristics:

- NEW, if the group is new (the default)
- OLD, if the group must be deleted and then created

If you use this operand to create a server group for a group that is already defined, you must specify the disposition. If you use this function to create a server group for a node group that is already defined, you can omit the NODE operand and just use the SRVGRP operand.
DEFINE GROUP/DELETE GROUP

**Examples:** Here is an example of a DEFINE GROUP statement to create a group of nodes:

```
DEF GROU NAME=NEWNODES,NODETYPE=ENODES,NODE=(ROMECM,FLORCM,NAPOCM)
```

Later, you can use the following statement to add another node to an existing group:

```
DEF GROU NAME=NEWNODES,NODETYPE=ENODES,NODE=(VENICM)
```

The following statement is defining a group of CLIST resources:

```
DEF GROU NAME=PAYCLIST,CLIST=((PAY1,/zerodot1,/zerodot2),(PAY2,/zerodot1,/zerodot2)),RESTYPE=CLIST
```

Use the following statement to delete the entire group of resources:

```
DEL GROU NAME=PAYCLIST,CLIST=ALL
```
PRINT GROUP

Use the PRINT GROUP statement to print all of the node groups for a specified node type, or all of the resource groups for a specified resource type.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRInt GROUp</td>
<td>[NODETYPE=node-type</td>
<td>RESTYPE=resource-type]</td>
</tr>
</tbody>
</table>

NODETYPE Specify the node type. Node types are listed in Table 70 on page 561 and Table 71 on page 562. You can omit NODETYPE, even if you do not specify RESTYPE. If both of these operands are omitted, NetView DM for MVS will print all the groups of nodes that contain nodes with different node types.

RESTYPE Specify the resource type name or type code, as described in “NetView DM for MVS Resource Types” on page 566.

Example: Here is an example of a PRINT GROUP statement to print all of the existing group definitions for a specified node type:

PRI GROU NODETYPE=SSP

Or, you can print all of the definitions for a specific resource type:

PRI GROU RESTYPE=JOB
Using Transmission Profiles

Transmission profiles provide an efficient way of using transmission lines. A transmission profile is a way of grouping together nodes which have the same transmission characteristics, referred to in this book as transmission profile attributes.

The most important transmission profile attribute is the type of transmission line used between the host and the logical units (LU}s) associated with the target nodes (the nodes to which you are transmitting). The lines can either be *leased* or *switched*. This is defined in the LINETYPE parameter of the node. So, you could define a separate transmission profile for nodes connected to NetView DM for MVS in each of the following different ways:

- **Dial-in lines**
- **Dial-out lines**
- The same *multipoint* line
- *Point-to-point leased* lines

The other transmission profile attributes are:

- Minimum number of transmission tasks. This is the number of *concurrent* tasks that can take place for a node.
- Maximum number of transmission tasks.
- Retry interval (in seconds) for a line failure.
- Number of times NetView DM for MVS will attempt an automatic transmission retry.

Transmission profiles are defined at installation time, using a macro called NDMTP, but you can change them later using a control statement called CHANGE LUTP. This control statement reassigns logical units to a transmission profile. You can also print a list of the names of logical units associated with one transmission profile, or all transmission profiles, using the PRINT TPLU statement. You can redefine transmission profile parameters, such as line type or maximum number of tasks, using the Interactive Operator Facility (IOF), described in Chapter 23, “Changing Transmission Profiles” on page 521.

Using Connection Profiles

You can assign each logical unit a specific connection profile (also referred to as a “CP”). A connection profile is a set of attributes that are common to a collection of logical units associated with nodes defined in the DRD.

Connection profiles are defined at installation time, using the NDMCP customization macro. The NDMCP macro specifies this common set of attributes. NetView DM for MVS uses these attributes to recognize the connection capabilities of the logical units assigned to a specific CP.

Up to 1000 CPs can be defined at installation time, using this macro once for each CP you want to define. Refer to the *Installation and Customization* book for information about the NDMCP macro and setting up connection profiles.

You can use a set of SUBMIT control statements to define or change the assignment of logical units to a specific connection profile. You can also deassign a logical unit from a connection profile.
You can use the following SUBMIT control statements to work with logical units:

- ASSIGN LUCP, to assign a logical unit, or units, to a connection profile
- CHANGE LUTP, to reassign a logical unit, or units, to a transmission profile
- DEASSIGN LUCP, to deassign a logical unit, or units, from a connection profile
- PRINT CPLU, to print the assignments of logical units to connection profiles
- PRINT TPLU, to print the assignments of logical units to transmission profiles.

In order for this change to have an immediate effect when the TCP is running with the NETCHNG startup parameter set to IMMEDIATE, you must also run Batch Utilities with NETCHNG=IMMEDIATE in the batch profile. The change will become effective:

- Immediately, if there is no active SNA session with the logical unit affected by the change
- At the end of the session, if an SNA session is active

The specified transmission profile or connection profile must already have been defined when the TCP was last cold-started.

The syntax and operands of the above statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.
ASSIGN LUCP

Use the ASSIGN LUCP statement to assign one or more logical units known by NetView DM for MVS (logical units having at least one associated node) to a connection profile (CP). If a logical unit is already assigned to another CP, the execution of this statement results in changing the assignment. Refer to the Installation and Customization book for information about how to define connection profiles using the NDMCP macro.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASSIGN LUCP</td>
<td>LUNAME=(name1,name2,...,name50)</td>
<td>CMEP, CMFP and NDMT nodes only</td>
</tr>
<tr>
<td></td>
<td>CPNAME=name</td>
<td></td>
</tr>
</tbody>
</table>

LUNAME         Names of one or more logical units to be assigned to the profile referred to by CPNAME. Up to 50 names can be specified.

If any of the logical unit names is not known by NetView DM for MVS, the processing of the statement is bypassed and an error message is issued.

CPNAME         Name of the connection profile. If the named profile is not defined, the processing of the statement is bypassed and an error message is issued.

The name must be up to 8 alphanumeric characters long, the first character of which is alphabetic.

Connection profiles (and their names) are defined at customization time with the NDMCP customization macro. See the Installation and Customization book.
CHANGE LUTP

Use the CHANGE LUTP control statement to reassign one or more logical units (LUs) to a transmission profile. Before you run this statement, you should redefine the transmission profile, or profiles, to establish the correct assignment of logical units to the new transmission profiles. You redefine transmission profiles using the generation macro called NDMTP, described in the *Installation and Customization* book.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE LUTP</td>
<td>LUNAME=(name1,name2,...,name50),TPNAME=name</td>
<td>All node types</td>
</tr>
</tbody>
</table>

**LUNAME** Specify the name of one or more logical units (up to 50) that you want to assign to the transmission profile named in the **TPNAME** operand. If NetView DM for MVS cannot find any of the logical unit names that you specify, it does not process the statement, and issues an error message.

**TPNAME** Specify the name of the transmission profile assigned to the logical units. If NetView DM for MVS cannot find the transmission profile you specify, it does not process the statement, and issues an error message.

**Note:** As a result of the CHANGE LUTP statement, the **LINETYPE** of the nodes associated with the logical unit, or units, being assigned may change. This happens if the **TPTYPE** of the transmission profile referred to by the **TPNAME** is different from the **TPTYPE** of the transmission profile to which the logical units are currently assigned.

**Example:** Here is an example of a CHANGE LUTP statement to reassign logical units to a transmission profile:

```
CHA LUTP LUNAME=(AAA,AAAA),TPNAME=NDMTPL
```
DEASSIGN LUCP

DEASSIGN LUCP

Use the DEASSIGN LUCP control statement to cancel the assignment of one or more logical units to a connection profile. If a logical unit is not assigned to a connection profile, the execution of this statement has no result.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAssign LUCP</td>
<td>LUNAME=(name1,name2,...,name50)</td>
<td>CMEP, CMFP and NDMT nodes only</td>
</tr>
</tbody>
</table>

LUNAME

Names of one or more logical units that have to be deassigned from their connection profiles. Up to 50 names can be specified.

If any of the logical unit names is not known by NetView DM for MVS, the processing of the statement is bypassed and an error message is issued.
PRINT CPLU

Use the PRINT CPLU statement to print a list of logical units assigned to a Connection Profile (CP).

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT CPLU</td>
<td>CPNAME=name</td>
<td>CMEP, CMFP and NDMT node types only</td>
</tr>
</tbody>
</table>

CPNAME

Name of the connection profile. If the named profile is not defined, the processing of the statement is bypassed and an error message is issued.

The name must be up to 8 alphanumeric characters long, the first character of which is alphabetic.

Connection profiles (and their names) are defined at customization time with the NDMCP customization macro. See the *Installation and Customization* book.
PRINT TPLU

Use the PRINT TPLU control statement to print the logical pages of one or more transmission profiles. The logical pages list the names of the logical units that are associated with the transmission profile. If you specify a TPNAME, only the logical units associated with this transmission profile are listed. If you do not specify a name, all of the defined transmission profiles are listed.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT TPLU</td>
<td>TPNAME=name</td>
<td>All node types</td>
</tr>
</tbody>
</table>

TPNAME Specify the name of the transmission profile.

Example: Here is an example of a PRINT TPLU statement to print all of the names of the logical units associated with the transmission profile called NDMTPL:

PRI TPLU TPNAME=NDMTPL
Chapter 14. Defining and Assigning Resources

This chapter explains how to define, delete, and print resources, and how to assign and deassign resources to nodes.

You use the following control statements to work with resources for LU0 node types:

- ADD statements, to define resources in the DRD and assign a resource, or group of resources, to a node or group of nodes
- DEASGN statements, to delete resource assignments
- DELETE DEFINITION, to delete a resource assignment, and then delete the resource definition from the DRD
- PRINT DEFINITION, to print all of the resource definitions for a specified resource type

If you are working with NetView DM for MVS user-defined nodes (CMEP, CMFP, NDMT), you use a specific set of control statements instead:

- ADD RESOURCE, to assign a resource, or group of resources, to a node or group of nodes
- DEASGN RESOURCE, to delete resources from specific nodes
- DEFINE RESOURCE, to define a resource in the DRD

There are also two control statements that you use with all resource types to track the operations. These are:

- DELETE HISTORY, to delete a resource history record associated with a resource
- PRINT HISTORY, to print a resource history record associated with a resource

This chapter is divided into these three categories of control statements. Within each section, the statements are described in alphabetical order. But first, read the following pages to find out about resource definitions, resource types, naming resources, and tracking resources using resource history records.

What Is a Resource Definition?

A resource definition describes resources in the network to NetView DM for MVS. Resource definitions are stored in the DRD. There are some resource types that you must define to NetView DM for MVS before users can validate a transmission plan that refers to the resource. Table 20 on page 241 shows which resource types must be defined. If a transmission plan refers to a resource which you have not defined, NetView DM for MVS issues a warning or error message when it tries to validate the plan.

When you define a resource, you must specify:

- The type of node the resource can be transmitted to
- The resource type, listed in Table 20 on page 241
- A symbolic name which identifies the resource
- A set of resource attributes for the resource at the host and at the node

After you define nodes and resources, you can assign the resource to a node or group of nodes. You can also organize nodes and resources into groups, and track resources in resource history records, as described in “Organizing Nodes and Resources into Groups” on page 225 and “Tracking Resources” on page 247.
Resource Types

Resources are classified into types. These types fall into these four broad categories:

- Microcode resources, such as microcode, microcode customization data, and related changes, updates or patches
- Software resources, such as system software, application software, and related updates and changes
- Application data, such as flat data, relational data, application logs, and related updates
- Problem documentation resources, such as dumps, configuration files, traces, and error logs

Throughout this book, resources are referred to as either LU0 resource types or LU 6.2 resource types. LU0 resource types are those resources that are supported by LU0 node types (shown in Table 70 on page 561). Here are the LU0 resource types:

- CLIST
- Dataset
- Errorlog
- Job
- Library
- Member
- Message out
- Panel
- Print
- Program
- Program temporary fix (PTF)
- Storage dump

LU 6.2 resource types are those supported by LU 6.2 node types (shown in Table 71 on page 562). Here are the LU 6.2 resource types:

- AS/400 data object containment structure
- AS/400 object
- Configuration file
- Dump
- Error log
- Flat data
- Microcode and microcode customization
- Procedure
- Relational data
- Software
- Trace

NetView DM for MVS supplies some initial values which indicate whether or not to define, assign, track, or group specific resource types. For LU 6.2 resource types, you can change these initial values at installation time using a macro called NDMRES, or after installation using a partial installation step (GENTYPE=PROFILE). Refer to Installation and Customization for information.

Table 20 on page 241 shows the resource types you can define, and indicates the initial values for defining, assigning, tracking, and grouping resources.
Table 20. Resource types and initial values

<table>
<thead>
<tr>
<th>Type</th>
<th>DEF</th>
<th>RES</th>
<th>ASS</th>
<th>TRA</th>
<th>GRP</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data-object containment structure</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0158</td>
</tr>
<tr>
<td>(CMEP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS/400 object (CMEP)</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0156</td>
</tr>
<tr>
<td>CLIST</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0040</td>
</tr>
<tr>
<td>Configuration file</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0220</td>
</tr>
<tr>
<td>Data set</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0030</td>
</tr>
<tr>
<td>Dump (CMEP)</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0230</td>
</tr>
<tr>
<td>Errorlog (DPPX)</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>0034</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0250</td>
</tr>
<tr>
<td>Flat data</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0100</td>
</tr>
<tr>
<td>Job</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>0041</td>
</tr>
<tr>
<td>Library</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>0052</td>
</tr>
<tr>
<td>Member</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y/N</td>
<td>0050</td>
</tr>
<tr>
<td>Microcode and microcode customization</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>Y</td>
<td>Y/N</td>
<td>0060</td>
</tr>
<tr>
<td>Panel</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0036</td>
</tr>
<tr>
<td>Print</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>Y/N</td>
<td>0038</td>
</tr>
<tr>
<td>Procedure</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0080</td>
</tr>
<tr>
<td>Program</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0042</td>
</tr>
<tr>
<td>Program temporary fix (PTF)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>0044</td>
</tr>
<tr>
<td>Relational data</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0120</td>
</tr>
<tr>
<td>Software</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y/N</td>
<td>0070</td>
</tr>
<tr>
<td>Storage dump (LU0)</td>
<td>Y/N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>0032</td>
</tr>
<tr>
<td>Trace</td>
<td>N</td>
<td>N</td>
<td>Y/N</td>
<td>N</td>
<td>Y/N</td>
<td>0240</td>
</tr>
</tbody>
</table>

The first column (Type) shows the resource types that you can use with NetView DM for MVS.

The second column (DEF) indicates whether or not you have to define the resource in the DRD. There are some types of resource that you do not have to define. The values can be:

- **Y (Yes)** You must always define resources of this type.
- **N (No)** You do not have to define resources of this type.
- **Y/N (Yes/No)** You can choose whether or not to define resources of this type.

**Note:** If you change an initial values from YES to NO for a resource type, then you will no longer be able to print or delete already existing definitions in the DRD for this resource type.

The third column (RES) indicates whether the resource is automatically restricted, optionally restricted, or not restricted. You can change this default value when you are defining the resource. If you want to assign the resource to a node, you must define it as restricted. The values are:

- **Y (Yes)** The resource type is automatically defined as restricted.
- **N (N)** The resource type is unrestricted, or you do not have to define it at all.
- **Y/N (Yes/No)** If you define the resource, it is automatically restricted. If you choose not to define it, the resource is unrestricted.
The fourth column (ASS), indicates whether or not you must assign the resource. If a resource is defined as restricted, you must assign it to a node, or group of nodes. NetView DM for MVS will only transmit these resource types to, or receive or delete these resource types from, those nodes to which you have assigned the resource. You must already have defined the node(s) to NetView DM for MVS. The values can be:

- **Y (Yes)** You must assign the resource.
- **N (No)** You cannot assign the resource.
- **Y/N (Yes/No)** You can assign the resource only if it is defined as restricted.

The fifth column (TRA) indicates whether or not the resource is tracked when you define it. NetView DM for MVS tracks resources in resource history records. These are explained in “Tracking Resources” on page 247. The values can be:

- **Y (Yes)** NetView DM for MVS automatically tracks the resource.
- **N (No)** NetView DM for MVS does not track the resource.
- **Y/N (Yes/No)** You can choose whether or not to track the resource.

The GRP column indicates whether or not you can organize resources of this type into groups. Refer to “Organizing Nodes and Resources into Groups” on page 225 for information. The values can be:

- **Y/N (Yes/No)** You can choose whether or not to group the resources.
- **N (No)** You cannot group the resources.
Naming Resources

Each resource in your network must be uniquely identified to NetView DM for MVS. To do this, you give the resource a resource name. You define the name when you are defining the resource, either using GIX or Batch Utilities.

The conventions for defining a resource name vary between the different resource types. The following pages begin by describing the naming conventions for LU0 resource types, and then describe the conventions for LU 6.2 resource types. The table below summarizes the basic differences between LU0 resource type conventions, and those for LU 6.2 resource types.

<table>
<thead>
<tr>
<th>LU0 Resource Types</th>
<th>LU 6.2 Resource Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific naming conventions for each resource type</td>
<td>Enterprise-structured names, or structured names for microcode.</td>
</tr>
<tr>
<td>The name must be fully qualified</td>
<td>The name does not have to be fully qualified. You can use partial naming.</td>
</tr>
<tr>
<td>Names cannot use subtrees</td>
<td>Some resource types can use subtrees.</td>
</tr>
</tbody>
</table>

Table 21. Comparison of naming conventions for LU0 and LU 6.2 resources

Naming LU0 Resources

Each LU0 resource type has a different set of naming conventions that you should follow when defining resource names. The syntax conventions for each resource are described in Appendix C, “Resource Naming Conventions” on page 569.

Naming LU 6.2 Resources

With NetView DM for MVS, you use a global name for each file, consisting of a set of tokens. For example, in the global file name CORPID.9135.NA.PATCH.1234, CORPID is a token, 9135 is a token, PATCH is a token, and so on. A global name can have up to 10 tokens separated by periods, and each token can have up to 16 characters (uppercase alphabetic and numeric characters only). The total name length cannot be more than 64 bytes, including the separator characters.

LU 6.2 resource types use enterprise-structured names. Enterprise-structured means that each company, or "enterprise," can design the name of its own resources. The only predefined token is the first token, which is the corpid. The corpid is the enterprise code of your company, defined in the structured netid assigned to your company, and registered, by IBM. As for the remaining tokens, you can specify whatever you want.

This is the basic format of an enterprise-structured name:

corpid.token.token.token . . . . token

If you are using Microcode, Software, Flat data, or Relational data resources, then you can also use subtree and version tokens in the resource name. A subtree is a set of tokens that already have a structure defined. You specify a leading token before the subtree, called a subtree identifier (STI) token. The STI token shows the type of object that you are naming. For NetView DM for MVS distribution or change control requests, you can use the following STI tokens:

2 There is also an STI token called GRP (Group object), but this is only used for local definition.
Naming Resources (System Administrator)

REF Refresh object
UPD Update object
FIX Fix object
LIB Library object
MEM File member object
OBJ AS/400 save restorable object type

Each STI token determines a predefined subtree structure, optionally followed by version tokens. Here is the structure of an enterprise-structured name using a subtree:

corpid(mm).(mm or nnn)....(mm or nnn).subtree

where the subtree format is:

(mm).(mm or nnn)....(mm or nnn)

mm refers to tokens that must match. nnn refers to tokens that need not match. For each of the STI tokens, these are the subtree structures that are defined. mm in brackets shows that the token must match.

REF(mm).level
UPD(mm).oldlevel.newlevel
FIX(mm).level.fixid(mm)
LIB(mm).libraryname.targetrelease
MEM(mm).libraryname.filename.membername.targetrelease
OBJ(mm).libraryname.objectname.objecttype.targetrelease

You can also use one or more version tokens in the resource name, for example, to distinguish between the French and German versions of a change object.

Example: Here is an example of an enterprise-structured name which uses both a subtree and a version token:

DUNKIN.DONUTS.MESSAGES:UPD.12.13.FRENCH

This global name describes the level 13 update from level 12 for the French version of the Messages sub-component of the Donuts' application owned by the Dunkin enterprise. The " : " introduces the subtree, which in this case, is UPD.12.13.

Microcode is the only LU 6.2 resource type which can use a convention other than enterprise-structured names. Microcode and microcode customization data can also use structured names, in which the first token must be MCODE or MCUST, respectively. Whenever you specify MCODE or MCUST as the first token of a name, using GIX or the SUBMIT or MAINTDAT Batch Utilities, NetView DM for MVS checks to make sure that the resource type is microcode, identified by a resource type code of 0060.

Whether you are using a structured name for microcode, or an enterprise structured name for any LU 6.2 resource type, you can also use partial naming, instead of specifying all the qualifiers in a name. The next section describes how to use partial naming.

Partial Naming and Matching for LU 6.2 Resource Types

With NetView DM for MVS you can partially name a resource definition and partially match resources in a transmission plan.
Partial Naming for Resource Definitions: For LU 6.2 resources only, you can perform partial naming of resources, whereby you can:

- Specify defaults for a resource name definition at customization time
- Update these values whenever necessary
- Define a unique resource name by a partial name so you will not have to repeat the definition and assignment for each resource name in the network.

When you define a specific resource, you can enter a resource name that is a partially-defined name. A name is partially defined when one or more tokens of the name are set to a percent sign (%).

Resource Name Masks: Partial naming is possible because at customization time you defined a resource name mask for the resource type of the resource you are defining. You can, for each type of LU 6.2 resource, specify one general mask and up to 10 particular masks. These allow NetView DM for MVS to verify a resource name during a resource definition.

Each resource name mask contains 10 values separated by periods (.), where each value corresponds to one possible token position in a resource name. These values determine whether NetView DM for MVS requires a specific value to be given to a token (Y), or whether it can be set to % (N), during the resource name definition.

For example, a resource name mask might look like this:

```
Y.N.Y.N.Y.Y.Y.N.N
```

If no resource masks are specified for a certain resource type at customization time, this default value is assumed for that resource type:

```
Y.Y.Y.Y.Y.Y.Y.Y.Y
```

The first token of a resource name is either:

- corporation id
- or
- MCODE or MCUST

Therefore, the first value in a resource name mask must either be Y, or a specified value that NetView DM for MVS explicitly compares with the corresponding value that you enter.

- Example of general mask:
  
  `RMASK1=Y.Y.Y.N.N.Y.Y.Y.N.N`

- Example of particular mask:
  
  `RMASK2=Mycorporationid.Y.Y.Y.Y.Y.Y.N.N.N`

A general mask must have Y in the first position; a particular mask must have an explicit value in the first position.

To find out how the masks have been defined in your installation of NetView DM for MVS, refer to the NDMRES customization macro defined in the customization JCL used to customize NetView DM for MVS. The Installation and Customization manual contains information about the NDMRES macro, and the JCL used to install NetView DM for MVS.
Name Resources (System Administrator)

**Resource Partial Naming Using Resource Name Masks:** The general mask is used by NetView DM for MVS during definition of a resource name when the first token that you specify does not match any of the explicit values in any defined particular masks you might have defined.

Here is an example of resource definition using partial names.

- A file is produced by an application program every day and has to be distributed to the nodes to which it is assigned.
- The name of the file consists of four tokens. Three of the four tokens are always the same and the fourth represents the date.

You define the resource (and assign it), once, by specifying an explicit value for the three fixed tokens, and % for the fourth one, as shown below:

```
SMITH.APPL1.DATA.%
```

This is possible because for the corresponding resource type, you defined the following *particular mask*:

```
RMASK3=SMITH.Y.Y.N.N.N.N.N.N
```

**Partial Matching of Resources in Transmission Plans:** Use partial matching in transmission plans for finding resources to send, retrieve, or delete a resource. When you prepare a transmission plan you can use *partial naming qualification*, specifying an asterisk (*) instead of a qualifier in one or more of the tokens of the resource name.

A partially matched resource name lets you write a function in a transmission plan in a *generalized way*. However, the *qualified tokens* in the resource name (as specified in your plan) must be sufficient to identify a unique resource definition in the DRD and a unique object in the source/target repository.

For the SEND function, the searched repository is the resource repository of NetView DM for MVS. For a RETRIEVE or DELETE function, the searched repository is the repository of the destination node.

If you specify a partial name in a transmission function and the resource (in the resource repository) belongs to a resource type that requires the existence of a resource definition (in the DRD), NetView DM for MVS looks for a resource that has this definition in the DRD:

- A % value in the resource name definition for those tokens indicated by * in the transmission plan. (In the resource name mask this token would be set to N.)
- The *exactly matching values* in the resource name definition for the other tokens. (In the resource name mask these tokens would be set to Y.)

Here is an example of a transmission function using partial matching:

- This file, for the current day, exists in the resource repository:
  
  SMITH.APPL1.DATA.911201

- You prepare a transmission function that looks like this:
  
  SEND RESO NAME=SMITH.APPL1.DATA.*, RESTYPE=FLATD, MATCHIND=(4,H),

- When the plan is submitted, the most recent version of the resource, that is SMITH.APPL1.DATA.911201, is fetched from the resource repository and sent to the specified node, *if the following resource definition exists in the DRD*:
  
  SMITH.APPL1.DATA.%
Tracking Resources

For some resource types, shown in Table 20 on page 241, you can create a resource history record when you define and assign a resource. These records are stored in the DRD, and record date and time stamps when resources are assigned, stored, deleted, installed, accepted, removed, backed up, uninstalled, executed and cleared, depending on the resource type. For LU 6.2 resource types, you can change the default tracking value at installation time with the NDMRES customization macro.

The operations that are recorded vary according to the resource type. The Status column of Table 22 on page 248 shows which operations are recorded. The time and date is recorded for the following operations:

- **Ac**: When a resource was last accepted.
- **As**: When a resource was last assigned. This status only applies to assignable resources, as defined at installation time.
- **B**: The last backup log.
- **C**: The last time resources were cleared.
- **D**: The last deletion of a resource.
- **E**: The last execution of a CLIST.
- **I**: The last installation of a resource.
- **R**: The last time a resource was removed.
- **S**: The last time a resource was stored.
- **U**: The last time a resource was uninstalled.
NetView DM for MVS updates these records whenever a function is successfully completed, if that function implies a change in the status of the resource: This means whenever:

- A resource is successfully assigned to a node.
- A resource is successfully sent, deleted, cleared, executed, installed, accepted, or removed.
- A resource is successfully deleted, reinstalled, or backed up as a side effect of an accept, remove, or install removably function.
- A component is successfully uninstalled from a node. In this case, the node will report all of the associated resource names and NetView DM for MVS will mark these resources as *uninstalled*.

These records are also updated when NetView DM for MVS receives unsolicited reports from nodes for functions that are executed locally.

### SUBMIT Control Statements for LU0 Resource Types

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

**Table 22. Resource history information for resource types**

<table>
<thead>
<tr>
<th>Resource type</th>
<th>Name at Node</th>
<th>Version or Mod.</th>
<th>Extent</th>
<th>Subcat ID</th>
<th>Key ID</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 object (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
</tr>
<tr>
<td>AS/400 data object containment structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
</tr>
<tr>
<td>Sub/Category</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/C/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sub/Category update</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/E/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration file</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>Yes</td>
<td></td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dump (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flat data</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library/Sublibrary</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Member</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microcode</td>
<td></td>
<td></td>
<td></td>
<td>Ac/As/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panel</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td></td>
<td></td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D/E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/D/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTF</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>As/I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational data structure</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td>As/Ac/B/D/I/R/S/U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td>As/S/D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ADD Statements

Use a set of ADD statements to define LU0 resources to the DRD, and then assign a resource, or group of resources, to a node, or group of nodes. You must already have defined the nodes to which you are assigning the resource, or resource group, otherwise NetView DM for MVS will reject your SUBMIT request. The resources that you are assigning do not have to be defined already:

- If you did define the resource, it is assigned to the node as requested, as long as you defined the resource as restricted. The resource definition does not change when you assign the resource.
- If you have not defined the resource to the DRD, the ADD statement defines it with the following attributes:
  - The resource is restricted
  - The resource authorization class is the default, specified in your user profile
  - VSE data sets have their “Organization” set to SAM
  - RPS data sets have “Share” set to “Add exclusive.”

SUBMIT automatically initializes a resource history record for the resource to track the resource information. It records the date and time the resource is assigned.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD CATegory</td>
<td>{NODE=node-name}</td>
<td>DPPX only</td>
</tr>
<tr>
<td>ADD CATUpdate</td>
<td>GROUP=node-group</td>
<td>VSE only</td>
</tr>
<tr>
<td>ADD CLiSt</td>
<td>{NAME=resource-name}</td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>ADD DATAset</td>
<td>GROUPID=resource-group</td>
<td>DPPX only</td>
</tr>
<tr>
<td>ADD DSCB</td>
<td>[DSAC=X'hhhh']</td>
<td></td>
</tr>
<tr>
<td>ADD FP</td>
<td>[.NODELNAME=library-name-at-node]</td>
<td></td>
</tr>
<tr>
<td>ADD FSP</td>
<td>[.NODENAME=name-at-node]</td>
<td></td>
</tr>
<tr>
<td>ADD LIBRary</td>
<td>[.PROTECT=</td>
<td></td>
</tr>
<tr>
<td>ADD PANEL</td>
<td>[None</td>
<td>Write</td>
</tr>
<tr>
<td>ADD PROGram</td>
<td>[.SRH=([UPD]ADDEXCL</td>
<td>ADDSHR)]</td>
</tr>
<tr>
<td>ADD PTF</td>
<td>[.VOLSER=volserial]</td>
<td></td>
</tr>
</tbody>
</table>

You must always specify either NODE= or GROUP=, and either NAME= or GROUPID=. See Table 23 on page 250 for information about which operands to use with each resource type.

NODE GROUP

Specify the name of the node, or group of nodes, that you are defining resources for. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. You must already have defined the node, or node group, in the DRD. For a group of nodes, resource entries and status information are assigned to every node in the group.

NAME GROUPID

Specify the name of the resource, or group of resources, that you are defining and adding to the node, or group of nodes. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names. Resource group names consist of from 1 to 8 alphanumeric characters, starting with a letter. When you assign a data set control block (DSCB), both its extents and all eight keys are assigned. You can only define specific keys using GIX.
ADD

DSAC Specify the data set authorizations to be verified when the resource is accessed at the node. You define these credentials according to DPPX rules. Specify four hexadecimal digits. Do not specify X'FFFF'.

NODELNAME Specify the name by which a library resource is known at the node.

NODENAME Specify the fully qualified name of the resource at the node. You cannot enter wildcards. For example, if you enter NODENAME".*.*", NetView DM for MVS treats it as if you did not specify a name at the node.

Refer to “Naming LU 0 Resources at the Node” on page 570 for information about how to name LU0 resources at the node.

PROTECT Specify the level of access for which authorization is required at the node. “Catauth” means that the resource is to have the same level of protection as its catalog.

SHR Specify the types of concurrent access to the resource to be allowed at the node.

VOLSER Specify the serial number (from 1 to 6 alphanumeric characters) of the DPPX volume on which a program temporary fix (PTF) is stored when it is transmitted.

Note: The following resources are considered to be sub-resources of a main resource type. They are implicitly assigned along with a main resource, as follows:

Keys Keys are implicitly assigned along with the DSCB to which they refer. When you ADD a DSCB that has not yet been defined in the DRD, SUBMIT defines it with a default configuration of all eight keys. You can only define specific keys using GIX.

Subcategories Subcategory Updates These are implicitly assigned when the category or category update to which they belong is assigned. When you ADD a category or category update that has not yet been defined in the DRD, SUBMIT defines it with a default configuration of all 16 subcategories or subcategory updates. You can only define specific subcategories or subcategory updates using GIx.

VSE Members VSE Panels VSE Programs VSE Sublibraries These are implicitly assigned along with the library to which they belong.

Usage: Table 23 shows which operands are mandatory, and which are optional, for each resource type.

<table>
<thead>
<tr>
<th>These:</th>
<th>Can only be used with:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSAC</td>
<td>CLIST DATaset PANEL PROgram</td>
</tr>
<tr>
<td>NODENAME</td>
<td>CLIST DATaset PANEL PROgram</td>
</tr>
<tr>
<td>SHR</td>
<td>CLIST DATaset PANEL PROgram</td>
</tr>
<tr>
<td>NODELNAME</td>
<td>LIBRARY</td>
</tr>
<tr>
<td>PROTECT</td>
<td>CLIST DATaset PANEL PROgram</td>
</tr>
<tr>
<td>VOLSER</td>
<td>CLIST DATaset PANEL PROgram PTF</td>
</tr>
</tbody>
</table>

Table 23. Applicability of ADD operands
DEASGN Statements

Use a set of DEASGN statements to delete resource assignments. Deleting a resource assignment means that the resource, or group of resources, are no longer assigned to the specified node or nodes, although they still remain defined in the DRD. If you want to delete a resource definition completely from the DRD, then you must use the DELETE DEFINITION statement, described in “DELETE DEFINITION” on page 252.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAsgn CATEgory</td>
<td>{NODE=node-name}</td>
<td>All node types</td>
</tr>
<tr>
<td>DEAsgn</td>
<td>GROUP=node-group</td>
<td></td>
</tr>
<tr>
<td>CATUpdate</td>
<td>{NAME=resource-name}</td>
<td></td>
</tr>
<tr>
<td>DEAsgn CLISt</td>
<td>GROUPID=resource-group</td>
<td></td>
</tr>
<tr>
<td>DEAsgn DATAset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn DSCB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn FP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn FSP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn LIBRary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn PANEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn PROGram</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEAsgn PTF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NODE GROUP**

Specify either the name of the node, or the group of nodes, to which the resource is currently assigned. Specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. The node or node group must already be defined in the DRD. In the case of a group of nodes, resource entries and status information are deassigned from each node in the group.

**NAME GROUPID**

Specify either the name of the resource, or group of resources, to be deassigned from the node or nodes. Appendix C, “Resource Naming Conventions” on page 569 describes how to specify LU0 resource names. Group names consist of from 1 to 8 alphanumeric characters, the first of which must be alphabetic.

**Example:** Here is an example of a DEASGN statement to delete the assignment of a category resource type from an SSP type node:

```
DEA CATE NODE=MYSSP,NAME=TEST
```
DELETE DEFINITION

Use the DELETE DEFINITION statement (for both LU0 and LU 6.2 resources) to delete both a resource assignment and the actual resource definition. This statement causes SUBMIT to scan the DRD, looking for all the assignments of the specified resource. When it finds a resource assignment, the assignment is deleted, providing the assignment is to a node whose node class the user is authorized to work with. Once all of the assignments are found and deleted, NetView DM for MVS deletes the resource definition from the DRD.

Once all of the assignments are found and deleted, NetView DM for MVS deletes the resource definition from the DRD. If you want to purge the DRD completely of records relating to the named resource, use the DELETE HISTORY control statement. This deletes the record used to track the resource history from the DRD.

**Note:** If an assignment is found to a node class that the user is not authorized to work with, then the execution of this statement is halted. The assignment is not deleted, and the resource definition is not deleted.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete DEFInition</td>
<td>NAME=resource-name, RESTYPE=resource-type, [NODETYPE=node-type]</td>
<td>All node types</td>
</tr>
</tbody>
</table>

**NAME**  
Specify the resource name. This can be a partial name, if required. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**RESTYPE**  
Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

**NODETYPE**  
Specify the node type that the resource you are deleting belongs to.

**Example:** Here is an example of a DELETE DEFINITION statement to delete the definition of a software resource:

```
DEL DEFI NAME=IBM.TAMPA.13.FIX1.PSNXID.PSNXREF, RESTYPE=0870
```
PRINT DEFINITION

Use the PRINT DEFINITION control statement to print either a single resource definition, or all of the resource definitions contained in the DRD.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT DEFINITION</td>
<td>RESTYPE=resource-type [NAME=resource-name]</td>
<td>All node types</td>
</tr>
</tbody>
</table>

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

NAME Optional. Specify the name of the resource that you want to print. This can be a partial name if required. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

Example: Here is an example of a PRINT DEFINITION statement, to print all of the definitions of software resource types in the DRD:

PRI DEFI RESTYPE=0070
### ADD RESOURCE

Use the ADD RESOURCE statement to assign a resource, or group of resources, to a node or group of nodes.

#### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD RESOURCE</td>
<td>NAME=resource-name, GROUPID=group-name, NODE=node-name, GROUP=group-name, RESTYPE=resource-type, [PARTDEF=YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>

#### NAME

Specify the name of the resource. Some tokens of the resource name can be specified as %. This is a partial name definition. The tokens specified as % must be those that are set to N (need not match) in the resource mask. The other tokens must be completely qualified. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

#### GROUPID

Specify the resource group name. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. The group of resources must already be defined in the DRD.

#### NODE

Specify the name of the node to which you want to assign the resource, or group of resources. The node must already be defined in the DRD.

#### GROUP

Specify the name of the node group to which you want to assign the resource, or group of resources. The group of nodes must already be defined in the DRD.

#### RESTYPE

Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, although you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

#### PARTDEF

Specify whether the resource name is partially defined (YES), or completely qualified (NO).

#### Example:

Here is an example of an ADD RESOURCE statement to define and assign software to a node:

```
ADD RESOURCE NAME=IBM.TAMPA.13.FIX1.PSNXID.PSNXREF,
    RESTYPE=0070,
    NODE=PSNX1
```
DEASSIGN RESOURCE

Use the DEASSIGN RESOURCE control statement to deassign resources from specific nodes.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEAsgn RESOurce</td>
<td>NAME= resource-name</td>
<td>GROUPID= group-name ,NODE= node-name</td>
</tr>
<tr>
<td></td>
<td>,GROUP= group-name ,RESTYPE= resource-type</td>
<td></td>
</tr>
</tbody>
</table>

NAME Specify the name of the resource to be deassigned from the specified node, or group of nodes. Some tokens of the resource name can be specified as %. This is a partial name definition. The tokens specified as % must be those that are set to N (need not match) in the resource mask. The other tokens must be completely qualified. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

GROUPID Specify the name of the group of resources that will be deassigned from the specified node, or group of nodes. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. The group must already be defined in the DRD.

NODE Specify the name of the node from which the resource, or group of resources, is to be deassigned.

GROUP Specify the name of the group of nodes from which the resource, or group of resources, is to be deassigned.

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

Example: Here is an example of a DEASSIGN RESOURCE statement to deassign a software resource type from a CMEP node:

DEA RESO NAME=CORPID.CMEPRES,NODE=TMSLEW,RESTYPE=SOFTW
DEFINE RESOURCE

Use the DEFINE RESOURCE control statement to define a resource to the DRD.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE RESOurce</td>
<td>NAME=resource-name</td>
<td>NDMT, CMEP, CMFP only</td>
</tr>
<tr>
<td></td>
<td>,RESTYPE=resource-type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,CLASS=A0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,RESTRICT={YES}[ NO]]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,PARTDEF={NO}[YES]}</td>
<td></td>
</tr>
</tbody>
</table>

NAME Specify the name of the resource that you are defining. Some tokens of the resource name can be specified as % This is a partial name definition. The tokens specified as % must be those that are set to N (need not match) in the resource mask. The other tokens must be completely qualified. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST. If you do specify it in this case, you must specify 0060 for microcode or microcode customization type resources.

CLASS Specify the authorization class of the resource. You can enter two alphanumeric characters. The first character must be alphabetic. All users are authorized to use class A0 resources.

RESTRICT Specify whether or not the defined resource is restricted. You can only specify RESTRICT=NO for those resources which are optionally restricted and assigned.

PARTDEF Specify if the resource name is partially defined (PARTDEF=YES), or completely qualified (PARTDEF=NO).

Example: Here is an example of a DEFINE RESOURCE statement to define relational data:

```
DEF RESO NAME=QUANTUM.DATA.FILE.%.,RESTYPE=0120,RESTRICT=Y,CLASS=B1,
PARTDEF=Y
```
SUBMIT Control Statements for Tracking Resources

The SUBMIT control statements on the following pages are used to work with resource history records to track resources.

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

DELETE HISTORY

Use the DELETE HISTORY statement to delete resource history records from the DRD.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node/Resource Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE HISTORY</td>
<td>RESTYPE=resource type</td>
<td>See Table 24 on page 259</td>
</tr>
<tr>
<td></td>
<td>.NODE=node-name</td>
<td>VSE library, member, panel or program only</td>
</tr>
<tr>
<td></td>
<td>[.EXT={0</td>
<td>1}</td>
</tr>
<tr>
<td></td>
<td>[.LIBNAME=library-name]</td>
<td>VSE library, member, panel or program only</td>
</tr>
<tr>
<td></td>
<td>[.LMTYPE=library/member type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MEMBER=member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NAME=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBLIB=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

See Table 24 on page 259 for information about which operands to use with each resource type.

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

NODE Specify the name of the node.

EXT KEYNUMB Specify the extent (either 0 or 1), or key number (from 1 to 8 characters). EXT and KEYNUMB are mutually exclusive. If you do not specify either, the default is EXT=0 with no KEYNUMB.

LIBNAME Specify a library name (from 1 to 7 alphanumeric characters).

LMTYPE Specify the library type. You can specify one of the following:

- C Core image library
- P Procedure library
- R Relocatable library
- S Source library

For member resource types, you can specify up to eight alphanumeric characters.
Note: If you specify a library type of **S**, you must also specify a sublibrary name in the **SUBLIB** operand.

**MEMBER** Specify the name of the member.

**MOD** Specify the modification level of the resource. The range is from 00 to 99.

**NAME** Specify the name of the resource. The name must be fully qualified. It can be up to 64 characters. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**SUBCAT** Specify a subcategory ID. The acceptable range is from 1 to 16.

**SUBLIB** Specify the sublibrary name.

Note: If the library type is **S** (**LTYPE=S**), you must specify a sublibrary name.

**VER** Specify the version of the resource. The acceptable range is from 00 to 99.
Usage: Table 24 shows which operands are mandatory, and which are optional, for each resource type.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Mandatory Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub/category</td>
<td>0010</td>
<td>RESTYPE, NODE, NAME</td>
<td>SUBCAT, VER, MOD</td>
</tr>
<tr>
<td>Sub/category update</td>
<td>0012</td>
<td>RESTYPE, NODE, NAME</td>
<td>SUBCAT, VER, MOD</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>RESTYPE, NODE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Configuration File</td>
<td>0220</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Dataset</td>
<td>0030</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Error Log</td>
<td>0250</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Flat Data</td>
<td>0100</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Library/sublibrary</td>
<td>0052</td>
<td>RESTYPE, NODE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLIB, LMTYPE, VER, MOD</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>RESTYPE, NODE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLIB, VER, MOD</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>RESTYPE, NODE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>RESTYPE, NODE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLIB, VER, MOD</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>RESTYPE, NODE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>RESTYPE, NODE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLIB, VER, MOD</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Relational Data</td>
<td>0120</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>RESTYPE, NODE, NAME</td>
<td></td>
</tr>
</tbody>
</table>

Example: Here is an example of a DELETE HISTORY statement to delete resource history records:

    DEL HIST RESTYPE=0030,NODE=SSPNOE,NAME=CHRIS
Use the PRINT HISTORY statement to print a node definition, together with all its attributes and tracking and assignment information. It is equivalent to the GIX Browse Network Configuration function.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT HISTORY</td>
<td>{GROUP=node-group-name}</td>
<td>NDMT, CMEP, CMFP only</td>
</tr>
<tr>
<td></td>
<td>{NODE=node-name}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{NODETYPE=node-type}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{RESTYPE=resource-type}</td>
<td></td>
</tr>
<tr>
<td></td>
<td>{NAME=resource-name}</td>
<td></td>
</tr>
</tbody>
</table>

**GROUP**
Specify the name of the node group (from 1 to 8 alphanumeric characters, the first of which must be alphabetic).

**NODE**
Specify the name of the node (from 1 to 8 alphanumeric characters, the first of which must be alphabetic).

**NODETYPE**
Specify the node type.

**RESTYPE**
Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST.

**NAME**
Specify the name of the resource that you want to print.

**Note:** For panels, programs and members of VSE nodes, NAME= specifies the name of the library to which these resources belong. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**Example:** Use the PRINT HISTORY statement to print the following information for a group of nodes, or a node or node type:

- A specific resource history
- All the resource histories
- All the resource histories of a specific resource type

For example, to print the history of a library resource type named SOFT, related to all nodes belonging to a node group called EASTERN, use the following statement:

```
PRI HIST GROUP=EASTERN,NAME=SOFT,RESTYPE=LIBRARY
```
Chapter 15. Managing Resources in the Repository

The MAINTDAT batch utility is used to work with resources in the resource repository. There are some control statements you can use with all resource types, and some that you use just with specific resource types. MAINTDAT statements belong to three different categories:

- Statements for all resource types
- Statements for selected resource types
- A statement to find out how much space is left in the repository.

This section is divided into these three categories of control statements. Within each section, the statements appear in alphabetical order. First read the following pages to learn about the resource repository, the format of the files you load, and using primary or embedded control statements.

To take advantage of the enhancements for the code page translation, add the language environment run library data set:

CEE.V1R4M/zerodot.SCEERUN

to the JOBLIB DD card of your batch jobs used to perform all the MAINDAT functions.

Loading Resources into the Resource Repository

You have to load resources from external data sets into the resource repository before you can transmit them in a transmission plan. The resource repository is a set of VSAM files which store data resources and software resources.

The resource repository consists of four distinct data sets—the holding file directory and data space, and the NetView DM for MVS library directory and data space. The data spaces contain the actual resources. The directories contain information about how the resource has been stored, and about how it should be processed. Each resource type is held either in the holding file, or in the NetView DM for MVS library file. Table 74 on page 566 shows where each resource type is stored.

You use the following MAINTDAT control statements to load resources into the repository:

- PREPARE statements to load all resource types
- INSERT/REPLACE statements for specific resource types
- FORMAT statements for specific resource types
- LOAD for LU 6.2 resource types

The PREPARE RESOURCE statement requires a user-written exit routine, to convert the resources from your user data file into the resource repository format. You can prepare all resource types with this statement. Refer to the Base Application Programming book for information about how to write an exit routine. The other PREPARE statements, the INSERT/REPLACE statements and the FORMAT statements do not need user exit routines. Use these statements to prepare LU0 resource types.

The LOAD statement does not need a user exit routine, but you can use an optional user exit routine to specify a specific compression/decompression algorithm. You can load all LU 6.2 resource types with this verb.

You must use a separate control statement for each resource, or group of resources, that you load into the repository. The type of data that you are loading into the repository (for example, a job, data set, or print resource), is stored in a physical sequential file. Physical sequential files can store either fixed-length or variable-length records.
There are some resource types that you can load into the repository in variable length format. These are:

- All LU 6.2 resource types
- CLIST
- JOB
- DATA SET
- PRINT

**Note:** During a load process, the resource length will be the same as the data set record if the resource is a **FIXED BLOCK**. If the resource is smaller than the data set record, it will be padded with blanks. This will not occur if the resource is a variable data set.

## Loading Resources from Variable-Length Records

**Physical sequential files** have a parameter called **RECFM**, which specifies whether the record format is variable (RECFM=V, or RECFM=VB), or not (RECFM=F).

**VSAM files** can be organized either as a **key sequenced data set (KSDS)**, or as an **entry sequenced data set (ESDS)**.

If you are using the PREPARE DATASET or PREPARE PRINT statements, to load data set or print data resource types into the repository, you must specify whether or not the file is a variable-length record. You do this using the **VARLEN** operand of the PREPARE statements. You do not need to specify whether the record is of variable length for any of the other PREPARE statements, including PREPARE RESOURCE.

Figure 95 shows the format of a variable-length record in a physical sequential file:

![Figure 95. Format of variable-length records in user files](image)

In a physical sequential file with record format V, each record begins with a 4-byte record descriptor word (RDW), where **LL** is a 2-byte binary number not exceeding 32 768 that represents the length of the record (including the RDW), and **00** is a 2-byte binary field that is always set to zero. If the input file is a variable-length record file, and **VARLEN=YES**, the PREPARE statement reads the records from the input file one after the other, including the RDW, and writes them to the repository without changing them in any way, unless you request compression.

In a physical sequential file with record format VB, the record begins with a block descriptor word (BDW) that describes how many variable-length records (logical records, each headed by its own RDW) the file contains. When the files are loaded into the repository, MAINTDAT unblocks the records and writes each logical record into the repository as if it were reading from a file with record format V.

VSAM files do not have RDWs. When you load a resource from a VSAM file and **VARLEN=YES**, NetView DM for MVS builds a RDW for each record, and writes the record to the repository. The RDW indicates the actual record length.

Once the resource is loaded into the repository, it has the MVS file format shown in Figure 96 on page 263. The records from a physical sequential file and those from a VSAM file look exactly the same when they are stored in the repository. The records stored in the repository are simply a continuous bytes stream, spanning as many repository records as needed. If you requested for the data to be compressed, both the data *and* the RDW are compressed.
MAINTDAT records the fact that the resource is made up of variable-length records, as well as the maximum record length, in the resource repository directory. When the TCP transmits the resource to the node, NetView DM for MVS also sends this directory information to the node.

### Uniform Text Processing

A **uniform text** is a text data object (resource) that contains only characters belonging to an identified graphic character set. The following NetView DM for MVS LU 6.2 resource types can contain a uniform text:

- Flat data
- Dump
- Trace
- Error log
- Procedure

These resources carry the information that tells whether their content is a uniform text. Resources that contain a uniform text must have a defined record structure. Information about the record structure (whether fixed or variable, the logical record length for fixed length record, or the maximum record length for variable length record) are associated with the resource.

This information is stored in the directory entry associated with a resource when loading the resource into the NetView DM for MVS resource repository, either via a LOAD or RETRIEVE function:

**LOAD**  
Information is provided as follows:

- Information about the uniform text and record format is provided *directly* by operands of the LOAD command.
- Logical record length (for fixed records) is provided *indirectly* by the LRECL parameter of the input file.
- The maximum record length (for variable length records) is determined by NetView DM for MVS while loading the data, and corresponds to the longest record read.

Only one code page, identified by the coded character set identifier (CCSID) (specified when customizing NetView DM for MVS using the NDMCOM macro), is used for all the resources LOADed at the host. This CCSID is also stored in the directory record associated with the resource.

All this information is sent to a target node when the resource is transferred to it.

**RETRIEVE**  
The node will send the relevant information when transferring the resource to the host. The code page of each resource being retrieved is identified by one of the CCSIDs chosen from Table 26 on page 264.

**Coded Character Set Identifier (CCSID):** Only one host CCSID is active at NetView DM for MVS at any point in time. The host CCSID is selected from the set of CCSIDs shown in Table 25 on page 264.
When translating a uniform text received from a PWS node, or from another host, the translation process needs both a source CCSID and the target (host) CCSID. The source CCSID has been received from the node with the resource; the target CCSID is the host CCSID selected by NetView DM for MVS customization (from Table 25). Table 26 shows the source CCSIDs that NetView DM for MVS can use for translation:

<table>
<thead>
<tr>
<th>CCSID</th>
<th>CP</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0025</td>
<td>0037</td>
<td>CECP: USA, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand</td>
</tr>
<tr>
<td>0111</td>
<td>00273</td>
<td>CECP: Austria, Germany</td>
</tr>
<tr>
<td>0115</td>
<td>00277</td>
<td>CECP: Denmark, Norway</td>
</tr>
<tr>
<td>0116</td>
<td>00278</td>
<td>CECP: Finland, Sweden</td>
</tr>
<tr>
<td>0118</td>
<td>00280</td>
<td>CECP: Italy</td>
</tr>
<tr>
<td>011D</td>
<td>00285</td>
<td>CECP: United Kingdom</td>
</tr>
<tr>
<td>0129</td>
<td>00297</td>
<td>CECP: France</td>
</tr>
<tr>
<td>01F4</td>
<td>00500</td>
<td>CECP: Belgium, Canada (AS/400), Switzerland, international Latin-1</td>
</tr>
<tr>
<td>011C</td>
<td>00284</td>
<td>CECP: Spain, Latin America</td>
</tr>
<tr>
<td>0367</td>
<td>00871</td>
<td>CECP: Iceland</td>
</tr>
</tbody>
</table>

Table 25. Possible NetView DM for MVS host CCSIDs

You can create SBCS tables to convert a source CCSID to a target CCSID not included in Table 25 and Table 26. You manage these customized SBCS tables by using the ADDTABLE utility at installation time. For more information on ADDTABLE, see the section on using utilities in the Installation and Customization book.

You can also manage double byte character set (DBCS) and mixed uniform text resources supported by DFSMS 1.2 with APAR AW14950 or higher.
Loading a Uniform Text: When loading a uniform text resource into the repository, the record structure of the user input file is handled in the following way:

- If the user input file has a fixed record format, the logical record length of the input file is stored in the holding file directory; the resource is marked as having “fixed record length.”
- If the user input file has variable record format, set the LOAD operand VARLEN to YES. Otherwise, the LOAD function terminates with return code 8 and an error message is issued.

The LOAD processor will store records being read from the input file in the data part of the resource repository, with each record headed by a Record Descriptor Word (RDW). The RDW conforms with the RDW used by MVS for sequential files with variable length record, a 4-byte field:

- The first bytes contain the length $LL$ of the logical record including the 4-byte RDW.
- The length can be from 4 to 32,756, and all bits of the third and fourth byte must be 0.

The resource is marked as having variable length records.

Note

Batch Utilities, GIX, and the D&CC API LOAD function can load any LU 6.2 resource type.

Unloading/Printing a Uniform Text Resource: When unloading a resource that contains a uniform text from the NetView DM for MVS resource repository to a user file, or when printing such a resource, the data is translated from the CCSID associated with the resource to the NetView DM for MVS defined CCSID.

You can also unload double-byte character set (DBCS) uniform text resources. However, no translation is performed and the resources are unloaded as they are.

You cannot unload these resources when they are compressed by a user compression algorithm and the CCSID of the algorithm name identifies a DBCS code page. Table 27 shows a list of the DBCS CCSIDs that are supported.

<table>
<thead>
<tr>
<th>Decimal CCSID</th>
<th>Hex CCSID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0932</td>
<td>03A4</td>
<td>COMBINED_JAPAN_CP1</td>
</tr>
<tr>
<td>0942</td>
<td>03AE</td>
<td>COMBINED_JAPAN_CP2</td>
</tr>
<tr>
<td>0934</td>
<td>63A6</td>
<td>COMBINED_KOREAN_CP1</td>
</tr>
<tr>
<td>0944</td>
<td>03B0</td>
<td>COMBINED_KOREAN_CP2</td>
</tr>
<tr>
<td>0949</td>
<td>03B5</td>
<td>COMBINED_KOREAN_CP3</td>
</tr>
<tr>
<td>0938</td>
<td>03AA</td>
<td>COMBINED_CHINESE_CP1</td>
</tr>
<tr>
<td>0948</td>
<td>03B4</td>
<td>COMBINED_CHINESE_CP2</td>
</tr>
</tbody>
</table>

Table 27. Possible DBCS CCSIDs

Using Primary and Embedded Statements

The FORMAT FILE, FORMAT PRINT, PREPARE DATASET, PREPARE PRINT, and LOAD RESO statements can be either primary or embedded statements. You can mix primary and embedded statements within a single job.

Using embedded statements, you can group several resources together in a single input file, and use a single primary control statement in the MAINTDAT job stream.
For the FORMAT FILE, FORMAT PRINT, PREPARE DATASET, and PREPARE PRINT statements, you insert embedded control statements in the input user data files, preceded by a *period ampersand (.*). You can change these two characters using the CTLID parameter in the NDMBATCH generation macro, as described in the *Installation and Customization* book.

For the LOAD RESO statement, you insert embedded control statements in the input user data files, preceded by `.EMBCNTRL&`. You can change this string using the EMBCNTRL operand in the LOAD RESOURCE statement, as described in “LOAD RESOURCE” on page 288.

The information you specify in an embedded statement applies to all the data that follows, until the next embedded statement, or until the end of the file. Each embedded statement must include all the required operands; no information is carried over from the previous embedded statement or from the primary statement. If the input file contains no embedded statements, the primary statement information applies to all the input data.

**Note:** If you use embedded statements in an input file, do not mix:

- Embedded FORMAT and PREPARE statements
- Embedded FORMAT FILE and FORMAT PRINT statements

You can mix PREPARE DATASET and PREPARE PRINT statements.

For examples of how to use embedded statements, see “PREPARE DATASET/PRINT” on page 296.

**Locking Resources in the Resource Repository**

User programs can use the NetView DM for MVS D&CC API to *lock* resources held in the resource repository. Once a resource is locked, you cannot delete or change the resource using GIX, the MAINTDAT batch utility, or the TCP. Resources can be unlocked using either the D&CC API Unlock_Catalog_Entry function, or the MAINTDAT control statement called ULOCK RESOURCE, described in “ULOCK RESOURCE” on page 311.

Refer to the *Distribution and Change Control Application Programming* book for information about how to lock and unlock resources using the D&CC API.
Unloading Resources from the Repository

You use the following MAINTDAT control statements to unload resources from the resource repository into a user data set:

- CONVERT statements, to unload all resource types
- UNLOAD RESOURCE statement, for LU 6.2 resource types.

The CONVERT RESOURCE statement requires an exit routine written by the user, to convert the resources from the repository format into user data format. You can convert all resource types with this statement.

Refer to the Base Application Programming book for information about how to write an exit routine. The other CONVERT statements do not require user exit routines. Use these statements to convert LU0 resource types.

The UNLOAD RESOURCE does not need a user exit. You can unload all LU 6.2 resource types with this statement. NetView DM for MVS automatically decompresses compressed resources during the Unload and Print Resource functions. Decompression occurs if the resource was compressed with:

- SNA compression (provided by NetView DM for MVS)
- The NVDMMLZW algorithm
- A user compression algorithm that was defined via the NDMCOM customization macro. If the decompression algorithm is not available, the Unload function will unload the compressed resource and issue a warning message; the Print Resource function will fail.

You must use a separate control statement for each resource, or group of resources, that you unload into your user data set. You unload a resource stored in the repository into an output file. The output file can either be a physical sequential file, or a VSAM KSDS or ESDS file. LU 6.2 resources can be converted into a sequential file.

A physical sequential output file can be a member of a partitioned data set, a GDG data set, or a sequential data set of any device type. It can have any record format: fixed, variable or undefined, blocked or unblocked. The maximum record length is 32768 bytes.

If you are unloading a resource stored as a variable-length record, each record is unloaded into the output file with its RDW, if the output file is physical sequential with a RECFM=V or RECFM=VB. If the output file is a VSAM KSDS or ESDS file, the record is unloaded into the output file without a RDW. You cannot unload variable-length records into physical sequential files that do not have a record format of V or VB.

Note: During an unload process, the resource length will be the same as the data set record if the resource is a FIXED BLOCK. If the resource is smaller than the data set record, it will be padded with blanks. This will not occur if the resource is a variable data set.
Using the DISP Operand

The discussion in this section applies to wherever the DISP operand is used, both in this chapter and in Chapter 16, “Preparing and Submitting Transmission Plans” on page 313.

How to Use the DISP Operand: The DISP operand has two sets of values: (new,old,mod); and (keep, delete). The set that NetView DM for MVS uses depends on the following conditions:

- Which control statement is used
- Which combination of values you specify for DISP
- Which resource is involved

The (new,old,mod) set specifies how the resource is prepared in the resource repository and is used by NetView DM for MVS during storage of the resource (see Table 28). The (keep,delete) set specifies the disposition of prepared resources after subsequent processing and is used by NetView DM for MVS during the execution of a Send, Replace or Convert statement (see Table 29).

<table>
<thead>
<tr>
<th>DISP Operand</th>
<th>Function</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>new,old,mod</td>
<td>Receive/Retrieve</td>
<td>Data Set, Dump, Error log, Library, Member, Panel, Program</td>
</tr>
<tr>
<td></td>
<td>Prepare</td>
<td>Dataset/Print</td>
</tr>
<tr>
<td></td>
<td>Prepare Resource</td>
<td>LU0 resources and Microcode</td>
</tr>
<tr>
<td></td>
<td>Retrieve Resource</td>
<td>LU0 resources</td>
</tr>
<tr>
<td></td>
<td>Prepare</td>
<td>Clist/Job/Program</td>
</tr>
<tr>
<td></td>
<td>Load Resource</td>
<td>Flat data, Relational data, Config, Dump, Errorlog, Trace</td>
</tr>
</tbody>
</table>

Send Resource (1) All resources (1) Used when the resource is stored at the target repository.

Table 28. (new,old,mod) values in relation to functions and resources

<table>
<thead>
<tr>
<th>DISP Operand</th>
<th>Function</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>keep,delete</td>
<td>Receive/Retrieve</td>
<td>Data set, Dump</td>
</tr>
<tr>
<td></td>
<td>Retrieve Resource</td>
<td>LU0 resources</td>
</tr>
<tr>
<td></td>
<td>Prepare</td>
<td>Dataset/Print</td>
</tr>
</tbody>
</table>

Send Resource (1) LU0 resources (1) Used at target repository.

Table 29. (keep,delete) values in relation to functions and resources

Explanation of Operand Values

NEW The resource does not already exist in the resource repository. This is usually the default value.

OLD An existing resource is to be replaced. If the resource does not exist in the resource repository, it is prepared as in the case of DISP=(NEW).

MOD An existing resource is to be modified. If the resource does not currently exist, it is prepared as in the case of DISP=(NEW). If the resource currently exists, the input data is appended to it. The attributes of the appended resource replace the existing ones. You can modify the original attributes of a resource by appending an empty one.
**KEEP**  The prepared resource is to be retained in the resource repository after a Send, Replace or Convert operation. This is usually the default value.

**DELETE**  The prepared resource is deleted by a subsequent Send, Replace or Convert. If the operation is not completed successfully, the resource is not deleted. If the transmission phase addresses a single node, the resource is deleted immediately after transmission; if the phase addresses a group of nodes, the resource is deleted at the end of the phase.

### MAINTDAT Control Statements for All Resource Types

The syntax and operands of the statements that you can use with any resource type are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

#### CONVERT RESOURCE

Use the CONVERT RESOURCE control statement to unload any type of resource from the resource repository. Unloading resources is the conversion of a resource from the resource repository into a file in the user data set. The CONVERT RESOURCE function requires an exit routine written by you, the user.

Refer to the *Base Application Programming* book for information about how to write an exit routine.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONvert RESOurce</td>
<td>USEREXIT= user-exit-name</td>
<td>All node types</td>
</tr>
<tr>
<td></td>
<td>[,TYPE= resource-type-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,EXT=0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>[,FCTLNNAME= resource-queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,KEYNUMB= key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,LIBNAME= library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,LMTYPE= library/member-type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,MEMBER= member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,MOD= 0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,NAME= resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,ORIGIN= origin-of-resource]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,SUBCAT= subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,SUBLNAME= sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,VER= 0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,USERINFO= user-information]</td>
<td></td>
</tr>
</tbody>
</table>

See Table 30 on page 272 for information about which operands to use with each resource type.

**USEREXIT**  Specify the name of the user-written routine that MAINTDAT must call to provide the data to the exit routine. You can specify up to 8 characters. The first character must be alphabetic.

This operand is *always* mandatory. You must specify the name of the user exit routine that you are using.

The remaining operands identify the resource. Each resource type has a different set of operands that identify the resource, for example, microcode requires the **TYPE** and **NAME** operands. These operands are referred to as the resource *identifiers*. 

---

Chapter 15. Managing Resources in the Repository  269
All of these operands are optional within the control statement, as you can specify the resource identifiers in one of two ways:

- You can specify the resource identifiers in the user exit routine. In this case, you do not have to specify them in the control statement. The advantage of this method is that it allows you to process more than one request from a single control statement, as the user exit routine can be called cyclically from MAINTDAT.
- You can specify the resource identifiers in the CONVERT RESOURCE control statement.

Whichever method you use, Table 30 on page 272 shows you which resource identifiers you have to specify for each resource type.

**Note:** If you specify the resource identifier values in the user exit routine and in the control statement, the values in the user exit routine override those in the control statement. You cannot specify the resource identifier partially in the user exit routine, and partially in the control statement.

**TYPE**
Specify the resource type code, as described in “NetView DM for MVS Resource Types” on page 566.

**EXT**
(DSCB resource types only). Specify either the extent (0 or 1), or the key number (from 1 to 8) (described below), for the DSCB resource type. You cannot enter both the extent and the key number. If you do not specify either an extent or a key, the default is EXT=0.

**FCTLNAME**
Specify the name by which the resource is queued in the repository. You can specify up to 8 characters. The first character must be alphabetic. This operand is mandatory for some resource types. Table 30 on page 272 shows which resource types require this operand.

For example:

- If you want to convert all the data sets with FCTLNAME=TABLES, then you specify this statement:
  
  ```
  CONVERT RESOURCE TYPE=/zerodot/zerodot3/zerodot,FCTLNAME=TABLES,USEREXIT=SCAFDATA
  ```

- If you have data sets retrieved from node AR01 and node WS01 (different ORIGINs), but the resources have the same FCTLNAME, yet you only want to convert the resources retrieved from WS01, then you must specify this statement:

  ```
  CONVERT RESOURCE TYPE=/zerodot/zerodot3/zerodot,FCTLNAME=TABLES,ORIGIN=WS/zerodot1,USEREXIT=SCAFDATA
  ```

**KEYNUMB**
(DSCB resource types only). Specify the key number (from 1 to 8), only if you have not already entered an extent. If you do not specify either an extent or a key, the default is EXT=0, and no KEYNUMB is applicable to the resource type.

**LIBNAME**
Specify a library name.

**LMTYPE**
Specify the library type:

- C Core image library
- P Procedure library
- R Relocatable library
- S Source library

For member resource types, you can specify up to eight alphanumeric characters.

**Note:** If you specify a library type of S, you must also specify a sublibrary name in the SUBLNAME operand.

**MEMBER**
Specify a member name.
MOD Specify the modification level. The default is 0, when this operand is applicable.

NAME Specify the resource name. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

ORIGIN Specify the name of the resource origin. You can specify up to 8 characters. The first character must be alphabetic. If the resource was retrieved from a node, this is the node name. If the resource was prepared at the host, the origin is DSX. This operand is mandatory for some resource types. Table 30 on page 272 shows which resource types require this operand.

SUBCAT Specify the subcategory (a value from 1 to 16). If you do not enter a value, NetView DM for MVS assumes a blank.

SUBLNAME Specify the sublibrary name. The default is to prepare the entire library.

Note: If the library type is S (LTYPE=S), you must specify a sublibrary name.

USERINFO Specify a string of 150 bytes that will be passed to the user module.

VER Specify the version number. The default is 0, when this operand is applicable.
**Usage:** Table 30 shows you which of the operands that identify the resource are mandatory, and which are optional, for each resource type. The `USEREXIT` operand is not included in this table, as it is always mandatory.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td>0158</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>TYPE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>TYPE, FCTLNAME</td>
<td>NAME, ORIGIN</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>TYPE, FCTLNAME</td>
<td>NAME, ORIGIN</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>TYPE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>TYPE, LIBNAME, LMTYPE</td>
<td>VER, MOD, SUBLNAME</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>TYPE, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>TYPE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME</td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>TYPE, FCTLNAME</td>
<td>NAME</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>TYPE, NAME</td>
<td>VER, MOD</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>TYPE, FCTLNAME</td>
<td>NAME</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>TYPE, FCTLNAME</td>
<td>NAME, ORIGIN</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
</tbody>
</table>

**Example:** This example is unloading microcode from a user data set into the repository:

```
CONVERT RESOURCE TYPE=MICROCODE,
    NAME=MCODE.3174.NA.FUNCTEC.AO500.CONTROL,
    USEREXIT=SCAFMICL
```
ERASE RESOURCE

Use the ERASE RESOURCE statement to erase a resource, or queue of resources, from the resource repository.

You can erase a single resource providing no other task is using the resource. You cannot erase a resource that has been locked in the Resource Repository. If the control statement addresses a resource that is locked, NetView DM for MVS issues an error message.

You can erase an entire queue of LU0 resources if you specify the TYPE and FCTLNAME operands. If one or more resources in the queue are currently in use, NetView DM for MVS only erases the resources that are not in use, and issues one or more error messages. NetView DM for MVS only erases those resources in the queue that are not locked. If locked resources are found, NetView DM for MVS issues a message telling you that some resources were not erased because they are locked.

Syntax

Note: These restrictions apply:

- For LU 6.2 resources, you must specify TYPE and NAME.
- For LU0 resources, you must specify TYPE and as many of the other operands to properly identify the resource.

FCTLNAME Specify the name by which the resource is queued in the repository. You can specify up to eight characters. The first character must be alphabetic. This operand is mandatory for some resource types. Table 30 on page 272 shows which operands are required for each resource type. If you only specify a FCTLNAME, NetView DM for MVS erases all of the resources queued under that name. If you also specify the ORIGIN operand, NetView DM for MVS also erases all of the resources in the queue with that origin.

TYPE Specify the resource type code, which is shown in “NetView DM for MVS Resource Types” on page 566.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERASE RESOURCE</td>
<td>[FCTLNAME=queue-name] ,TYPE=resource-type [{EXT=0</td>
<td>1}KEYNUMB= key-number] [LIBNAME=library-name] [LMTYPE={library\member-type}] [MEMBER=member-name] [MOD=0</td>
</tr>
</tbody>
</table>

See Table 31 on page 276 for information about which operands to use with each resource type.
ERASE RESOURCE

EXT (DSCB resource types only). Specify either the extent (0 or 1), or the key number (from 1 to 8) (described below), for the DSCB resource type. You cannot enter both the extent and the key number. If you do not specify either an extent or a key, the default is EXT=0.

KEYNUMB (DSCB resource types only). Specify either the extent (0 or 1), or the key number (from 1 to 8). If you do not specify either an extent or a key, the default is EXT=0.

LIBNAME Specify a library name.

LMTYPE Specify the library type:

- C Core image library
- P Procedure library
- R Relocatable library
- S Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLNAME operand.

MEMBER Specify the member name.

MOD VER Specify the version number and modification level of the resource.

NAME Specify the resource name.

You can use one or more wildcard characters (*) in the resource name in the following ways:

- The wildcard can represent any token except the first.
- The wildcard replaces a token completely.

For LU0 resources, if you specify a NAME, you must also specify a value for the FCTLNAME, or ORIGIN, or both operands. If you do not specify a NAME, NetView DM for MVS erases all of the resources described by the FCTLNAME, or by the FCTLNAME or ORIGIN operands.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

ORIGIN Specify the name of the resource origin. You can specify up to eight characters. The first character must be alphabetic. If the resource was retrieved from a node, this operand is the node name. If the resource was prepared at the host, the origin is DSX. This operand is mandatory for some resource types. Table 31 on page 276 shows which operands are required for each resource type.

STATUS Resources are not erased if one or more of the resources in the queue have a different status from the value you specify here (or the default value, if you do not enter one). You can specify:

- STATUS=COMP This means that the resource is erased only if it is complete, and you have not interrupted a write process or a write process that is pending.
- STATUS=ANY This means the resource is erased whether it is complete or incomplete, even if you interrupt a write process.
STATUS=APP  This applies to resources which were not completed because of an interrupted writing process. If you specify STATUS=APP, the erase is performed only on the appended part of the resource.

SUBCAT  Specify the subcategory (from 1 to 16). If you do not enter a value, a blank is assumed.

SUBLNAME  Specify the sublibrary name. The default is to prepare the entire library.

Note: If the library type is S (LMTYPE=S), you must specify a sublibrary name.

MATCHIND  Specify the list of matching indicators that MAINTDAT uses to identify the resource to be processed. The MATCHIND values can be:

- n  The qualifier ordinal number referred to by the matching indicator.
- L  Set low value.
- H  Set high value.
- I  Ignore value. This means that only one resource name in the holding file matches the other token specifications, but this token can have any value.

If you do not specify a MATCHIND value, the name is considered to be fully qualified and the related resource is erased. If you do specify this value, the resource that satisfies the matching rules is erased.

MULTIERA  Specify that you can erase all the LU62 resources matching the selection criteria you use in the NAME operand. The MULTIERA values can be:

- NO  Erase a single resource from the resource repository, the resource you specify in the NAME operand. This is the default value.
- YES Erase all the resources that match the selection criteria you specify in the NAME operand from the resource repository.

Do not specify the MATCHIND operand if MULTIERA=YES.
**Usage:** Table 31 shows which operands are mandatory, and which are optional, for each resource type.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td>0158</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>TYPE, NAME</td>
<td>STATUS, VER, MOD</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS, ORIGIN, NAME</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS, ORIGIN, NAME</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>TYPE, NAME</td>
<td>STATUS, VER, MOD</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>TYPE, LIBNAME</td>
<td>SUBLNAME, LMTYPE, STATUS, VER, MOD</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, STATUS, VER, MOD</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>TYPE, NAME</td>
<td>STATUS, VER, MOD</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, STATUS, VER, MOD</td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS, NAME</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>TYPE, NAME</td>
<td>STATUS, VER, MOD</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, STATUS, VER, MOD</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS, NAME</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>TYPE, FCTLNAME</td>
<td>STATUS, ORIGIN, NAME</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>TYPE, NAME</td>
<td>STATUS, MATCHIND</td>
</tr>
</tbody>
</table>

**Example:** This example is to erase a flat data resource type from the resource repository:

```plaintext
ERA RESO NAME=FLA.STATE.*.NDM,
   TYPE=0100,
   MATCHIND=(3,1)
```

The MATCHIND operand is specifying that the third qualifier of the resource name, which is partially defined with an asterisk (*), can have any value, but the result must be unique.
PREPARE RESOURCE

Use the PREPARE RESOURCE control statement to copy an LU0 or LU 6.2 resource from a user file into the resource repository, using a user-written program. For the protocol interface, refer to the Base Application Programming book.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREpare RESOurce</td>
<td>USEREXIT=user-exit-name</td>
<td>All node types</td>
</tr>
<tr>
<td></td>
<td>[,TYPE=resource-type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,DISP={NEW</td>
<td>OLD</td>
</tr>
<tr>
<td></td>
<td>[,EXT=0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>[,FCTLNAME=resource-queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,KEYNUMB=key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,LIBNAME=library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,LMTYPE=library</td>
<td>member-type]</td>
</tr>
<tr>
<td></td>
<td>[,MEMBER=member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,NAME=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,SUBLNAME=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,USERINFO=user-info]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

See Table 32 on page 279 for information about which operands to use with each resource type.

USEREXIT

Specify the name of the user-written program that MAINTDAT must call to prepare the data from the user. You can specify up to eight characters. The first character must be alphabetic. You must enter a value for this operand.

This operand is mandatory. You must specify the name of the user exit routine that you are using.

The remaining operands identify the resource. Each resource type has a different set of operands that identify the resource, for example, microcode requires the TYPE and NAME operands. These operands are referred to as the resource identifiers. All of these operands are optional within the control statement, as you can specify the resource identifiers in one of two ways:

- You can specify the resource identifiers in the user exit routine. In this case, you do not have to specify them in the control statement. The advantage of this method is that it allows you to process more than one request from a single control statement, as the user exit routine can be called cyclically from MAINTDAT.
- You can specify the resource identifiers in the PREPARE RESOURCE control statement.

Whichever method you use, Table 32 on page 279 shows you which resource identifiers you have to specify for each resource type.

**Note:** If you specify the resource identifier values in the user exit routine and in the control statement, the values in the user exit routine override those in the control statement. You cannot specify the resource identifier partially in the user exit routine, and partially in the control statement.

**TYPE** Specify the resource type code, as described in “NetView DM for MVS Resource Types” on page 566.
PREPARE RESOURCE

DISP  Specify how the resource is prepared in the resource repository. The values you can specify are:

- **NEW** This means that the resource does not currently exist in the resource repository. If the resource does not currently exist in the resource repository, it is prepared as in the case of DISP=(NEW).
- **OLD** This means that you are replacing an existing resource. If the resource does not currently exist in the resource repository, it is prepared as in the case of DISP=(NEW). If the resource currently exists, the input data is appended to it; the attributes of the appended resource replace the existing attributes. You can modify the original attributes of a resource by appending an empty one.
- **MOD** This means that you are modifying an existing resource. If the resource does not currently exist in the resource repository, it is prepared as in the case of DISP=(NEW). If the resource currently exists, the input data is appended to it; the attributes of the appended resource replace the existing attributes. You can modify the original attributes of a resource by appending an empty one.

EXT  (DSCB resource types only). Specify the extent (0 or 1). You cannot specify an extent here and a key number in the KEYNUMB field. If you do not specify an extent, or a key number, the default is EXT=0.

FCTLNAME  Specify the name by which the resource is queued in the repository. You can specify up to eight characters. The first character must be alphabetic. This operand is mandatory for some resource types. Table 32 on page 279 shows which operands are required for each resource type.

KEYNUMB  (DSCB resource types only). Specify the key number (from 1 to 8). You cannot specify an extent in the EXT field and a key number here. If you do not specify an extent or a key number, the default is EXT=0, and no KEYNUMB is applicable to the resource type.

LIBNAME  Specify a library name.

LMTYPE  Specify the library type:

- **C** Core image library
- **P** Procedure library
- **R** Relocatable library
- **S** Source library

For member resource types, you can specify up to eight alphanumeric characters.

**Note:** If you specify a library type of S, you must also specify a sublibrary name in the SUBLNAME operand.

MEMBER  Specify the member name.

MOD  Specify the modification level. The default is 0, when applicable for the resource type.

NAME  Specify the resource name. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

SUBCAT  Specify the subcategory (from 1 to 16). If you do not enter a value, a blank is assumed.

SUBLNAME  Specify the sublibrary name. The default is to prepare the entire library.

**Note:** If the library type is S (LMTYPE=S), you must specify a sublibrary name.

USERINFO  Specify a string of 150 bytes that will be passed to the user module.

VER  Specify the version number. The default is 0, when applicable for the resource type.
Usage: Table 32 shows you which of the operands that identify the resource are mandatory, and which are optional, for each resource type. The USEREXIT operand is not included in this table, as it is always mandatory.

Table 32 shows which operands are mandatory, and which are optional, for each resource type.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td>0158</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>TYPE, NAME</td>
<td>VER, MOD, DISP</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>TYPE, FCTLNAME</td>
<td>NAME, DISP</td>
</tr>
<tr>
<td>Dump (CMEP)</td>
<td>0230</td>
<td>TYPE, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>TYPE, NAME, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>TYPE, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>TYPE, NAME</td>
<td>VER, MOD, DISP</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>TYPE, LIBNAME, LMTYPE</td>
<td>VER, MOD, SUBLNAME, DISP</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME, DISP</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>TYPE, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>TYPE, NAME</td>
<td>VER, MOD, DISP</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME, DISP</td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>TYPE, NAME, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>TYPE, NAME</td>
<td>VER, MOD, DISP</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>VER, MOD, SUBLNAME, DISP</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>TYPE, NAME, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>TYPE, NAME, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>TYPE, NAME</td>
<td></td>
</tr>
</tbody>
</table>

Example: This PREPARE RESOURCE statement is copying microcode from a user file into the repository by calling a user-written program:

```
PREPARE RESOURCE TYPE=MICROCODE,
    NAME=MCOO.3174.NA.ITIBMROM.T0831741.CFG.CFG17.2,
    USEREXIT=SCAFMICL
```
PRINT DIRECTORY

Use the PRINT DIRECTORY control statement to print all the directories of the specified resource type.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT DIREctory</td>
<td>TYPE=resource-type</td>
<td>All node types</td>
</tr>
</tbody>
</table>

**TYPE** Specify the resource type code, as described in “NetView DM for MVS Resource Types” on page 566. You must enter a value for this operand.

**Example:** This example shows how to print the directories for all flat data resource types:

PRI DIRE TYPE=0100
PRINT RESOURCE

Use the PRINT RESOURCE statement to print resources stored in the repository. Resources that are stored in a compressed form are decompressed when they are printed.

When printing a resource, NetView DM for MVS automatically decompresses the resource if the resource was compressed by one of these methods:

- SNA compression
- NVDMLZw algorithm
- User compression algorithm that was defined with the NDMCOM macro.

If the resource contains uniform text, NetView DM for MVS performs data translation on the resource. Refer to “Uniform Text Processing” on page 263

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT RESOURCE</td>
<td>TYPE=resource-type</td>
<td>All node types</td>
</tr>
<tr>
<td></td>
<td>[.FCTLNAME=queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.ORIGIN=origin]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.CONV=EBCDIC</td>
<td>HEX]</td>
</tr>
<tr>
<td></td>
<td>[.EXT=0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>[.KEYNUMB=key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.LIBNAME=library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.LMTYPE={library</td>
<td>member-type}]</td>
</tr>
<tr>
<td></td>
<td>[.MEMBER=member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NAME=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBLNAME=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L</td>
<td>H</td>
</tr>
</tbody>
</table>

**TYPE**
Specify the resource type code, as described in “NetView DM for MVS Resource Types” on page 566. You must enter a value for this operand.

**FCTLNAME**
Specify the name by which the resource is queued in the repository. You can specify up to eight alphanumeric characters. The first character must be alphabetic.

**ORIGIN**
Specify the name of the resource origin. If the resource was retrieved from a node, this is the node name. If the resource was prepared at the host, the origin is DSX.

**CONV**
Specify whether the printout should be in:

- Character (EBCDIC) format
- Dump (HEX) format (for LU0 resources only).

**EXT**
(DSCB resource types only). Specify either the extent (0 or 1), or the key number (from 1 to 8) (described below), for the DSCB resource type. You cannot enter both the extent and the key number. If you do not specify either an extent or a key, the default is EXT=0.

**KEYNUMB**
(DSCB only). Specify either the extent (0 or 1), or the key number (from 1 to 8). You cannot specify an extent and a key number. If you do not specify an extent or a key number, the default is EXT=0, and no KEYNUMB is applicable to this resource type.
PRINT RESOURCE

LIBNAME Specify a library name.
LMTYPE Specify the library type:
  C  Core image library
  P  Procedure library
  R  Relocatable library
  S  Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the
SUBLNAME operand.

MEMBER Specify the member name.
MOD Specify the modification level. The default is zero.
NAME Specify the resource name. Appendix C, “Resource Naming Conventions” on page 569
describes the syntax of resource names.
SUBCAT Specify the subcategory (from 1 to 16). The default is a blank.
SUBLNAME Specify the sublibrary name. The default is to prepare the entire library.

Note: If the library type is S (LMTYPE=S), you must specify a sublibrary name.
VER Specify the version number. The default is zero.
MATCHIND Specify the list of matching indicators that MAINTDAT uses to identify the resource to be
processed. The MATCHIND values can be:

  n  The qualifier ordinal number referred to by the matching indicator.
  L  Set low value.
  H  Set high value.
  I  Ignore value. This means that only one resource name in the holding file matches
      other token specifications, but this token can have any value.

If you do not specify a MATCHIND value, the name is considered to be fully qualified
(that is, the match is unique), and the related resource is printed. If this parameter is
specified, the resource that satisfies the matching rules is printed.

Usage: Table 33 shows which operands are mandatory, and which are optional, for each resource
type.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment</td>
<td>0158</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>structure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>TYPE, NAME</td>
<td>CONV, VER, MOD</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>TYPE, FCTLNAME, ORIGIN, NAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
</tbody>
</table>
Table 33 (Page 2 of 2). Applicability of PRINT RESOURCE operands

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>TYPE, FCTLNAME, ORIGIN, NAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>TYPE, NAME</td>
<td>CONV, VER, MOD</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>TYPE, LIBNAME</td>
<td>SUBLNAME, LMTYPE, CONV, VER, MOD</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, CONV, VER, MOD</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>TYPE, FCTLNAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>TYPE, NAME</td>
<td>CONV, VER, MOD</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, CONV, VER, MOD</td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>TYPE, FCTLNAME, NAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>TYPE, NAME</td>
<td>CONV, VER, MOD</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>TYPE, LIBNAME, LMTYPE, MEMBER</td>
<td>SUBLNAME, CONV, VER, MOD</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>TYPE, FCTLNAME, NAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>TYPE, FCTLNAME, ORIGIN, NAME</td>
<td>CONV</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>TYPE, NAME</td>
<td>CONV, MATCHIND</td>
</tr>
</tbody>
</table>

Example: Here is an example of a PRINT RESOURCE statement to print a flat data resource type:

```
PRI RESO NAME=FLA.STATE.*.NDM,
   TYPE=0100,
   MATCHIND=(3,1)
```

The MATCHIND operand is specifying that the third qualifier of the resource name, which is partially defined with an asterisk (*), can have any value.
MAINTDAT Control Statements for Specific Resources

The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

CONVERT DATASET/DUMP

Use the CONVERT DATASET statement to unload a data set from the resource repository into a file in your user data set. Similarly, you use the CONVERT DUMP statement to unload a dump from the repository into a user data set.

**Note:** You can also use these statements to convert data sets and dumps using the FORMAT batch utility, for compatibility with previous releases of NetView DM for MVS.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONvert DATAset</td>
<td>FCTLNAME=queue-name</td>
<td></td>
</tr>
<tr>
<td>CONvert DUMP</td>
<td>[ORIGIN=origin]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[{ID=(resource-name,...)</td>
<td>GROUPID=group-name}]</td>
</tr>
<tr>
<td></td>
<td>[DESTYPE=PS</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>[OUTDD=SYSOUT</td>
<td>ddname]</td>
</tr>
<tr>
<td></td>
<td>[OPENOUTD={NO</td>
<td>YES}]</td>
</tr>
</tbody>
</table>

You must specify a **FCTLNAME**, but **ORIGIN** is optional. You can specify either **ID** or **GROUPID**, but not both. If you specify a **GROUPID**, you must also specify an **ORIGIN**.

<table>
<thead>
<tr>
<th><strong>FCTLNAME</strong></th>
<th>Specify a file control name, which identifies a queue in the resource repository. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. If you specify a queue name without also specifying an origin, NetView DM for MVS will convert all of the data sets under that file control name.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ORIGIN</strong></td>
<td>Specify the name of the node that the resource was retrieved from. If the resource was prepared at the host, the origin is DSX. If you specify a file control name and an origin without also specifying an <strong>ID</strong> or <strong>GROUPID</strong>, NetView DM for MVS will convert all of the data sets with that origin in the specified queue-name.</td>
</tr>
<tr>
<td><strong>ID</strong></td>
<td>Specify one or more resource names from the resource repository. Appendix C, “Resource Naming Conventions” on page 569 describes resource names. You can specify multiple resource names, to convert more than one resource at a time. You must separate multiple entries by commas, and enclose them in parentheses.</td>
</tr>
<tr>
<td><strong>GROUPID</strong></td>
<td>Specify the name of a resource group that is defined in the DRD.</td>
</tr>
</tbody>
</table>

**Note:** You must specify a **FCTLNAME**, but **ORIGIN** is optional. You can specify either **ID** or **GROUPID**, but not both. If you specify a **GROUPID**, you must also specify an **ORIGIN**.
You use the following operands to specify the output data set at the host:

**DESTYPE**  Specify the type of output file. This can be:

- **PS** (physical sequential)
- **ESDS** (VSAM entry-sequenced)
- **KSDS** (VSAM key-sequenced)

A physical sequential output file can be a member of a partitioned data set, a GDG data set, or a sequential data set of any device type.

**OUTDD**  Specify the name of the output user data set that MAINTDAT uses to store the converted data. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

The output user data set is specified by a DD statement in the JCL file. The default is SYSOUT. If the conversion cannot be started (for example, if no resource is found in the resource repository and **OPENOUTD=NO**), NetView DM for MVS does not open and close the output data set.

**OPENOUTD**  Specify whether to open the output data set if the CONVERT statement is unsuccessful.

If the resource repository does not include any resource with the name you specify in the ID operand, and you specify NO, NetView DM for MVS does not open the data set. If you specify YES, it opens the data set.

**Example:**  Here is an example of a CONVERT DATASET statement to unload a physical sequential data set from the resource repository into an output user data set called KATIA:

```
CONVERT DATASET ID=DPPXDAT2,FCTLNAME=QUEUE1,
ORIGIN=DSX,DESTYPE=PS,OUTDD=KATIA
```
INSERT/REPLACE CLIST

Use the INSERT CLIST statement to load a new CLIST resource type into the resource repository. You can use this statement with all node types, apart from VSE nodes. You should use the PREPARE CLIST statement for these node types, and also for CLISTS that are over 47 bytes long, to avoid record truncation.

Use the REPLACE CLIST to load a CLIST that already exists in the resource repository.

CLISTs are created from one or more literals that are part of the control statement. If the literal contains a single quote ("), you must enter it twice (""). You must insert a comma between every CMD='data' statement. If you do not enter a comma, no further CMD= statements will be processed.

SSP CLISTs can have a record length from 40 to 120 bytes. Records larger than 120 bytes are truncated when received by the node. SSP CLISTs inserted by MAINTDAT can be used only for the INITIATE FUNCTION request of transmission plan preparation. You cannot just send them to the node.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSert CLIST</td>
<td>ID=(resource-name[,ver,mod])</td>
<td></td>
</tr>
<tr>
<td>REPlace CLIST</td>
<td>,CMD='data'[,CMD='data',CMD='data ',...,]</td>
<td>{DPPX</td>
</tr>
</tbody>
</table>

ID Specify the name, version, and modification level of the CLIST to be inserted into the library. You can only specify one ID in each statement. “Naming Resources” on page 243 describes the syntax of resource names.

CMD Specify a literal value, up to a maximum of 256 bytes long. Each literal becomes a single record in the CLIST in the order of entry. No checks are made for validity. If larger CLISTS are required they should be built at the node, and retrieved using the MAINTDAT PREPARE CLIST statement.

TYPE Specify the type of node for which the CLIST was written. The CLIST is stored in such a way that it is available for transmission to the specified node type only.
INSERT/REPLACE JOB

Use the INSERT JOB statement to load a new job resource type into the resource repository. You can only use this statement with VSE node types. Use the REPLACE JOB statement to load a job that already exists in the resource repository.

Jobs are created from one or more literals which are part of the control statement.

Note: If the literal contains a single quote ("), you must enter it twice ("').

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSERT JOB</td>
<td>ID=(name[,ver,mod]), JSC='data'[,'data',...]</td>
<td></td>
</tr>
<tr>
<td>REPLACE JOB</td>
<td>JSC='data'[,'data',...]</td>
<td></td>
</tr>
</tbody>
</table>

ID Specify the name, version, and modification level of the job to be inserted into the library. You can only specify one ID in each statement. ID is a single unqualified name.

JSC Specify a literal value to a maximum of 80 bytes long. Each literal becomes a single record in the job in the order of entry. No checks are made for validity.
LOAD RESOURCE

Use the LOAD RESOURCE statement to load LU 6.2 resource types directly into the resource repository. You do not need to write a user exit. The resource types that you can load directly are:

- AS/400 data object containment structure
- AS/400 object
- Configuration file
- Flat data
- Microcode (and Microcode customization)
- Procedure
- Relational data
- Software
- Dump
- Error log
- Trace

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOAd RESOurce</td>
<td>NAME=resource-name, .RESTYPE=resource-type, .INDD=ddname,</td>
<td>CMFP, CMEP,</td>
</tr>
<tr>
<td></td>
<td>[.MATCHATT=((n,M</td>
<td>N),...)] [,PRONCSIZE=nnnnnn], [.GENERATT=((n,G</td>
</tr>
<tr>
<td></td>
<td>[.TYPEAT=((n,UNSPECIFIED</td>
<td>ORDEREDC</td>
</tr>
<tr>
<td></td>
<td>SUBTREE</td>
<td>SYSTYPE</td>
</tr>
<tr>
<td></td>
<td>[.OBJECTSZ=resource-size], .TOCOMPRS=[NO][YES], [.FSPARMS=X'nn'],</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.USERPARM='user-parameter'], [.CLCODES=(X'nn';X'nn')], [.FSCOMPTQ=X'nn']</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.USERCOMP=compression-algorithm-name], [.DISP={NEW</td>
<td>OLD</td>
</tr>
<tr>
<td></td>
<td>[.VARLEN={NO</td>
<td>YES}], [.EMBCNTRL=(NO)\YES</td>
</tr>
</tbody>
</table>

NAME EMBCNTRL

Specify the resource name. Do not specify this parameter if you specify EMBCNTRL=YES. The embedded control statement provides the resource name. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

RESTYPE

Specify the resource type code. This operand is mandatory, except that you can omit it for microcode and microcode customization resources, if you specify the first token of the resource name as either MCODE or MCUST. See “NetView DM for MVS Resource Types” on page 566.
INDD Specify the DD name of the file containing the data. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. The input files must be sequential files (blocked or unblocked). You must specify this operand.

MATCHATT Specify the list of matching attributes that define, in the global name, how the specified token can be used during token value generation. The qualifier number referred to by the matching attribute is specified by n. The values you can enter are:

M Must-match indicator. It can be used only for those tokens that allow it.
N Need-not-match indicator. This is the default value, but if you specify TYPEAT=(n,UNSPECIFIED), then a default of M is assumed for the token at position n.

PROCSIZE Specify the estimated size in KB of the free DASD space needed on the target installation drive to install the resource. The target uses this value to check the available DASD space.

Enter an integer from 1 to 999 999. You can only specify this operand for those resource types that support the Install function.

Note: This parameter is supported by NetView DM/2 Version 2 Release 1 target workstations only. NetView DM/6000 and NetView DM for NetWare products check the disk space automatically. Specify a processing size only if the resource is to be installed on a NetView DM/2 Version 2 Release 1 workstation.

GENERATT Define whether or not the target server can generate appropriate token values. You define this for each token in the global name. The qualifier number referred to by this attribute is specified by n. The values you can enter are:

G The target server can generate token values.
N The target server cannot generate token values. This is the default value.

TYPEAT Define the type of the attribute for each token. The qualifier number refers to the token's position in the global name. For example, the fifth token in the name would have 5 as its value for n. The values can be:

UNSPECIFIED This means that values assigned to the token cannot be assumed to have any numeric or alphabetic order. This implies that NetView DM for MVS cannot perform any partial matching operations on these tokens.
ORDEREDC This means that the related token is such that an alphanumeric order is established for the values. This implies that this token is eligible for a search via partial matching. This is the default value.
ORDEREDN This means that the related token is such that a decimal numeric order is established for the values. This implies that this token is eligible for a search via partial matching.
ORDEREDD This means that the related token is such that a date format order is established for the values. This implies that this token is eligible for a search via partial matching.
ORDEREDT This type means that the related token is such that a time format order is established for the values. This implies that this token is eligible for a search via partial matching.

The following tokens have attributes that are defined by SNA. These tokens are checked for compatibility with the architecture.
LOAD RESOURCE

SUBTREE  This type means that this token indicates the beginning of a subtree structure in the global name.

SYSTYPE  This type means that the token contains a value corresponding to a system type.

LUNAME  This type means that the token specifies the destination logical unit name (LUNAME).

NETID  This type means that the token specifies the destination network ID (NETID).

G00V00  This type means that the token contains a value corresponding to an MVS generic data set member.

CLCODES  These are the level 3 and level 4 SNA/File Services data object class codes. Level 1 and level 2 codes are derived from the resource type code. If you specify this operand, you must specify both values. The values you can specify are defined by SNA/File Services, and are listed in “SNA/File Services Class Codes” on page 584.

If the resource to be loaded is a NetView DM for MVS/2 resource containing Extended Attributes, then the level 3 and level 4 class code values must be set to E4.

COMPRSD  Specify whether or not the input data stream is already compressed.

If you specify COMPRSD=YES, you must also define a value that is greater than 0 for the OBJECTSZ operand. If you do not define OBJECTSZ, the function is stopped.

If you specify COMPRSD=NO, do not specify OBJECTSZ.

OBJECTSZ  Specify the size of the resource when decompressed. You must enter a value for this operand if the resource to be loaded is compressed (if COMPRSD=Yes). If the resource is not compressed, do not enter a value.

TOCOMPRS  Specify whether you want the input data stream to be compressed when it is loaded into the repository. Do not specify YES for this option, if the input data stream is already compressed (in other words, if COMPRSD=YES).

If you specify here that you want the data stream compressed, you must also specify the compression algorithm to be used, with either the FSCOMPTQ or USERCOMP operand. If you do not enter any of these operands, FSCOMPTQ defaults to X'01'.

FSCOMPTQ  Specify the SNA/File Services compression technique used to compress the input data stream. Currently, the only value allowed by the SNA/File Services formats is X'01'.

Note: Do not specify this parameter if you specify a USERCOMP operand. The USERCOMP operand specifies a user-defined compression algorithm, rather than one defined by SNA/Files Services.

FSPARMS  Specify the input parameters for the SNA/File Services compression technique. Currently, one hexadecimal byte represents the prime control character. This operand is optional, and you can only specify it if you also specify the FSCOMPTQ operand.

USERCOMP  Specify the name of a user-defined algorithm to compress the resource. You can specify up to 12 characters. This name is always written and transferred to the target node in the EBCDIC character set.

Note: Do not specify this operand if you specified the FSCOMPTQ to use an SNA/File Services compression technique.
If you want to specify a compression exit based on the LZW technique (used by NetView DM for MVS), then enter `USERCOMP=NVDMLZW`. In this situation, do not make any entries for USERPARM.

**USERPARM** Specify the input parameters for the user specified compression algorithm. The algorithm can be up to 27 bytes long. This operand is optional. You can only specify these user parameters if you also specified the USERCOMP operand to use a user-defined compression algorithm.

**Note:** The entire character string must be enclosed within single quotation marks.

**DISP** Specify how the resource is prepared in the resource repository. You cannot specify this operand for system resources (microcode, microcode customization, software, and procedure resource types). The values you can specify are:

- **NEW** This means that the resource does not already exist in the resource repository.
- **OLD** This means that an existing resource is to be replaced. If the resource does not exist in the resource repository, it is prepared as in the case of `DISP=NEW`. If the resource has been previously locked by an application program accessing the D&CC API, NetView DM for MVS does not load the resource and issues an error message.
- **MOD** You can append data to an existing resource in the resource repository. When you use `DISP=MOD`, the following conditions must be verified:
  - The resource must not be compressed, or it must be compressed with the SNA/File Services compression technique.
  - The resource must be compatible with the resource existing in the holding file. It means:
    - The resource that is loaded must contain uniform text or binary text, depending on the resource in the resource repository.
    - If the resource in the resource repository is compressed with the SNA/File Services compression technique, the resource to append must also use the same compression technique with the same prime control character.
    - The record format that you obtain when specifying the VARLEN, UTEXT, and INDD operands must match the record format of the resource already loaded in the resource repository.

**UTEXT** Specify whether the resource being loaded contains a uniform text. A uniform text is a text data object that contains only characters belonging to an identified character set. When `UTEXT=YES` is specified, the NetView DM for MVS Coded Character Set (defined by NDMCOM customization macro) and the record organization (fixed or variable) of the user input file are stored in the resource directory. The Coded Character Set Identifier and record structure are transferred to the node along with the resource during the send function.

`UTEXT=YES` can be specified only when loading flat data, procedure, dump, error log, or trace resources. If the input data set has record format V or VB and `UTEXT=YES`, it is mandatory to code `VARLEN=YES`. If `VARLEN=NO` is coded, or VARLEN is not coded at all, the LOAD function terminates with return code 8.

**VARLEN** Specify whether the resource you are loading will be stored as a variable length record (`VARLEN=YES`) resource.

*When VARLEN=NO is coded the following applies:*
LOAD RESOURCE

RECFM=F  The resource is loaded as bytes steam when UTEXT=NO. If
          UTEXT=YES is specified, the resource is loaded as a fixed record
          length resource.

RECFM=V or VB  The record description words are removed from the records, and
          the resource is stored as a byte stream resource in the resource
          repository. UTEXT=YES is not allowed.

When VARLEN=YES is coded the following applies:

RECFM=F  The load function terminates with return code 8.
RECFM=V or VB  The record description words are stored with the records in the
          resource repository. The resource is marked as having variable
          record lengths.

Note:  A resource in sequential file format cannot be loaded if RECFM=U.

EMBCNTRL  Specify the ability to load multiple resource starting from a single data set.  The values
          you can enter are:

  NO  All the records of the data set specified in the INDD operand are loaded in the
          holding file as data records.  If you specify NO for the EMBCNTRL operand, do not
          specify a user string.  The default is NO.

  YES  The LOAD RESOURCE identifies the records starting with the control string and
          uses the information that follows them to load all the related data records up to the
          successive control string.  Specify a user string up to 12 characters long.  If you do
          not specify a user string, the default is .&EMBCNTRL&.  Omit the NAME operand,
          from the primary statement which is provided in the embedded control statement and
          identifies the resource uniquely.

Note:  The LOAD resource loads the files in the holding file, taking in consideration as
          values for the LOAD RESOURCE operands the values that are specified in the primary
          statement.  If no value is specified in the primary statement, the values used in the
          embedded control statement are used.  If no value is specified in the embedded control
          statement, the default values that are specified in the LOAD RESOURCE operand are
          used.

Example:  Here is an example of a LOAD RESOURCE statement to load a flat data resource type into
          the repository.

          LOAD RESO NAME=FLA.PROVA.18.NDM,
          RESTYPE=ZERODOT1/ZERODOT2/ZERODOT3,
          INDD=DSETS,
          MATCHATT=(3,N)
          TYPEAT=(3,ORDEREDN),
          DISP=(OLD)

LOAD RESOURCE (Embedded Form):  The embedded LOAD RESOURCE statement has a fixed format
          as shown in Figure 97 on page 293.  Each embedded statement must contain all the entries.  If you do
          not want to specify any value for an operand, leave the operand value blank: the default value is
          assumed. Numeric entries must be right-justified, and preceded by zeros or blanks.  The meaning of each
          entry and of its default, where applicable, is the same as for the corresponding operand entry described in
          the previous section.

          The embedded control statement record you want to load can be separated into segments.  Each segment
          must begin with the .&EMBCNTRL&. string, or with the string you specified in the EMBCNTRL operand,
          and end with +, except the last.  The position of the parameter that is specified in the embedded control
statement must reflect the format that follows, even when the record is separated into segments. The user string must be exactly 12 characters long. If you specify a user string that is fewer than 12 characters, you must complete it with blanks.

<table>
<thead>
<tr>
<th>Position</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-12</td>
<td>PREFIX</td>
<td>&amp;EMBCNTRL&amp;. or prefix selected by LOAD RESOURCE.</td>
</tr>
<tr>
<td>13-16</td>
<td>RESTYPE</td>
<td>Resource type.</td>
</tr>
<tr>
<td>17-80</td>
<td>NAME</td>
<td>Resource name.</td>
</tr>
<tr>
<td>81</td>
<td>CLCODES3</td>
<td>Class codes.</td>
</tr>
<tr>
<td>82</td>
<td>CLCODES4</td>
<td>Class codes.</td>
</tr>
<tr>
<td>83-112</td>
<td>MATCHATT</td>
<td>Match attributes. See note 1.</td>
</tr>
<tr>
<td>113-142</td>
<td>GENERATT</td>
<td>Generate attributes at the target. See note 2.</td>
</tr>
<tr>
<td>143-182</td>
<td>TYPEAT</td>
<td>Type of match attribute. See note 3.</td>
</tr>
<tr>
<td>183</td>
<td>COMPRSD</td>
<td>Resource compression state. The allowed values are: Y=Yes, N=No.</td>
</tr>
<tr>
<td>184-189</td>
<td>OBJECTSZ</td>
<td>Object decompressed size.</td>
</tr>
<tr>
<td>190</td>
<td>TOCOMPRS</td>
<td>Resource to be compressed or not. The allowed values are: Y=Yes, N=No.</td>
</tr>
<tr>
<td>191</td>
<td>FCOMPTQ</td>
<td>SNA compression technique.</td>
</tr>
<tr>
<td>192</td>
<td>FSPARMS</td>
<td>SNA compression operands.</td>
</tr>
<tr>
<td>193-204</td>
<td>USERCOMP</td>
<td>User compression algorithm.</td>
</tr>
<tr>
<td>205-231</td>
<td>USERPARM</td>
<td>User compression operand.</td>
</tr>
<tr>
<td>232</td>
<td>DISP</td>
<td>Disposition on load operation. The allowed values are: N=New, O=Old, M=Modify.</td>
</tr>
<tr>
<td>233</td>
<td>UTEXT</td>
<td>Resource being loaded contains a uniform text. The allowed values are: Y=Yes, N=No.</td>
</tr>
<tr>
<td>234</td>
<td>VARLEN</td>
<td>Resource must be load in variable record format. The allowed values are: Y=Yes, N=No.</td>
</tr>
<tr>
<td>235-240</td>
<td>PROCSIZE</td>
<td>Size of the resource at the target.</td>
</tr>
<tr>
<td>241-255</td>
<td>&quot;</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

Figure 97. Format of embedded LOAD statement

Notes:

1. Each field has 10 elements made up of three characters. The layout is XXY, where XX = 01-10 and Y = M or N.
2. Each field has 10 elements made up of three characters. The layout is XXY, where XX = 01-10 and Y = N or G.
3. Each field has 10 elements made up of four characters. The layout is XXYY, where XX = 01-10 and YY can assume the following values:
   - UN = UNSPECIFIED
   - OC = ORDEREDC
   - ON = ORDEREDN
   - OD = ORDEREDD
   - OT = ORDEREDT
   - SY = SYSTYPE
   - LU = LUNAME
   - NE = NETID
   - GV = G00V00
   - ST = SUBTREE

See “LOAD RESOURCE” on page 288 for more details on the operands.
PREPARE CLIST/JOB

Use the PREPARE CLIST statement to load an LU0 CLIST resource type from a user input data set at the host into the resource repository. Similarly, you can use the PREPARE JOB statement to load jobs into the resource repository.

Note: You can also use these statements to load these resource types from a user input data set using the FORMAT batch utility, for compatibility with previous releases of NetView DM for MVS.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREpare CLIST</td>
<td>INDD={ddname/filedefname}</td>
<td></td>
</tr>
<tr>
<td>PREpare JOB</td>
<td>.NAME=resource-name</td>
<td></td>
</tr>
<tr>
<td>PREpare PROGram</td>
<td>[.DISP=[NEW</td>
<td>OLD]]</td>
</tr>
<tr>
<td></td>
<td>[.FMN={filemodenumber}]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.LRLEN=80</td>
<td>lrlength]</td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.SRCTYPE=PS</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

You must always specify INDD and NAME. Note that the CLIST resource type is not supported by VSE nodes, and the JOB resource type is only supported by VSE nodes.

INDD Specify the input data set name. You must specify this operand. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. *ddname* is specified by a DD statement in the JCL file.

NAME Specify the name that you want to give to the CLIST or JOB when it is stored in the repository. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

DISP Specify how the resource is prepared in the resource repository. The values you can specify are:

- **NEW**  This means that the resource does not already exist in the resource repository.
- **OLD**  This means that an existing resource is to be replaced. If the resource does not exist in the resource repository, it is prepared as in the case of DISP=(NEW).

FMN Specify the optional file mode number which you can assign to a CLIST (the range can be from 0 to 6; the default is one).

LRLEN Specify the logical record length (in bytes) of the data. It must be at least 2 bytes, and no greater than 256 bytes for CLISTS, and 80 bytes for jobs. SSP CLISTS can have a record length from 40 to 120 bytes (records larger than 120 bytes are truncated when they are received by the node).

For DPPX nodes, the logical record length must be a multiple of 256, otherwise the TCP will not transmit the resource.

MOD  

VER  Specify a modification level and a version number for the resource. You can use these later to help identify the resource.

SRCTYPE Specify the type of input data set that you are loading. You can specify:

- **PS**  Physical sequential
**ESDS** VSAM entry-sequenced

**KSDS** VSAM key-sequenced

A physical sequential output file can be a member of a partitioned data set, a GDG data set, or a sequential data set of any device type.

**Examples:** This is an example of a PREPARE JOB statement:

```plaintext
PRE JOB NAME=JOB1,
        INDD=USER,
        VER=00,
        MOD=00,
        SRCTYPE=PS,
        DISP=(NEW)
```

And here is an example of a PREPARE CLIST statement:

```plaintext
PRE CLIST NAME=CLIST1,
        INDD=USER,
        VER=00,
        MOD=00,
        VARLEN=N,
        SRCTYPE=PS,
        DISP=(NEW)
```
## PREPARE DATASET/PRINT

Use the PREPARE statements to load data sets and *print data* resource types into the repository.

**Note:** You can also use these statements to load data sets and *print data* resource types into the repository using the FORMAT batch utility, for compatibility with previous releases of NetView DM for MVS.

### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREpare DATAset</td>
<td>INDD={ddname</td>
<td>filedefname}</td>
</tr>
<tr>
<td>PREpare PRINt</td>
<td>,(FCTNAME</td>
<td>NODE</td>
</tr>
<tr>
<td></td>
<td>,ID=resource-name</td>
<td>VSE only</td>
</tr>
<tr>
<td></td>
<td>,LBLRIND={LB</td>
<td>LR}</td>
</tr>
<tr>
<td></td>
<td>,[ALIAS=(name1,name2,...,name16)]</td>
<td>VSE only</td>
</tr>
<tr>
<td></td>
<td>,CATALOG=name,CATPSW= password</td>
<td>DPPX only, SSP, VSE only</td>
</tr>
<tr>
<td></td>
<td>,CC=(No</td>
<td>ANSI</td>
</tr>
<tr>
<td></td>
<td>,COMPRS=(No</td>
<td>Yes)</td>
</tr>
<tr>
<td></td>
<td>,CONTIG=(No</td>
<td>Yes)</td>
</tr>
<tr>
<td></td>
<td>,[DATATYPE={BIN</td>
<td>EBCDIC</td>
</tr>
<tr>
<td></td>
<td>,[DISP=NEW</td>
<td>OLD</td>
</tr>
<tr>
<td></td>
<td>,[DISAC={X'0000'</td>
<td>X'hhh'}]</td>
</tr>
<tr>
<td></td>
<td>,[DSORG={SAM</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>,[EXT=0</td>
<td>number-of-extents]</td>
</tr>
<tr>
<td></td>
<td>,[FILEPSW=file-password]</td>
<td>POSC only</td>
</tr>
<tr>
<td></td>
<td>,[FMN={1</td>
<td>filename}]</td>
</tr>
<tr>
<td></td>
<td>,[KEYLEN=key-length]</td>
<td>VSE only</td>
</tr>
<tr>
<td></td>
<td>,[KEYPOS=key-position]</td>
<td>POSC only</td>
</tr>
<tr>
<td></td>
<td>,[LANCHAR={LOCAL</td>
<td>MIRRUPD</td>
</tr>
<tr>
<td></td>
<td>,[LBLLEN=256</td>
<td>logical-block-size]</td>
</tr>
<tr>
<td></td>
<td>,[LRLEN=256</td>
<td>logical-record-length]</td>
</tr>
<tr>
<td></td>
<td>,[MAINTAIN=(No</td>
<td>Yes)]</td>
</tr>
<tr>
<td></td>
<td>,[NLBLR=number-of-blocks-or-records]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>,[PROTECT={None</td>
<td>Write</td>
</tr>
<tr>
<td></td>
<td>,[RECOVERY={No</td>
<td>Yes}]</td>
</tr>
<tr>
<td></td>
<td>,[SHR={UPD</td>
<td>ADDEXCL</td>
</tr>
<tr>
<td></td>
<td>,[SRCTYPE={PS</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>,[VARLEN=(No</td>
<td>Yes)]</td>
</tr>
<tr>
<td></td>
<td>,[PAD=(No</td>
<td>Yes)]</td>
</tr>
<tr>
<td></td>
<td>,[VERIFY=(No</td>
<td>Yes)]</td>
</tr>
<tr>
<td></td>
<td>,[VOLSER=volume-serial-number]</td>
<td></td>
</tr>
</tbody>
</table>

ID, LBLRIND, and FCTNAME|NODE|GROUP are optional only if they are specified as embedded control statements in the input file. Otherwise, these operands are all required. If you do not specify values for any of these operands, NetView DM for MVS assumes you are using embedded control statements.

**INDD**

Specify the input data set name. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. *ddname* is specified by a DD statement in the JCL file.
FCTLNAME
NODE
GROUP

Use one of these operands to identify the resource in the repository. You can use from 1 to 8 alphanumeric characters (starting with a letter) to define either the file control name in which the resource will be stored in the repository, or the node or node group to which the resource will be sent.

This operand is required when you are not using embedded control statements.

ID

Specify the simple or fully qualified name to identify the data set in the holding file. You can specify up to 44 alphanumeric characters. The data set is also known by this name at the node, unless the name at the node is specified in a transmission plan. The acceptable number of qualifiers you can use ranges from 1 to 22.

This operand is required when you are not using embedded control statements.

LBLRIND

Specify whether logical blocks (LB), or logical records (LR), are implied in the space allocation operands called NLBLR and EXT.

This operand is required when you are not using embedded control statements.

ALIAS

Specify the alias names (if any), used at the DPPX node to identify the data set. You can specify up to 16 alias names, if the data set is not compressed. If it is compressed, the maximum number of alias names is 10. Each alias consists of from 1 to 8 alphanumeric characters. The first character must be alphabetic.

CATALOG

Specify the name of the catalog that the dataset will be cataloged in at the node. The catalog name can be a simple or fully qualified name (up to 44 alphanumeric characters). The acceptable number of qualifiers you can use ranges up to 22.

CATPSW

Specify the password of the appropriate level for the catalog (up to 8 alphanumeric characters). If you specify this operand, you must also specify the CATALOG operand.

CC

Specify whether the first byte of each logical record is an ANSI or SCC control character, or whether there is no control character.

COMPRS

Specify whether or not you want to store the data in a compressed format.

CONTIG

Specify whether you want the space for each extent (primary or secondary) to be allocated contiguously on the DPPX volume.

DATATYPE

Specify the data characteristics. These can be:

- **BIN** Binary for PC format
- **EBCDIC** EBCDIC format
- **4680PSS** 4680 PSS format

These values are all supported for data to be sent to a POSC node. For data to be sent to a PDOS node, BIN and EBCDIC are supported.

DISP

Specify the disposition of prepared data sets in the holding file. The default value is **(NEW,KEEP)**. **DISP=(MOD,DELETE)** is useful in preparing data sets for a single destination node.

The values you can enter are:

- **NEW|OLD|MOD** These values specify the disposition of data sets as they are prepared:
  - NEW means that the data sets do not exist in the holding file.
  - OLD means that existing data sets are to be replaced.
  - MOD means that existing data sets are to be modified.
If the resource does not exist in the holding file, it is prepared as if DISP=(NEW). If the resource is already in the holding file, the input data is appended to it, if the resources are both either compressed or not compressed. If some resources are compressed and others not compressed, NetView DM for MVS issues an error message and does not process the prepare request. The attributes of the appended resource replace the existing ones.

**Note:** You can modify the original attributes of a resource by appending an empty resource.

**KEEP|DELETE** These values specify the disposition of prepared resources after they are subsequently processed at the host by the SEND, REPLACE, or CONVERT statements. This attribute can be overridden by the one specified in a transmission plan.

- **KEEP** means that prepared resources are to be retained in the holding file.
- **DELETE** means the resource is deleted by subsequent SEND, REPLACE, or CONVERT operations. If the operation is not completed successfully, the resource is not deleted. For SEND or REPLACE operations the resource is deleted only after the successful end of the transmission (for a single node), where SEND or REPLACE is specified. If a group of nodes is involved, the resource is deleted at the end of the phase (for each resource affected).

**DSAC** Specify the data set authorization class to be verified when the data set is accessed at the DPPX node. You define these credentials according to DPPX rules. Specify four hexadecimal digits.

**DSORG** Specify the type of data set organization.

**EXT** Specify the number of logical blocks or logical records (according to LBLRIND) for secondary allocation. Secondary allocation refers to each extension of the data set beyond the primary allocation. The maximum value is 65,535. The default is 0.

**FILEPSW** Specify the file password at the node. You can specify up to eight alphanumeric characters.

**FMN** Specify an optional file mode number. You can specify a number from 0 to 6. The default is 1.

**KEYLEN** Specify the length of the key field in bytes (from 1 to 255).

**KEYPOS** Specify the starting position of the key in the record, from the beginning of the data record.

**LANCHAR** Specify the Local Area Network CHARacteristics, if required. The default is LANCHAR=LOCAL. The values you can enter are:

- **LOCAL** A local file
- **MIRRUPD** A mirrored file, distributed on update
- **MIRRCLO** A mirrored file, distributed on close
- **COMPUPD** A compound file, distributed on update
- **COMPCLO** A compound file, distributed on close
**LBLEN**

Specify the logical block size (in bytes) of the data set in the resource repository. You can specify any number from 2 to 32,761. The default is 256.

**Note:** The logical block size does not have to be a multiple of the logical record length.

For DPPX nodes, the logical block size should not exceed 4,096, and must be a multiple of 256, otherwise the node will reject it.

For RPS nodes, the logical block size should be a multiple of the logical record length, otherwise the node will round it down to the next lowest multiple.

**LRLEN**

Specify the logical record length (in bytes) of the data set in the resource repository. The logical record length is not necessarily the same as the logical record length of the input data set. The number you specify must be at least 2 and no greater than the logical block size you specify, or the default value (32,761).

This operand is required when **LBLRIND=LR** (you are using logical records), and you are not using embedded control statements. When **LBLRIND=LB** (you are using logical blocks), the default is the logical block size (256).

If you are loading variable-length records, specified as **VARLEN=YES**, then the logical record length must be not less than four.

**MAINTAIN**

Specify whether or not the Relative Sequential Access Method (RSAM) of DPPX is to provide all maintenance for the data set at the node. **MAINTAIN=YES** means that users will comply with the restriction of not having logical records whose first 2 bytes contain X’FFFF’. **MAINTAIN=NO** means that DPPX users will have their own mechanism for deleting records and recognizing deleted records.

**NLBLR**

Specify the number of logical blocks or logical records (according to the LBLRIND operand) for primary allocation. The maximum value you can specify is 65,535. If you do not specify a value for the NLBLR operand, the number of records read from the input file for this data set is assumed, even if the number of records is greater than 65,535.

**PROTECT**

Specify the level of authorization required at the node for access to the data. The Catauth option means that the data set has the same level of protection as the catalog that it is stored in.

**RECOVERY**

Specify whether DPPX has to take special actions to preserve the integrity of the data set, in the event of a system malfunction.

**SHR**

Specify the types of concurrent access to the data set that are allowed at the node. **UPD** allows concurrent access by any number of receive-only users, any number of send/receive for update users, and one send/receive for an add user. This is the default value. **ADDEXCL** allows concurrent access by any number of receive-only users, or exclusive access by one send/receive for an update or add user. **ADDSSHR** allows concurrent access by any number of receive-only users, and one send/receive for an update or add user.

**SRCTYPE**

Specify the type of input data set. This can be:

- Physical sequential (PS)
- VSAM entry-sequenced (ESDS)
- VSAM key-sequenced (KSDS)

**VARLEN**

Specify whether the resource you are loading is in a variable-length record (**VARLEN=YES**), or a fixed-length record (**VARLEN=NO**). Only specify **VARLEN=Y** if the input file has **RECFM=V**, and if the resource is to be stored in the repository in variable-length format. The values you can enter are:
VARLEN=NO  The record is a not a variable-length record. Depending on the RECFM of the input file, the following takes place:

RECFM=F  The record format also states that the record is of fixed length, so the data is written into the repository.

RECFM=V  The record format states that the record is variable, but you are stating that it is fixed. If the value specified by LRLEN (the logical record length) in the PREPARE statement is greater than the length of the data of the current input record, the rest of the data part of the input record is padded with blanks up to the logical record length value. NetView DM for MVS can write the record into the repository.

VARLEN=YES  The record is a variable-length record. Only specify this if the input file also states that the record is variable (by RECFM=V). If you specify VARLEN=YES when the input file has RECFM=F, the PREPARE statement terminates with a return code 8. NetView DM for MVS issues an error message, and continues processing.

PAD  Specify whether to pad part of the input record with blanks. If you specify PAD=YES when RECFM=V and the value specified by LRLEN is greater than the length of the data of the current input record, the rest of the data part of the input record is padded with blanks up to LRLEN and written to the repository. You cannot specify VARLEN=YES and PAD=YES together.

VERIFY  Specify whether the data is to be read back and verified after each write operation.

VOLSER  Specify the serial number (from 1 to 6 alphanumeric characters) of the physical volume on which the data set is to be stored when it is transmitted. The default is a blank.

**PREPARE DATASET and PREPARE PRINT (Embedded Form):** The embedded PREPARE statement has a fixed format as described below. Each embedded statement must contain all the entries, except when a blank entry indicates a default. Numeric entries must be right-justified, and filled on the left with zeros or blanks. The meaning of each entry and of its default, where applicable, is the same as for the corresponding operand entry described in the previous section.
<table>
<thead>
<tr>
<th>Position</th>
<th>Operand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>PREFIX</td>
<td><code>.&amp;</code> or prefix selected by installation</td>
</tr>
<tr>
<td>3</td>
<td>MODIFIER</td>
<td><code>D = PREPARE DATASET, P = PREPARE PRINT</code></td>
</tr>
<tr>
<td>4</td>
<td>QUEUE</td>
<td><code>C = NODE, G = GROUP, F = FCTLNAME</code></td>
</tr>
<tr>
<td>5-12</td>
<td>NAME</td>
<td>Queue name</td>
</tr>
<tr>
<td>13</td>
<td>LBLRIND</td>
<td><code>B = LB, R = LR</code></td>
</tr>
<tr>
<td>14-18</td>
<td>LRLLEN</td>
<td>Logical record length or blank for default</td>
</tr>
<tr>
<td>19-23</td>
<td>LBLEN</td>
<td>Logical block size or blank for default</td>
</tr>
<tr>
<td>24-29</td>
<td>VOLSER</td>
<td>Volume serial number, left justified, or blank for default</td>
</tr>
<tr>
<td>30</td>
<td>SHR</td>
<td><code>1 = UPD, 2 = ADDEXCL, 3 = ADDSHR</code></td>
</tr>
<tr>
<td>31-35</td>
<td>NLBLR</td>
<td>Number of blocks/records for primary allocation, or blank for default</td>
</tr>
<tr>
<td>36-40</td>
<td>EXT</td>
<td>Number of blocks/records for secondary allocation, or blank for default</td>
</tr>
<tr>
<td>41</td>
<td>CONTIG</td>
<td><code>C = YES, blank = NO</code></td>
</tr>
<tr>
<td>42</td>
<td>PROTECT</td>
<td><code>N = NONE, W = WRITE, R = READWRITE, C = CATAUTH</code></td>
</tr>
<tr>
<td>43</td>
<td>VERIFY</td>
<td><code>V = YES, blank = NO</code></td>
</tr>
<tr>
<td>44</td>
<td>MAINTAIN</td>
<td><code>M = YES, blank = NO</code></td>
</tr>
<tr>
<td>45</td>
<td>CC</td>
<td><code>A = ANSI, S = SCC, blank = NO</code></td>
</tr>
<tr>
<td>46</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>47</td>
<td>RECOVERY</td>
<td><code>Y = YES, blank = NO</code></td>
</tr>
<tr>
<td>48-51</td>
<td>DSAC</td>
<td>Data set authorization credential (4 hexadecimal digits). There is no default; zero must be coded if the default is intended.</td>
</tr>
<tr>
<td>52</td>
<td>COMPRS</td>
<td><code>Y = YES, blank = NO</code></td>
</tr>
<tr>
<td>53</td>
<td>*</td>
<td>Reserved</td>
</tr>
<tr>
<td>54-97</td>
<td>ID</td>
<td>Data set name</td>
</tr>
<tr>
<td>98-n</td>
<td>BLANKS</td>
<td>Pad to input record length (<code>n</code> = record length)</td>
</tr>
</tbody>
</table>

**Figure 98. Format of embedded PREPARE statement**
Examples: Here is an example of how to specify a PREPARE DATASET operation. If you want to load three data set resources into the holding file using only primary control statements, you would include three statements in the job stream:

```plaintext
PREPARE DATASET INDD=DATADD1,
       ID=DATASET1,
       FCTLNAME=DATA01,
       LRLEN=80,LBLRIND=LR,SRCTYPE=PS

PREPARE DATASET INDD=DATADD2,
       ID=DATASET2,
       FCTLNAME=DATA02,
       LRLEN=80,LBLRIND=LR,SRCTYPE=PS

PREPARE DATASET INDD=DATADD3,
       ID=DATASET3,
       FCTLNAME=DATA03,
       LRLEN=80,LBLRIND=LR,SRCTYPE=PS
```

To do this using embedded statements, you would include all the file update resources in one host file (called, for example, UPDATES), and place your embedded statements in this file, at the beginning of each resource. For example:

```plaintext
.DRDATA/zerodot1.R8/zerodot.....................1
............................./zerodot/zerodot/zerodot.
DATASET1...................
.......data..............
.......data..............
.......data..............
```

302  NetView DM for MVS User's Guide
PREPARE MESSAGE

Use the PREPARE MESSAGE statement to prepare an LU0 message for transmission to nodes. You cannot use this statement with LU 6.2 nodes.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREpare MESSage</td>
<td>FCTLNAME=\textit{queue-name} \newline \text{MSG}='text'</td>
<td>All LU0 nodes</td>
</tr>
</tbody>
</table>

**FCTLNAME** Specify the name by which the resource is queued.

**MSG** Specify the message that you want to transmit to the node. You can enter up to 252 characters. You must enclose the message in single quotation marks. You can use the MSG operand up to 16 times in a single control statement.

**Example:** Here is an example of a PREPARE MESSAGE statement to send a message to the node, specified by the name of the queue the resource is held in:

PREPARE MESS FCTLNAME=QUEUE1,MSG='SAMPLE MESSAGE'
PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB

PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB

Use these PRINT statements to print CLIST, dataset, dump, errorlog, or job resource types. Resources that are stored in a compressed form are decompressed when they are printed. Formatted dumps must be formatted at the node, before being retrieved.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT CLIST</td>
<td>NAME=resource-name</td>
<td></td>
</tr>
<tr>
<td>PRINT JOB</td>
<td>[,MOD=0</td>
<td>nn], [VER=0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT DATASET</td>
<td>FCTLNAME=queue-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>,NAME=resource-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>,ORIGIN=origin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,CONV=EBCDIC</td>
<td>HEX]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT DUMP</td>
<td>FCTLNAME=queue-name</td>
<td></td>
</tr>
<tr>
<td>PRINT ERRORlog</td>
<td>,NAME=resource-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>,ORIGIN=origin</td>
<td></td>
</tr>
</tbody>
</table>

FCTLNAME Specify the name of the queue where the resource is held in the holding file. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

NAME Specify the resource name. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

ORIGIN Specify the name of the resource origin. If the resource was retrieved from a node, this is the node name. If the resource was prepared at the host, the origin is DSX.

CONV Specify whether the printout should be in character (EBCDIC), or dump (HEX) format.

MOD

VER Specify the modification level and version number of the resource.

Example: Here is an example of a PRINT CLIST statement to print a CLIST:

PRI CLIS NAME=VOLCAT.USERCAT.RESOURCE,MOD=1,VER=1
QUERY SPACE

You can use the QUERY SPACE statement at any time when you are working with the resource repository. This statement produces a printout of the amount of free space (in kilobytes) that remains in the data portion of the holding file and the NetView DM for MVS library.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUEry SPACE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are no operands for this control statement.
UNLOAD RESOURCE

Use the UNLOAD RESOURCE control statement to unload all LU 6.2 resource types directly from the resource repository into a user file. You do not need to write a user exit. Attributes related to this resource are printed on the SYSOUT file. The resource types you can unload are:

- AS/400 data object containment structure
- AS/400 object
- Configuration file
- Flat data
- Microcode (and microcode customization)
- Procedure
- Relational data
- Software
- Dump
- Error log
- Trace

When unloading a resource, NetView DM for MVS automatically decompresses the resource if the resource was compressed by one of these methods:

- SNA compression
- NVDMLZW algorithm
- User compression algorithm that was defined with the NDMCOM macro

If the resource contains “Uniform Text,” NetView DM for MVS performs “Data Translation” on the resource. Refer to “Uniform Text Processing” on page 263

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNLoad RESOurce</td>
<td>NAME=resource-name</td>
<td>CMFP, CMEP only</td>
</tr>
<tr>
<td></td>
<td>.RESTYPE=resource-type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.OUTDD=SYSOUT][ddname]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SCRATCH=NO][YES]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>[.OUTTYPE={NORMAL</td>
<td>APPEND</td>
</tr>
<tr>
<td></td>
<td>[.FIRSTQ=(DSN_mask[,n])]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.RESOHEAD=NO][YES]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.VOLUME=volume]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.UNIT=SYSDA.unit_type]</td>
<td></td>
</tr>
</tbody>
</table>

NAME

Specify the resource name.

You can use one or more wildcard characters (*) in the resource name. The wildcard, which can represent any token except the first one, replaces a token completely. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

RESTYPE

Specify the resource type code. This is mandatory unless the resource type is microcode or microcode customization, and the first token of the resource name is MCODE or MCUST.
OUTDD Specify the name of the output user data set that MAINTDAT uses to store the converted data. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. This operand is specified by a DD statement in the JCL file. The default is SYSOUT. The output file must be a sequential file (blocked or unblocked).

**Holding file resource has fixed length records:**
- If the output file has fixed record format, the length of the record of the user file must be greater or equal to the resource logical record length on the holding file. If the logical record length of the output user file is greater than the logical record length of the holding file resource, the output file records will be padded with blanks. If the logical record length of the output user file is smaller than the logical record length of the holding file resource, the output file records are truncated. A warning message is issued and the UNLOAD function ends with RC=4.
- If the output user file has records with variable lengths, the maximum record length of the output file should be greater than or equal to the logical record length of the holding file resource plus 4. RDWs are added in front of any record before writing the record to the output user file. If these conditions are not met, records are truncated. The UNLOAD function issues return code 4 and a warning message is issued.

**Holding file resource has variable length records:**
- If the output user file has records with fixed lengths, the unload process fails with return code 8 and an error message is issued.
- If the output file has a variable length record format, the maximum record length of the user file must be greater or equal to the maximum record length on the holding file. If not, the UNLOAD process will fail with return code 8 and an error message will be issued. The RDWs present in the holding file resource are stored in the user output file.

SCRATCH Specify whether you want the resource to be deleted from the repository after it has been unloaded. If you specify SCRATCH=YES and the resource has been previously locked by an application program accessing the D&CC API, NetView DM for MVS does not delete the resource and issues an error message.

MATCHIND Specify the list of matching indicators to be used by the host when unloading resources from the node. The MATCHIND values can be:

- `n` The qualifier ordinal number that the matching indicator refers to.
- `L` Set low value.
- `I` The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
- `H` Set high value.

The MATCHIND operand is valid only if you also specify the NAME operand. If the NAME operand is partially qualified, `H` default values are provided.

OUTTYPE Specify that you can unload more than one resource using only an UNLOAD statement. The OUTTYPE values are:

- `NORMAL` Unload the resource according to the values you specify in the NAME and MATCHIND operands. This is the default.
Unload all the resources selected from the resource repository in the data set specified in the OUTDD operand. All the resources are appended in the OUTDD data set. If you specify RESOHEAD=YES, a header appears before any resource you unload.

**Note:** For the data set specified in the OUTDD operand, the DISP operand must be equal to MOD.

**MULTIFILE** Unload each resource selected from the resource repository in a different data set. The MULTIFILE value creates as many data sets as the resources selected from the resource repository. The data sets are dynamically allocated with a specific naming convention that allows the user to identify the resource that generated the data sets.

Specify the FIRSTQ operand when OUTTYPE=MULTIFILE.

You cannot specify the OUTDD operand when OUTTYPE=MULTIFILE.

**FIRSTQ** Specify the mask that is used to build the name of the data sets dynamically allocated when OUTTYPE=MULTIFILE. The DSN_mask can have the following formats:

**Fixed** Without wildcard characters.

You unload each resource whose name matches the selection criteria specified in the NAME operand in a data set whose name begins with the DSN_mask that you specify in the FIRSTQ operand. The resource also has the DYN0000N token as the last qualifier. \( N \) is a sequential number that increases by one every time a resource is unloaded. For example, if the resource repository contains the following resources:

- FLAT
- FLAT.RESO01
- FLAT.RES01.NEW
- FLAT.RES02
- FLAT.RES02.12345
- FLAT.RES03
- FLAT.RES04

and you specify NAME=FLAT.* and FIRSTQ=(NVDM.UNLOADED.RESO) in the UNLOAD statement, the names of the data sets generated are:

<table>
<thead>
<tr>
<th>RESOURCES</th>
<th>DATA SET NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAT</td>
<td>NVDM.UNLOADED.RESO.DYN00001</td>
</tr>
<tr>
<td>FLAT.RES01</td>
<td>NVDM.UNLOADED.RESO.DYN00002</td>
</tr>
<tr>
<td>FLAT.RES01.NEW</td>
<td>NVDM.UNLOADED.RESO.DYN00003</td>
</tr>
<tr>
<td>FLAT.RES02</td>
<td>NVDM.UNLOADED.RESO.DYN00004</td>
</tr>
<tr>
<td>FLAT.RES02.12345</td>
<td>NVDM.UNLOADED.RESO.DYN00005</td>
</tr>
<tr>
<td>FLAT.RES03</td>
<td>NVDM.UNLOADED.RESO.DYN00006</td>
</tr>
<tr>
<td>FLAT.RES04</td>
<td>NVDM.UNLOADED.RESO.DYN00007</td>
</tr>
</tbody>
</table>

**Variable** With a wildcard character.

You unload each resource whose name matches the selection criteria specified in the NAME operand in a data set whose name reflects the following variable mask that you specified in the FIRSTQ operand:

\[
\text{FIRSTQ} = (\text{DSN}\_\text{token1}.\text{DSN}\_\text{token2}.*\text{,2})
\]

where \( \text{DSN}\_\text{token1}.\text{DSN}\_\text{token2}.* \) is the DSN_mask.
The wildcard in the FIRSTQ operand is replaced by the token in position \( n \) of the resource name, where \( n \) is the value specified in the second term of the FIRSTQ operand.

Exceptions occur in the following cases:

- The resource name has fewer than \( n \) tokens.
- The token in position \( n \) has a numeric value.
- The resulting data set name has already been generated.

In all these cases the wildcard is replaced by the DYN0000N token where \( N \) is a sequential number. For example, if the resource repository contains the following resources:

```
FLAT
FLAT.RESO1
FLAT.RESO1.NEW
FLAT.RESO2
FLAT.RESO2.12345
FLAT.RESO3
FLAT.RESO4
```

and you specify NAME=FLAT.* and FIRSTQ=(NVDM.UNLOADED.RESO.*,2) in the UNLOAD statement, the names of the data sets generated are:

<table>
<thead>
<tr>
<th>RESOURCES</th>
<th>DATA SET NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLAT</td>
<td>NVDM.UNLOADED.RESO.DYN00001</td>
</tr>
<tr>
<td>FLAT.RESO1</td>
<td>NVDM.UNLOADED.RESO.RESO1</td>
</tr>
<tr>
<td>FLAT.RESO1.NEW</td>
<td>NVDM.UNLOADED.RESO.DYN00002</td>
</tr>
<tr>
<td>FLAT.RESO2</td>
<td>NVDM.UNLOADED.RESO.RESO2</td>
</tr>
<tr>
<td>FLAT.RESO2.12345</td>
<td>NVDM.UNLOADED.RESO.DYN00003</td>
</tr>
<tr>
<td>FLAT.RESO3</td>
<td>NVDM.UNLOADED.RESO.RESO3</td>
</tr>
<tr>
<td>FLAT.RESO4</td>
<td>NVDM.UNLOADED.RESO.RESO4</td>
</tr>
</tbody>
</table>

Whether you use fixed format or variable format, you must follow these rules:

- The name of the generated data set cannot have more than five tokens.
- Each token cannot have more than eight characters. If a token of the resource name is longer than 8 characters, the corresponding token in the generated data set name is truncated up to the eighth character.

**RESOHEAD** Specify if you want a header inserted before any resource you unload from the resource repository in a data set. The RESOHEAD values can be:

- **NO** No header is added before the resource you unload from the resource repository. This is the default.
- **YES** A header is added before the resource you unload from the resource repository. You can specify this value only if OUTTYPE=APPEND. The header record has the following format:

  \[ .&RESO global name \]

  Where *global name* is the name of the resource you want to unload.

**VOLUME** Specify the volume serial numbers for the UNLOAD data sets.

Each value can be up to six characters. You can use the VOLUME operand when OUTTYPE=MULTIFILE.

This operand is optional.
UNLOAD RESOURCE

UNIT Specify the storage device type that the VSAM data sets reside in.

- The UNIT operand relates to the volume (volume label) of the VOLUME operand.
- You can specify up to eight characters.

This operand is optional.

Example: Here is an example of an UNLOAD RESOURCE statement to unload a flat data resource type from the repository into a user data file:

```
UNL RESO NAME=CORPID.XXX.*,RESTYPE=FLATD,SCRATCH=Y,MATCHIND=(3,L)
```

The name is partially defined, using the * token. Suppose that there are two resources in the repository:

- CORPID.XXX.A
- CORPID.XXX.B

This statement selects the lowest value for the variable token, as the matching indicator operand is set to L (set lowest value). The name with the lowest value in this example is CORPID.XXX.A.
Use the ULOCK RESOURCE control statement to unlock LU 6.2 resources that have been locked in the resource repository. Resources are locked by programs accessing the D&CC API and invoking the Lock_Catalog_Entry verb (DCCLOCKC), described in the Distribution and Change Control Application Programming book.

You cannot delete or change a locked resource until either the application program issues the Unlock_Catalog_Entry verb, or until you submit a ULOCK RESOURCE control statement.

To use the ULOCK RESOURCE statement, you must have Maintain Resource Repository and Plan Library authorization, as this statement is for resource repository maintenance purposes only.

The ULOCK RESOURCE statement removes the lock from the resource regardless of the application program that locked the resource.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ULock RESOurce</td>
<td>NAME=resource-name ,RESTYPE=resource-type</td>
<td>CMFP, CMEP only</td>
</tr>
</tbody>
</table>

NAME Specify the resource name. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

RESTYPE Specify the resource type code. Refer to “NetView DM for MVS Resource Types” on page 566 for information about possible type codes. This operand is mandatory.

Example: Here is an example of how to use a ULOCK RESOURCE statement to remove the lock status from a flat data resource type:

ULO RESO NAME=CORPID.PROVA.18.NDM,RESTYPE=FLATD
Chapter 16. Preparing and Submitting Transmission Plans

This chapter begins by describing the purpose and structure of transmission plans. It then describes the control statements you can use with transmission plans. These are divided into three categories:

- SUBMIT control statements to prepare and submit plans
- SUBMIT transmission functions for LU0 resource types
- SUBMIT transmission functions for LU 6.2 resource types

The statements are described in alphabetical order within each category.

What Is a Transmission Plan?

Transmission plans contain scheduling information. A plan describes operations you want to perform on resources, and groups these operations together according to nodes.

A transmission plan consists of one or more phases. Phases contain sequences of one or more transmission functions, followed by an optional batch job.

You create a transmission plan in the NetView DM for MVS plan library. You submit it to the transmission control program (TCP) which transmits the plan. Submitted plans are stored in the transmission control file (TCF). When you create a plan, you define:

- A plan name, which identifies the plan
- Whether the plan is recursive or not
- The number of hours a recursive plan can be delayed by each day
- Any notes associated with the plan

Note: When running more than one NetView DM for MVS application on the same system, do not give identical names to transmission plans from different NetView DM for MVS applications. This applies even if separate plan libraries are used for the various NetView DM for MVS applications. The use of identically named transmission plans can cause problems.

Recursive Plans

A NetView DM for MVS plan can be recursive. This means that once you have submitted it to the TCF, NetView DM for MVS will execute the plan every day at the scheduled time, until it is deleted from the TCF. In effect, NetView DM for MVS generates a new plan every 24 hours. The plan generated in this way is called the plan recursion for that day. You can delete a plan recursion, if transmission is not required on a certain day.

You schedule the plan when you define attributes for a phase. You also define a start-time delay. This means that if the plan cannot be executed on time, NetView DM for MVS will start the plan within the number of hours you specify in the start-time delay, or not at all.
What Is a Transmission Plan? (End User)

A recursive plan can contain one or more phases.

**Notes**

- If a plan is recursive, the validation of the plan is performed only when the plan is submitted. If you make changes to the plan later, for example, to the password or logon values of the node, the initial values continue to be considered in the next recursions of the plan. To avoid failures in the transmission of such a plan, you should erase the plan and resubmit it.

- Because you submit a recursive plan only once, any changes that you make after the plan has been submitted will not come into effect immediately, even if you are running Batch Utilities with NETCHNG=IMMEDIATE. For the changes to have effect, you must resubmit the plan.

**Phases**

A phase is a group of one or more functions that have the same scheduled time and date, and address the same node or group of nodes. For example, if you want to send various resources to a single node or single group of nodes, you can group these functions together into a phase. If, however, you want to retrieve a resource from a node and send the same resource to a different node or group of nodes, you have to create two separate phases.

When you define a phase you define:

- A phase name.
- The name of the node or group of nodes the phase is addressing.
- The date and time when the phase is executed. If the plan is recursive, all of the phases must have the same scheduled date, but they can have different scheduled times.
- A cutoff time or interval within which the phase must be completed.
- Whether the phase is to be held or not.
- Whether the processing of the phase depends on the successful execution of another phase within the same plan.
- Whether or not tracking is to be performed for the phase.
- Whether a batch job is to be run at the end of the phase.
- A transmission window within which the phase must be executed.

Within a phase you can also specify a set of logical conditions. See “Conditioning between Phases” on page 315 and “Conditioning within a Phase” on page 315.

**Functions**

The basic level of transmission operation is the function. The function specifies the transmission activity to be performed on a resource or resources. Transmission functions are listed in Table 34 on page 327 for LU0 resources, and Table 37 on page 343 for LU 6.2 resources.
What Is a Transmission Plan? (End User)

Conditioning within a Phase

Within a plan, you can make the execution of a phase or transmission function *conditional* upon the return code from some previous phase or function. When the TCP executes the functions in a plan, the functions terminate with a return code (from 0 to 8), which indicates whether the function was successful. Table 46 on page 405 shows the transmission function return codes. You can condition the processing of functions in a sequence by assigning a *termination value* to the functions when you create the plan. NetView DM for MVS then compares the termination value you assigned with the return code produced by transmission.

When the termination value is 0, no termination value is applied. If a you specify a value other than zero, and a function executes and returns with a return code that is also not zero, greater than or equal to the value you defined, *none* of the remaining functions in the sequence will be executed.

There are three levels at which you can specify the termination value:

- **At the phase level.** The termination value you define for the phase will apply to *all* the functions in that phase. You can override this value for individual sequences within the phase (see level 3 below).

- **At the phase for node level.** The termination value you specify will apply to all the phases for a specific node, instead of all the phases as above.

- **At the sequence level.** You can specify a sequence termination value, which will apply to all the functions in a particular sequence.

Conditioning between Phases

At the phase level, but not at the sequence or functional level, you can specify:

- A conditioning phase
- A conditioning value
- Conditioning criteria

These specify whether the phase is to be executed at all. The phase that must be completed first is called the *conditioning phase*. The phase whose execution depends on the return code from the conditioning phase is called the *conditioned phase*. When you define a conditioned phase, you specify a *conditioning value* and *conditioning criteria*. The conditioning criteria for a phase can either be:

- **GREATER THAN OR EQUAL TO:** The conditioned phase is executed if the conditioning phase was unsuccessful. In this case, the return code from the conditioning phase is “greater than or equal to” the termination value assigned to the conditioned phase.

- **LESS THAN OR EQUAL TO:** The conditioned phase is executed if the conditioning phase was successful. In this case, the return code from the conditioning phase is “less than or equal to” the termination value assigned to the conditioned phase.
What Is a Transmission Plan? (End User)

The following example illustrates a plan with three phases. Phases 2 and 3 are both conditioned by phase 1:

**Example**

**Phase 1:** Send some new software to a node, and execute it there.

**Phase 2:** If Phase 1 is successful, delete the previous version of the software stored at the node.

**Phase 3:** If Phase 1 is unsuccessful, retrieve diagnostic information from the node, delete the new version of the software, and send some messages.

In this example, you would enter the following values for Phase 2:

- **Conditioning phase:** PHASE01
- **Conditioning criteria:** 2
- **Conditioning value:** 04

and these values for Phase 3:

- **Conditioning phase:** PHASE01
- **Conditioning criteria:** 1
- **Conditioning value:** 08
Defining a Transmission Plan

You may wish to print a transmission plan which already exists in the plan library, to find one with a structure similar to the one you want to create. At installation time, a default plan model is stored in the plan library. You can use this as a model.

Use these control statements in the following order:

1. Start with a single DEFINE PLAN statement to specify the plan name and the plan attributes (see “DEFINE PLAN” on page 319).

2. Follow this with a DEFINE PHASE statement to specify the phase name and attributes of the first or only phase in the plan (see “DEFINE PHASE” on page 321).

3. One or more of the transmission functions listed in Table 34 on page 327 or Table 37 on page 343. These will be executed in the order in which you specify them in the job stream.

4. If the plan contains more than one phase, then define additional DEFINE PHASE statements for each phase, followed by its function control statements.

The definition of each plan ends when any control statement other than those listed in 1 to 3 above is found (for example, another DEFINE PLAN statement, a SUBMIT PLAN statement, or an END statement), or when the end-of-file condition is found.

Here are some examples of the SUBMIT statements required to delete some existing plan definitions from the library and to define a new plan called MYPLAN02. This plan consists of two phases, with phase PH0002 being conditioned by PH0001.

```assembler
TITLE 'EXAMPLE OF SUBMIT STATEMENTS'
DELETE PLAN NAME=MYPLAN01
DELETE PLAN NAME=MYPLAN02
DEFINE PLAN NAME=MYPLAN02,AUTOSUB=YES
DEFINE PHASE NAME=PH0001,NODE=VSEXXX,DATE=031288,TIME=0900
REP DATASET NAME=AAA.CCC.XX/zerodot2
RETRIEVE DATASET NAME=SSSS,NODENAME=X.Y.Z
INITIATE JOB NAME=POWER12
DEFINE PHASE NAME=PH0002,GROUP=VSEYYY,DATE=031288,TIME=0900,
COND=(PH0001,LE,4)
REP DATASET NAME=SSSS,NODENAME=Q.W.E
SEND LIBRARY NAME=LIB44,SUBLIB=A
END
```

Figure 100. Sample SUBMIT statements to define a plan
Submitting a Transmission Plan

Once you have created a plan in the plan library, the plan will not come into effect until you submit the plan to the TCF. You can either submit the plan with the SUBMIT PLAN control statement, or, if you specified AUTOSUB=YES in the DEFINE PLAN statement, the plan is submitted automatically.

Note: You require a special user authorization (called Manage Submitted Plans, described in Chapter 12, “Defining User Profiles” on page 211) to submit plans. This authorization also allows you to manage plans in the TCF after you have submitted them. Your system administrator decides which users to grant this authorization to.

When you run the SUBMIT utility to submit a plan, NetView DM for MVS validates the plan by looking for any errors. Errors that NetView DM for MVS looks for fall into two major categories:

Severe
You cannot submit a plan if the plan has any severe errors.

Warning
Although you can submit a plan, a warning tells you that NetView DM for MVS will not be able to execute transmission functions successfully in the way in which you have defined them.

<table>
<thead>
<tr>
<th>The NetView DM for MVS validation process checks to see that:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• You have defined sequences in the plan correctly.</td>
</tr>
<tr>
<td>• The administrator has defined the nodes and resources in the DRD.</td>
</tr>
<tr>
<td>– See Chapter 14, “Defining and Assigning Resources” on page 239.</td>
</tr>
<tr>
<td>• The resources are presently contained in the resource repository. See Chapter 15, “Managing Resources in the Repository” on page 261.</td>
</tr>
<tr>
<td>• User authorizations to nodes and to resources have been respected. See Chapter 12, “Defining User Profiles” on page 211.</td>
</tr>
<tr>
<td>• Groups of resources and nodes have been defined correctly in the DRD. See “Organizing Nodes and Resources into Groups” on page 225.</td>
</tr>
<tr>
<td>• Conditioning among phases is logically correct. See “Conditioning within a Phase” on page 315.</td>
</tr>
<tr>
<td>• The functions and resources in each phase are valid for the types of node.</td>
</tr>
<tr>
<td>– See Table 34 on page 327.</td>
</tr>
<tr>
<td>– See Table 37 on page 343.</td>
</tr>
</tbody>
</table>

The Messages and Codes book lists all the error messages that can be generated during validation, and describes how to fix them.

Note: When a phase is conditioned by another phase, the validation process does not check that the resources mentioned in the conditioned phase are already contained in the repository. This is because the resources may be received into the repository from another node by the conditioning phase.
SUBMIT Control Statements for Transmission Plans

You can use the following SUBMIT control statements to prepare and submit transmission plans.

The syntax and operands of these statements are described on the following pages, together with examples of how to use them.

DEFINE PLAN

Use the DEFINE PLAN statement to define a transmission plan in the NetView DM for MVS plan library.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE PLAN</td>
<td>NAME=plan-name [,AUTOSUB=Yes</td>
<td>No] [,DELAY=nn] [,NOTE='user-text'] [,RECURS=Yes</td>
</tr>
</tbody>
</table>

NAME: Specify the name of the plan. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

Note: Do not use these special characters in a plan name: $, #, @, %, _. GIX may interpret the underscore character as a blank and IOF cannot interpret the other special characters, producing unreliable results.

AUTOSUB: Specify whether you want to automatically submit the plan to the transmission control file, after it has been defined. If you do specify YES, the plan is inserted in the plan library, submitted to the TCF, and deleted from the plan library immediately after its successful submission. If the plan is not submitted successfully, NetView DM for MVS saves the plan in the plan library, and does not delete it.

Specify NO to insert the plan into the plan library, but not submit the plan. This is the default value.

DELAY: This operand applies to recursive plans only (RECURS=YES). Specify a start-time delay for the recursive plan. Enter a number from 1 to 24 that specifies how many hours the execution of a recursive plan can be delayed each day. The plan will be executed in the interval between the scheduled time and the end of the start-time delay.

NOTE: Use this operand to enter any relevant reference information for the plan, up to a maximum of 132 characters.

RECURS: Specify whether the plan is recursive (YES), or non-recursive (NO).

PRIORITY: Use this operand to specify a priority for the execution of the plan.

Enter an integer from 0 to 9 to indicate the priority with which the plan is executed. 0 represents the highest priority, 9 represents the lowest priority. The default value is 4. The phases of the plan are scheduled first by priority and then by scheduled date and time.
DEFINE PLAN

**Example:** Here is an example of a DEFINE PLAN statement to define a transmission plan called TEST to the plan library:

```
DEF PLAN NAME=TEST,DELAY=10,RECURS=YES,PRIORITY=8
```

The plan called TEST in this example is defined in the Plan Library. It has a start-time delay of 10, which means that the plan can be delayed for up to 10 hours after its scheduled start time. It has a priority of 8.
DEFINE PHASE

Use the DEFINE PHASE statement to define a phase in a transmission plan.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME=phase-name</td>
<td></td>
<td>All node types</td>
</tr>
<tr>
<td>,{NODE=nodeid,GROUP=groupid}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[COND=(phasename,op,rc)]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[CONDRC=0</td>
<td>nn]</td>
<td></td>
</tr>
<tr>
<td>,[CUTTIME=hhmm</td>
<td>CUTINT=hhmm])</td>
<td></td>
</tr>
<tr>
<td>,[DATE=mmdyly]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[NOTE='user-text']</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[PASSWORD=(logonid,password)]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[PROC=(procname,procrc)]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[STATUS=RELEASE</td>
<td>HOLD]</td>
<td></td>
</tr>
<tr>
<td>,[TIME=hhmm]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[CONDMODE=ALL</td>
<td>SINGLE]</td>
<td></td>
</tr>
<tr>
<td>,[TRACK=Y[ES]</td>
<td>N[Ot]]</td>
<td></td>
</tr>
<tr>
<td>,[WINDOWS=((SU</td>
<td>MO</td>
<td>TU</td>
</tr>
<tr>
<td>,[CUTWDATE=mmdyly]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>,[CUTWTIME=hhmm]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NAME

Specify the name of the phase. You can specify from 1 to 8 alphanumeric characters.

**Note:** Do not use these special characters in a phase name: $, #, @, %. GIX may interpret the underscore character as a blank and IOF cannot interpret the other special characters, producing unreliable results.

NODE | GROUP

Specify either the name of the node, or the name of a group of nodes, that the phase addresses. The nodes and the node group must be defined in the DRD.

COND

You can specify these values for a conditioned phase:

- **Phasename** is the phase whose completion conditions the start of this phase (from 1 to 8 alphanumeric characters, with the first character alphabetic).
- **Op** specifies when the conditioned phase should be executed. It can be:
  - **LE** The phase is executed if the actual return code from phasename is lower than or the same as the rc you enter.
  - **GE** The phase is executed if the actual return code from phasename is higher than or the same as the rc you enter.
- **Rc** is the return code for the conditioning phase. You can specify a value from 0 to 99.

CONDRC

Specify the conditioning return code. Specify a value greater than 0 and, if any function ends with a return code greater than or equal to this value, the phase is terminated.

DATE

Specify the date when you want the phase to be executed. If the plan is recursive, all of the phases must have the same scheduled date, but they can have different scheduled times.

If you do not enter a value, the date is the same as when the phase is submitted.
DEFINE PHASE

NOTE You can enter any type of information related to this phase that you want to record for future reference, up to a maximum of 132 characters.

PASSWORD You can specify a password here that will override the value specified when the node was defined (see “CHANGE NODE/DEFINE NODE” on page 217). If you do not specify a value here, NetView DM for MVS uses the value specified when the node was defined. If the logon ID and password are not provided in the DRD, the phase definition is rejected.

PROC Use this operand to define a batch job to be run when the phase terminates. Specify procname and procr, where procname is the name of the batch job (from 1 to 8 alphanumeric characters, with the first character alphabetic), and procr is the phase return code value that conditions the submission of the batch job.

The batch job will be submitted to the operating system at the end of the phase, if the phase return code is less than or equal to this value. Use the DSXJOBS file to store the batch jobs to be run.

STATUS Specify the status that the phase should have when it is executed. RELEASE means that the phase is available for initiation when its scheduled time is reached. HOLD means that the phase is to be held until released by an operator or by a node user command (“remote phase release”). Phases defined as “released” can later be held by a control operator command.

TIME Specify the time when the phase will be executed. If you do not enter a time, the time when you submit the plan is used instead.

CUTTIME | CUTINT Specify the cutoff time for the phase. This is the time when execution of the phase should stop. You can either specify the time absolutely, using the CUTTIME operand, or as an interval, using the CUTINT operand. This interval is added to the time specified in the TIME operand, or the time assumed at submission.

When the cutoff time is reached, the TCP deletes all of the phase by nodes for this phase, unless their status is COMPLETED. If the cutoff takes place while NetView DM for MVS is sending resources to intermediate nodes to be fanned out, the transmission is interrupted immediately. This usually results in disk space being tied up at the node. You have to intervene to free this disk space. The cutoff time must be different from the scheduling time. If the cutoff time is less than the actual time, the phase cutoff will take place on the next day.

If you do not specify a value for CUTTIME or CUTINT, there is no cutoff time for the phase.

CONDMODE Use this operand if you specified a group of nodes in the GROUP operand. Specify the type of conditioning mechanism that you want to use. The default is ALL, which means that the phase conditioning is performed when all the phases for node of the conditioning phase are completed. If you specify CONDMODE=SINGLE, the return code of each phase for node in the conditioning phase is used to condition the execution (or the logical deletion) of the phase for node that addresses the same node in the conditioned phase, as follows:

- The phases for node for which the conditioning criterion is met are released
- The phases for node for which the conditioning criterion is not met are marked as “deleted by conditioning” (return code=12).

CONDMODE=SINGLE is accepted only if the conditioning phase and the conditioned phase (or phases) address the same group of nodes.
DEFINE PHASE

TRACK Specify whether tracking is to be performed for this phase. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro. You now have the choice to change this default value.

If you specify YES, all of the resources in the phase will be tracked. If you specify NO, the resources in the phase will not be tracked, except for LU 6.2 nodes.

WINDOWS Specify transmission windows for this phase. A transmission window is a time window within which you want transmission activities for this phase to take place.

You can specify one window per day in each phase. The initial time is the time when you want the phase to start. The initial time belongs to the day specified in the window.

The end time is the time when you want transmission activities to stop. The end time could belong to the day after the one specified in the window, because the maximum interval between times is 23 hours and 59 minutes. This means that each phase can contain up to seven windows, one for each day of the week.

Windows in a phase must not overlap, for example, if you define a window for Monday between 06:00 and 01:00 on Tuesday, you cannot define a window that begins on Tuesday between 0:00 and 01:00.

Enter the WINDOWS operand in the following format:

,WINDOWS=((SU,0600,1300),(MO,0600,1300))

CUTWDATE Specify a cutoff date for the phase with transmission windows defined. This is the date when execution of the phase containing windows must stop.

Note: If you define transmission windows, then you should use the CUTWDATE and CUTWTIME operands instead of the CUTTIME and CUTINT operands.

CUTWTIME Specify a cutoff time for the phase with transmission windows defined. This is the date when execution of the phase containing windows must stop.

Example: Here is an example of a DEFINE PHASE statement that defines a phase called PHASE1, with three transmission windows:

DEF PHAS NAME=PHASE1,NODE=NODECMEP,
WINDOWS=((SU,1300,0800),(MO,1700,2000),(FR,1700,1659)),
CUTWDATE=122994,CUTWTIME=1700

If you submit the transmission plan containing this phase inside a window, the TCP puts the phase in READY status. If you submit the plan outside of a window, the TCP holds the phase and puts it in the new status of HWIN, until the next window starts.

In this example, the phase will start at the beginning of a window, such as 13:00 on Sunday. If the phase is not completed by 08:00 on Monday, the status becomes HWIN, until the start of the next window.
DELETE PLAN

DELETE PLAN

Use the DELETE PLAN statement to delete a plan from the plan library.

You can delete your own transmission plans from the plan library. If you are the system administrator or you have the Manage Other Users’ Plans authorization, you can also delete plans created by other users.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete PLAN</td>
<td>NAME=plan-name</td>
<td>All node types</td>
</tr>
</tbody>
</table>

NAME          Specify the plan name. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

Example:     Here is a DELETE PLAN statement to delete a plan called MYPLAN from the plan library:
              DEL PLAN NAME=MYPLAN
QUERY SPACE

Use this statement to find out how much space is left in the plan library.

Syntax

<table>
<thead>
<tr>
<th>QUERY SPACE</th>
<th>No operands</th>
</tr>
</thead>
</table>

This statement has no operands.
SUBMIT PLAN

Use the SUBMIT PLAN statement to submit a plan contained in the plan library to the transmission control file (TCF) for processing. You cannot use this statement if you specified that you wanted to submit the plan automatically (AUTOSUB=YES) in the DEFINE PLAN statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBmit PLAN</td>
<td>NAME=plan-name [,.NEWNAME=tcf_plan-name]</td>
<td>All node types</td>
</tr>
</tbody>
</table>

NAME Specify the plan name. You can specify from 1 to 8 alphanumeric characters, starting with a letter.

NEWNAME Specify the new plan name in the TCF. You can specify from 1 to 8 alphanumeric characters, starting with a letter. This operand is optional. If not specified, the plan will be stored in the TCF with the name in the plan library (as specified by the NAME operand).

Example: This is an example of a SUBMIT PLAN statement to submit the plan called MYPLAN to the TCF:

SUB PLAN NAME=MYPLAN,
          NEWNAME=TCFPLAN
SUBMIT Transmission Functions for LU0 Node Types

This section describes the transmission functions that you can use with LU0 node types (listed in Table 70 on page 561). The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

Table 34 lists the transmission functions (control statements), that apply to LU0 node types, and the specific node types for each transmission function:

<table>
<thead>
<tr>
<th>Transmission Function</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete CLIST</td>
<td>Except VSE</td>
</tr>
<tr>
<td>Delete Data set</td>
<td>All node types</td>
</tr>
<tr>
<td>Delete Library/Member</td>
<td>VSE only</td>
</tr>
<tr>
<td>Delete Panel/Program</td>
<td>All node types</td>
</tr>
<tr>
<td>Initiate Function</td>
<td>Except VSE</td>
</tr>
<tr>
<td>Initiate JOB</td>
<td>VSE only</td>
</tr>
<tr>
<td>Retrieve CLIST</td>
<td>Except VSE</td>
</tr>
<tr>
<td>Retrieve Data set</td>
<td>All node types</td>
</tr>
<tr>
<td>Retrieve Dump</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>Retrieve Errorlog</td>
<td>DPPX only</td>
</tr>
<tr>
<td>Retrieve Library/Member</td>
<td>VSE only</td>
</tr>
<tr>
<td>Retrieve Panel/Program</td>
<td>All node types</td>
</tr>
<tr>
<td>Retrieve PTF</td>
<td>DPPX only</td>
</tr>
<tr>
<td>Send CLIST</td>
<td>Except VSE</td>
</tr>
<tr>
<td>Send Data set</td>
<td>All node types</td>
</tr>
<tr>
<td>Send Dump</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>Send Library/Member</td>
<td>VSE only</td>
</tr>
<tr>
<td>Send Messages</td>
<td>All node types</td>
</tr>
<tr>
<td>Send Panel</td>
<td>All node types</td>
</tr>
<tr>
<td>Send Print</td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>Send Program</td>
<td>All node types</td>
</tr>
<tr>
<td>Send PTF</td>
<td>DPPX only</td>
</tr>
</tbody>
</table>

Table 34. Transmission functions for LU0 node types
DELETE

DELETE

Use these DELETE statements to erase specific resource types from the node, or nodes, specified in the preceding DEFINE PHASE statement. You can either specify a single resource (NAME), or a group of resources (GROUPID).

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete CLIST</td>
<td>{NAME=resource-name</td>
<td>GROUPID=resource-group}</td>
</tr>
<tr>
<td>DELete PROGram</td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td>DELete PANELI</td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NODENAME=name-at-node]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete DATaset</td>
<td>{NAME=resource-name</td>
<td>GROUPID=resource-group}</td>
</tr>
<tr>
<td></td>
<td>.CATLG=catalog-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.CATPSW=catalog-password]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DSORG=SAM</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>[.FILEPSW=file-password]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NODENAME=name-at-node]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete LIBRary</td>
<td>{NAME=resource-name</td>
<td>GROUPID=resource-group}</td>
</tr>
<tr>
<td>DELete MEMBer</td>
<td>.LIBNAME=library-name</td>
<td></td>
</tr>
<tr>
<td>DELete PANELI</td>
<td>[.LMTYPE=C</td>
<td>R</td>
</tr>
<tr>
<td>DELete PROGram</td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NODELNAM=library-name-at-node]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NODESLNA=sublibrary-name-at-node]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBLIB=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

The LIBNAME and GROUPID operands are not supported for the DELETE LIBRARY statement.

NAME
GROUPID

Specify either the name of a single resource (NAME), or the name of a group of resources (GROUPID), to be deleted at the node.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

MOD

Specify the modification level of the resource.

VER

Specify the version level of the resource.

NODENAME

Specify the fully qualified name of the resource at the node. Refer to “Naming LU 0 Resources at the Node” on page 570 for information about how to name LU0 resources at the node.
KEYNUMB Specify a key number (from 0 to 8). Specifying 0 deletes all the keys of the related DSCB.

SUBCAT Specify the subcategory ID (from 00 to 16). The default is 00, which means to clear all of the subcategories.

Note: The following operands are for VSE nodes only.

CATLG Specify the name of the catalog where the resource is stored at the node.

CATPSW Specify the password required to access the catalog at the node.

DSORG Specify how the data set is organized.

FILEPSW Specify the password required for accessing the file where the data set is stored at the node.

LIBNAME Specify the name of the library to which the member, panel, or program belongs.

LMTYPE Specify the library type:

C Core image library
P Procedure library
R Relocatable library
S Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

NODELNAM Specify the name the library is known by at the node.

NODESLNA Specify the name the sublibrary is known by at the node.

SUBLIB Specify the sublibrary name.

Note: If the library type is S (LMTYPE=S), you must specify a sublibrary name.
INITIATE

Use the INITIATE FUNCTION statement to initiate the preceding DEFINE PHASE statement at the specified node. This means the transmission and execution of one or more command lists (CLISTs). The CLIST must be present in the repository, and must be authorized by an entry in the DRD. The size of the CLIST cannot be greater than 65280 bytes. The CLIST is not stored permanently at the node. It is stored in a temporary data set, which is deleted after processing.

The INITIATE JOB statement sends and executes VSE JOBS, in the same way as INITIATE FUNCTION performs with CLISTs. The JOB must exist in the repository, but VSE JOBS need not have been previously assigned to the node.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INItiate FUNCtion</td>
<td>{NAME=resource-name</td>
<td>GROUPID=resource-group}</td>
</tr>
<tr>
<td></td>
<td>[,VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,MOD=0</td>
<td>nn]}</td>
</tr>
<tr>
<td></td>
<td>[,REST={No</td>
<td>Yes}]</td>
</tr>
<tr>
<td></td>
<td>[,SYNC={Yes</td>
<td>No}]</td>
</tr>
<tr>
<td></td>
<td>[,EXECTIME=time]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INItiate JOB</td>
<td>NAME=resource-name</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td></td>
<td>[,MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[,VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

**NAME** Specify the name of the CLIST, or JOB resource type to be processed. Remember that a CLIST or JOB must already exist in the resource repository.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**Note:** Resource names starting with $DELETE.$PENDING are reserved, and may not be used.

**GROUPID** Specify the name of a group of resources (CLIST, or JOB resource types) defined in the DRD (from 1 to 8 characters).

**VER** Specify the version level of the resource.

**MOD** Specify the modification level of the resource.

**REST** Specify whether the function should be restarted if it ends with a temporary error. A temporary error (for example, a “LOST TERM”) can occur either while the CLIST is being sent, or when it is being executed. In either case, if **REST=**Y, the CLIST will be sent again and then executed again.

**SYNC** Specify whether you want the CLIST to be executed asynchronously or synchronously at the node. Specify:

- **SYNC=NO**, to execute the CLIST asynchronously. You cannot specify this value if you specified a group of resource in the GROUPID operand. You can only specify this value if the target of the phase is a POSC node type. Other types of node will
reject the INITIATE FUNCTION statement. If **SYNC=NO**, the node sends a response to NetView DM for MVS as soon as it receives the CLIST. NetView DM for MVS then terminates the SNA session with the node. Later, after the time specified in the **EXECTIME** operand, NetView DM for MVS will open another SNA session with the node, and send a “request for acknowledgement” to the node. If the CLIST has already finished executing, successfully or otherwise, the node will inform NetView DM for MVS of this. The return code will be returned and associated with the CLIST in the TCF. Otherwise, NetView DM for MVS must wait, and send additional “requests for acknowledgement” to the node at a later time, as follows:

- **If there are other** asynchronous INITIATE functions outstanding (due to asynchronous INITIATE functions being specified in other phases for the same node), NetView DM for MVS will issue the next “request for acknowledgment” at the time specified by the next outstanding INITIATE function.

- **If there are no other** asynchronous INITIATE functions outstanding, NetView DM for MVS will issue “requests for acknowledgment” at intervals specified in the TCP **RETINT** parameter, until an acknowledgment is received.

- **If the TCP job is terminated while asynchronous INITIATE functions are outstanding, and then the TCP restarts**, NetView DM for MVS will issue “requests for acknowledgement” based on the RETINT parameter, until all the acknowledgments are received.

- ****SYNC=YES** means that the CLIST will be executed synchronously by the node. This means that NetView DM for MVS waits (and the SNA session with the node remains active) until the node sends a response to NetView DM for MVS to say that the CLIST has completed its execution, either successfully or otherwise. **SYNC=NO** is the default value.

**EXECTIME**

You must specify this operand if you chose asynchronous execution (**SYNC=NO**), otherwise it is ignored. Specify the number of minutes (from 1 to 1440) that you estimate will be required to execute the CLIST at the node. When the time interval you specify here expires, after the time the CLIST was received at the node, NetView DM for MVS queries the node for the result of the CLIST execution.

**Example:** Here is an example of an INITIATE FUNCTION statement:

```plaintext
INI FUNC NAME=CL08ED1,
       VER=0,
       MOD=0
```
RECEIVE/RETRIEVE

Use the RECEIVE or RETRIEVE statements to retrieve the specified resources from the node, or group of nodes, that are defined in the preceding DEFINE PHASE statement.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve CLSt</td>
<td>{NAME=resource-name}</td>
<td>Except VSE</td>
</tr>
<tr>
<td>RETrieve PANELI</td>
<td>GROUPID=resource-group</td>
<td>Except VSE</td>
</tr>
<tr>
<td>RETrieve PROGram</td>
<td>{.CATLG=catalog-name}</td>
<td>Except SSP, VSE</td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NODENAME=name-at-node]</td>
<td>Except VSE</td>
</tr>
<tr>
<td></td>
<td>[.REPLACE=No</td>
<td>Yes]</td>
</tr>
<tr>
<td></td>
<td>[.RESYNC=Yes</td>
<td>No]</td>
</tr>
<tr>
<td></td>
<td>[.SCRATCH=No</td>
<td>Yes]</td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

Table 35 on page 334 shows how you can combine the NAME, GROUPID, CATLG, and NODENAME operands.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve DATAset</td>
<td>{NAME=resource name}</td>
<td>Except SSP, PDOS</td>
</tr>
<tr>
<td></td>
<td>GROUPID=resource-group</td>
<td>VSE</td>
</tr>
<tr>
<td></td>
<td>{.CATLG=catalog-name}</td>
<td>DPPX, SSP, VSE, only</td>
</tr>
<tr>
<td></td>
<td>{.COMPRS={No</td>
<td>Yes}}</td>
</tr>
<tr>
<td></td>
<td>{.DATATYPE={BIN</td>
<td>EBCDIC</td>
</tr>
<tr>
<td></td>
<td>{.DISP={NEW</td>
<td>OLD</td>
</tr>
<tr>
<td></td>
<td>{.DSORG={SAM</td>
<td>ESDS</td>
</tr>
<tr>
<td></td>
<td>{.FCTLNAME=queue-name}</td>
<td>Except VSE</td>
</tr>
<tr>
<td></td>
<td>{.FILEPSW=file-password}</td>
<td>Except PDOS</td>
</tr>
<tr>
<td></td>
<td>{.NODENAME=name-at-node}</td>
<td>Except VSE</td>
</tr>
<tr>
<td></td>
<td>{.REPLACE={No</td>
<td>Yes}}</td>
</tr>
<tr>
<td></td>
<td>{.RESYNC={Yes</td>
<td>No}}</td>
</tr>
<tr>
<td></td>
<td>{.SCRATCH={No</td>
<td>Yes}}</td>
</tr>
</tbody>
</table>
### Statements, Operands, Node Types

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve DUMP</td>
<td>{NAME=resource-name}</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td></td>
<td>[GROUPID=resource-group}</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td></td>
<td>[,CATLG=catalog-name]</td>
<td>Except SSP, VSE, PDOS</td>
</tr>
<tr>
<td></td>
<td>[,COMPRS=No</td>
<td>Yes]</td>
</tr>
<tr>
<td></td>
<td>[,DISP=(NEW</td>
<td>OLD</td>
</tr>
<tr>
<td></td>
<td>[,FCTLNAME=queue-name]</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td></td>
<td>[,NODENAME=name-at-node]</td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td></td>
<td>[,REPLACE=No</td>
<td>Yes]</td>
</tr>
<tr>
<td></td>
<td>[,RESYNC=Yes</td>
<td>No]</td>
</tr>
<tr>
<td></td>
<td>[,SCRATCH=No</td>
<td>Yes]</td>
</tr>
</tbody>
</table>

Table 35 on page 334 shows how you can combine the NAME, GROUPID, CATLG, and NODENAME operands.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve ERROrlog</td>
<td>NAME=resource-name</td>
<td>DPPX nodes only</td>
</tr>
<tr>
<td></td>
<td>[,CL=X’cc’]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,DISP=NEW</td>
<td>OLD]</td>
</tr>
<tr>
<td></td>
<td>[,FCTLNAME=queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,FROMDA=(X’aa’[,X’dd’])]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,FROMDATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,FROMTIME=hhmm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,NLBLR=number-of-blocks]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,SC=X’ss’)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,TODA=(X’aa’[,X’dd’])]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,TODATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,TOTIME=hhmm]</td>
<td></td>
</tr>
</tbody>
</table>

You must specify only one of NODELNAM, GROUPID, or NODESLNA.
You cannot use the LIBNAME and GROUPID operands with the RETRIEVE LIBRARY statement.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve LIBRary</td>
<td>{NAME=resource-name}</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td>RETrieve MEMBer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETrieve PANEL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETrieve PROgram</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**RECEIVE/RETRIEVE**

### Usage:
The table below shows the combination of the NAME, GROUPID, CATLG, and NODENAME operands that you can use:

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve PTF</td>
<td>NAME=resource-name, CATLG=catalog-name, [FCTLNAME=queue-name], [REPLACE=No</td>
<td>Yes], [RESYNC=Yes</td>
</tr>
</tbody>
</table>

#### Table 35. Applicability of RECEIVE and RETRIEVE operands

1. Must not contain any asterisks (*).
2. Must contain at least one asterisk (*).
3. Can contain asterisks (*).

You can also use asterisks to construct the name at the host. The table below shows the results you will get with different combinations of asterisks:

<table>
<thead>
<tr>
<th>Actual Resource Name at Node</th>
<th>Resource Name as Defined in Transmission Plan</th>
<th>Resulting Name at Host When Retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.B.C</td>
<td>*</td>
<td>C</td>
</tr>
<tr>
<td>A.B.C</td>
<td><em>.</em></td>
<td>B.C</td>
</tr>
<tr>
<td>A.B.C</td>
<td><em>.</em>.*</td>
<td>A.B.C</td>
</tr>
<tr>
<td>A.B.D</td>
<td>C.<em>.</em></td>
<td>C.B.D</td>
</tr>
<tr>
<td>A.B.C</td>
<td><em>.E.</em></td>
<td>A.E.C</td>
</tr>
</tbody>
</table>

#### Table 36. Using asterisks to construct the name at the host

**NAME**

Specify the name of the resource as it is known at the host. When you are retrieving resources from a VM node, you must not specify a value for NAME if you specify CATLG. The names in the catalog will also be used at the host.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.
CATLG Specify one of the following:
- The name of a DPPX user, volume, or master catalog
- An RPS partitioned data set
- The name of a CMS file
- A POSC directory

All of the resources in the catalog are retrieved for storage at the host, under the names found in the catalog.

CATPSW Specify the password required for accessing the catalog at the node.

CL Specify the class of error to be extracted. The entry should be two hexadecimal characters corresponding to a valid DPPX class of error. If you do not specify a value, NetView DM for MVS will use the default specified in the DPPX CLIST SYSICL.SYSERLOG.

COMPRS Specify whether the resource is to be transmitted in compressed form.

DATATYPE Specify the data characteristics.

DISP Specify the disposition of the resources to be retrieved (see DISP operand on page 297).

DSORG Specify how the data set is organized.

FCTLNAME Specify the name of the file control name under which resources are to be stored in the repository. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. If you do not specify a value for FCTLNAME, the default is the name of the node or group of nodes in the preceding DEFINE PHASE statement.

FILEPSW Specify the password required to access the file where the data set is stored at the node.

FROMDA Specify the start and end device address to search for errors. If you do not enter values, no limitation is placed on the search for error records. X'aa' is the adapter address. X'dd' is the device or secondary component address. These are both two-digit hexadecimal numbers. If you specify only the adapter address, all devices for that adapter are included. You cannot specify a device address without specifying the adapter. The results will be unpredictable if you use FROMDA, and the values you specify for CL and SC are other than hardware I/O. TODA is ignored if you do not specify a value for FROMDA.

FROMDATE FROMTIME TODATE TOTIME Specify the start and end date and time when you want one or more errorlogs to be searched. If you specify FROMDATE, but not TODATE, only the error log for the date specified in FROMDATE is retrieved. If you specify a start time with no date, today's date is used for both start and finish. If you do not specify TOTIME, the value specified for FROMTIME is used, if any. If neither is specified, no time limitations are placed on the extraction.

GROUPID Specify the name of a group of resources to be retrieved. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

LIBNAME Specify the name of the library to which the resource belongs. Do not use this with the RETRIEVE LIBRARY statement.

LMTYPE Specify the library type:
RECEIVE/RETRIEVE

C  Core image library
P  Procedure library
R  Relocatable library
S  Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

MOD  Specify the modification level of the resource.
VER  Specify the version number of the resource.
MODE  Specify the type of transaction request to be performed:
  • NORMAL indicates that this is a request to receive transactions
  • RETRANS indicates that this is a request for retransmission of transactions previously transmitted.

NLBLR  Specify the number of 256-byte logical blocks that should be allocated to store the extracted records. This space must be available to the system catalog at the 8100. The entry may be from 1 to 5 digits (from 1 to 65535).

NODELNAM  Specify the name by which the library is known at the node.

NODENAME  Specify the name under which the resource is stored at the node. Refer to “Naming LU 0 Resources at the Node” on page 570 for information about how to name LU0 resources at the node.

NODESLNA  Specify the name by which the sublibrary is known at the node.

REPLACE  Specify whether the retrieved resource is to replace the resource of the same name in the resource repository. If you specify a value for both REPLACE and DISP, the REPLACE value is ignored. REPLACE is meaningful only if the existing resource has the same file control name and origin, as well as the same resource name. REPLACE=Y has the same meaning as DISP=(OLD,KEEP). REPLACE=N has the same meaning as DISP=(NEW,KEEP).

RESYNC  Specify the resynchronization level to be applied, if necessary, during transmission of the resource.

SC  Specify the subclass of error to be extracted. Incident records not of this subclass will be rejected. Specify any two valid hexadecimal digits, indicating a valid DPPX subclass of error for the class indicated. If you do not enter a value, the default specified in the DPPX CLIST called SYSICL.SYSERLOG will be used.

Refer to DPPX Diagnosis: Service Aids, SC27-0627, for a description of the subclass types.

SCRATCH  Specify whether the retrieved resource is to be deleted from the node following retrieval and storage at the host.

SUBLIB  Specify the sublibrary name.

Note: If the library type is S (LMTYPE=S), you must specify a sublibrary name.

Example: Here is an example of a statement to retrieve a CLIST resource type:

RET CLIS NAME=CL08ED1,
      NODENAME=DS08ED1
SEND/REPLACE

Use the SEND statements to send resources to the node, or group of nodes, defined in the preceding DEFINE PHASE statement. Resources must be contained in the resource repository at the host. Use the REPLACE statement (which is the same as SEND) if the resource already exists at the node and you want to replace it.

Note: The SEND function is not supported by VSE nodes.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND CLIST</td>
<td>{NAME=resource-name}</td>
<td>Except VSE</td>
</tr>
<tr>
<td>SEND PANEL</td>
<td>GROUPID=resource-group</td>
<td>Except VSE</td>
</tr>
<tr>
<td>SEND PROGRAM</td>
<td>[.DSAC=X'0000'</td>
<td>X' hhhh']</td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NODENAME=name-at node]</td>
<td>Except VSE</td>
</tr>
<tr>
<td>REPLACE CLIST</td>
<td>[.PROTECT=None</td>
<td>Write</td>
</tr>
<tr>
<td>REPLACE PANEL</td>
<td>[.RESYNC=Yes</td>
<td>No]</td>
</tr>
<tr>
<td>REPLACE PROGRAM</td>
<td>[.SHR=UPD</td>
<td>ADDEXCL</td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.VOLSER=volume-serial-number]</td>
<td>DPPX, RPS only</td>
</tr>
</tbody>
</table>

See below for VSE nodes.

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>REPLACE LIBRARY</td>
<td>{NAME=resource-name}</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td>REPLACE MEMBER</td>
<td>GROUPID=resource-group</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td>REPLACE PANEL</td>
<td>NODES=library-name-at-node</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td>REPLACE PROGRAM</td>
<td>.LIBNAME=library-name</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td></td>
<td>[.LMTYPE={C</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.NODENAME=library-name-at-node]</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td></td>
<td>[.RESYNC=Yes</td>
<td>No]</td>
</tr>
<tr>
<td></td>
<td>[.SUBLIB=sublibrary-name]</td>
<td>VSE nodes only</td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
</tbody>
</table>

You can specify only one of NODENAME, GROUPID, or NODES. You cannot use the LIBNAME and GROUPID operands with the REPLACE LIBRARY statement.
### Statements

<table>
<thead>
<tr>
<th>SEND DATASET (not VSE)</th>
<th>OPERANDS</th>
<th>NODE TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATLG=catalog-name</td>
<td></td>
<td>VSE only</td>
</tr>
<tr>
<td>[CATPSW=catalog-password]</td>
<td></td>
<td>VSE only</td>
</tr>
<tr>
<td>[COMPRS=No</td>
<td>Yes]</td>
<td></td>
</tr>
<tr>
<td>[.DECOMPRES=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.DSAC=X'0000'</td>
<td>X'hhhh']</td>
<td></td>
</tr>
<tr>
<td>[.FCTLNAME=queue-name]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>[.FILEPSW=file-password]</td>
<td></td>
<td>Except PDOS</td>
</tr>
<tr>
<td>[.GROUPID=resource-group]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>[.NODENAME=name-at-node]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>[.ORIGIN=DSX</td>
<td>origin]</td>
<td></td>
</tr>
<tr>
<td>[.PROTECT= None</td>
<td>Write</td>
<td>Readwrite</td>
</tr>
<tr>
<td>[.RESYNC=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.SHR=(UPD</td>
<td>ADD</td>
<td>EXCL</td>
</tr>
<tr>
<td>[.VOLSER=volume-serial-number]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>[.BYDESTID=(No</td>
<td>Yes)]</td>
<td></td>
</tr>
</tbody>
</table>

You can specify a value for either NAME or GROUPID, but not both. If you specify only a value for FCTLNAME (or FCTLNAME and ORIGIN), then you can send several data sets at once.

For RPS node types you can only specify SHR=ADDEXCL.

### Statements

<table>
<thead>
<tr>
<th>SEND DUMP</th>
<th>OPERANDS</th>
<th>NODE TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>{NAME=resource-name}</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>GROUPID=resource-group</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.COMPRS=No</td>
<td>Yes]</td>
<td></td>
</tr>
<tr>
<td>[.DECOMPRES=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.DSAC=X'0000'</td>
<td>X'hhhh']</td>
<td></td>
</tr>
<tr>
<td>[.FCTLNAME=queue-name]</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.GROUPID=resource-group]</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.NODENAME=name-at-node]</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.ORIGIN=DSX</td>
<td>origin]</td>
<td></td>
</tr>
<tr>
<td>[.RESYNC=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.VOLSER=volume-serial-number]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
</tbody>
</table>

### Statements

<table>
<thead>
<tr>
<th>REPLACE DUMP</th>
<th>OPERANDS</th>
<th>NODE TYPES</th>
</tr>
</thead>
<tbody>
<tr>
<td>{NAME=resource-name}</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>GROUPID=resource-group</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>[.COMPRS=No</td>
<td>Yes]</td>
<td></td>
</tr>
<tr>
<td>[.DECOMPRES=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.DSAC=X'0000'</td>
<td>X'hhhh']</td>
<td></td>
</tr>
<tr>
<td>[.FCTLNAME=queue-name]</td>
<td></td>
<td>DPPX only</td>
</tr>
<tr>
<td>[.GROUPID=resource-group]</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.NODENAME=name-at-node]</td>
<td></td>
<td>Except VSE, PDOS</td>
</tr>
<tr>
<td>[.ORIGIN=DSX</td>
<td>origin]</td>
<td></td>
</tr>
<tr>
<td>[.PROTECT= None</td>
<td>Write</td>
<td>Readwrite</td>
</tr>
<tr>
<td>[.RESYNC=Yes</td>
<td>No]</td>
<td></td>
</tr>
<tr>
<td>[.VOLSER=volume-serial-number]</td>
<td></td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>Statements</td>
<td>Operands</td>
<td>Node Types</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>SEND/REPLACE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEND PRINTER</td>
<td>{NAME=resource-name} GROUPID=resource-group [CLASS=class]</td>
<td>DPPX, RPS only</td>
</tr>
<tr>
<td>REPLACE PRINTER</td>
<td>[COMPRS=No</td>
<td>Yes] [COPIES=1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You can specify a value for either the NAME or the GROUPID operand, but not both. One of them must be specified if the print data set is restricted, otherwise NetView DM for MVS will assume that it is unrestricted.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND MESSAGES</td>
<td>[FCTLNAME=queue-name]</td>
<td>DPPX, RPS, SSP, VSC, POSC, PDOS,</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND PTF</td>
<td>NAME=resource-name [CATLG=name-at-node]</td>
<td>DPPX nodes only</td>
</tr>
<tr>
<td>REPLACE PTF</td>
<td>[FCTLNAME=queue-name] [RESYNC=Yes</td>
<td>No] [VOLSER=volume-serial-number]</td>
</tr>
</tbody>
</table>

When a PTF is being sent to a diskette which contains previous PTF levels, you must use REPLACE, since the PTF control record ID is the same for all fix levels. PTFs sent to the nodes are not applied, but the INITIATE FUNCTION statement can be used to send a user-created CLIST that will apply the PTF.

**CATLG**
Specify the name of the catalog where the resource will be stored at the node.

**CATPSW**
Specify the password required for accessing the catalog at the node.

**CLASS**
Specify the print class to be used for printing one or more files at the node. You can specify a value from 1 to 36. If you do not specify a value, NetView DM for MVS will not request a class, and will use the default class at the destination DPPX node.

**COMPRS**
Specify whether the resource is stored in the repository in compressed form. You can only specify this for REPLACE functions.

**COPIES**
Specify the number of copies of one or more files to be printed (from 1 to 99).

**DECOMPRS**
If the resource is sent compressed, specify whether it is to be decompressed when it is received at the node. You can only specify this for REPLACE functions.

**DSAC**
Specify the data set authorization class to be verified when the resource is accessed at the node. You determine these credentials according to DPPX rules, specifying four hexadecimal digits. Do not specify X’FFFF’.

**FCTLNAME**
Specify the file control name under which the data to be transmitted is stored at the host. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic. The default file control name is the node, or group of nodes, in the preceding DEFINE PHASE statement.
If you only specify a value for FCTLNAME, all of the data sets queued under the value for FCTLNAME will be transmitted. If you also specify a value for ORIGIN, all of the data sets that are queued under queue-name and which have the specified ORIGIN name will be sent. If you specify a value for ORIGIN and a data set name (values for NAME or GROUPID), only the data sets indicated will be transmitted.

**FILEPSW**
Specify the password required for accessing the file where the data set is stored at the node.

**GROUPID**
Specify the name of a group of resources to be sent. You can specify from 1 to 8 alphanumeric characters. The first character must be alphabetic.

**LIBNAME**
Specify the name of the library to which the resource belongs. This applies to resource types other than library.

**LMTYPE**
Specify the library type:
- **C** Core image library
- **P** Procedure library
- **R** Relocatable library
- **S** Source library

For member resource types, you can specify up to eight alphanumeric characters.

**NAME**
Specify the name of the resource to be sent. If you do not specify a name, all resources queued under the values indicated for FCTLNAME and ORIGIN will be sent.

**Note:** If you specify a library type of **S**, you must also specify a sublibrary name in the SUBLIB operand.

**MOD**
Specify the modification level and version number of the resource.

**VER**
Specify the resynchronization level to be applied, if necessary, during transmission of the resource.

**Note:** If you specify RESYNC=YES, the resource will not be “released” if an interruption occurs. This means you cannot update the resource (delete, convert, replace), because it is marked in use.
SCRATCH  Specify whether the data sets are to be deleted from the node after printing. NetView DM for MVS will not monitor the actual printing or deleting of the data set.

SHR  This operand is valid for CLISTs only. It is ignored for panels or programs, as they only have the SHR options given to them when they were retrieved. Specify the types of concurrent access to be allowed at the node.

SUBCATE  Specify the subcategory (from 0 to 16). Specifying 0 sends all subcategories of the related category.

SUBLIB  Specify the sublibrary name.

Note: If the library type is S (LTYPE=S), you must specify a sublibrary name.

VOLSER  Specify the volume serial number (from 1 to 6 alphanumeric characters) of the physical volume on which the resource is to be stored.

BYDESTID  This operand lets you send a different data set to each node in a group of nodes, while referring to them using the same NAME=resource-name operand. You can only use this operand with the SEND DATASET and REPLACE DATASET transmission functions. The default is BYDESTID=NO.

If you specify BYDESTID=YES, then you must specify a value for either NAME or GROUPID, but not both. Do not specify a value for FCTLNAME, because when you specify BYDESTID=YES, NetView DM for MVS assumes that the resources which belong to each node in the group are to be found in the file control name that has the same name as the node.

Note: This operand is not allowed for nodes with an intermediate connection.

Example:

Here is a plan which consists of one phase that addresses one node group named GROUP1, consisting of two nodes (DEST1 and DEST2):

```
DEFINE PLAN  NAME=PLAN1
DEFINE PHASE NAME=PHASE1, GROUP=GROUP1
SEND DATA  NAME=DATASET1, BYDESTID=YES
```

Here is the plan without using BYDESTID, which is equivalent to the plan shown above.

```
DEFINE PLAN  NAME=PLAN1
DEFINE PHASE NAME=PHASE1, NODE=DEST1
SEND DATA  NAME=DATASET1, ORIGIN=DSX, FCTLNAME=DEST1

DEFINE PHASE NAME=PHASE2, NODE=DEST2
SEND DATA  NAME=DATASET1, ORIGIN=DSX, FCTLNAME=DEST2
```

The resources to be sent or replaced using BYDESTID must be prepared by specifying:

NAME  This is the common name you give to all the resources (see Appendix C, “Resource Naming Conventions” on page 569).
**FCTLNAME**  This is the name of the node (from 1 to 8 characters) to which the resource must be sent.

NetView DM for MVS does not check whether the resource is present in the resource repository, at the time the plan is validated or submitted.

**Example:**  Here is an example of a statement to replace a data set at the node called EDX1:

```
REP DATA NAME=DS08ED1,
    FCTLNAME=EDX1
```
SUBMIT Transmission Functions for LU 6.2 Node Types

This section primarily describes the transmission functions that you can use with LU 6.2 node types (listed in Table 71 on page 562). The syntax and operands of these statements are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

Some of these functions can be used with all node types, as shown in Table 37.

Functions as they apply only to LU 6.2 resources are shown in Table 38 on page 344.

Transmission Functions and Node Types: Table 37 lists the transmission functions (control statements) that apply primarily to LU 6.2 node types, and the specific node types for each transmission function:

<table>
<thead>
<tr>
<th>Transmission Function</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept Microcode</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Accept Resource</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Activate Node</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Delete Microcode</td>
<td>CMEP or CMFP</td>
</tr>
<tr>
<td>Delete Resource</td>
<td>CMEP, CMFP and NDMT</td>
</tr>
<tr>
<td>Initiate Procedure</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Install Microcode</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Install Resource</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Remove Microcode</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Remove Resource</td>
<td>CMEP only</td>
</tr>
<tr>
<td>Retrieve Microcode</td>
<td>CMEP or CMFP</td>
</tr>
<tr>
<td>Retrieve Resource</td>
<td>CMEP, CMFP and NDMT</td>
</tr>
<tr>
<td>Send Microcode</td>
<td>CMEP or CMFP</td>
</tr>
<tr>
<td>Send Resource</td>
<td>CMEP, CMFP and NDMT</td>
</tr>
<tr>
<td>Uninstall Component</td>
<td>CMEP only</td>
</tr>
</tbody>
</table>

Table 37. Transmission functions for node types
Table 38. Functions you can use with LU 6.2 resources

<table>
<thead>
<tr>
<th>Resources</th>
<th>Accept (1)</th>
<th>Delete</th>
<th>Install</th>
<th>Remove (1)</th>
<th>Retrieve</th>
<th>Send</th>
<th>Initiate</th>
<th>Uninstall</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS/400 object</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration File</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flatdata</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Dump</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error log</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microcode</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Procedure</td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relational Data</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Software</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Trace Info</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Accept and Remove are only applicable if the resource is installed as removable.
2. Procedures can also be transmitted within a "change file," and installed as software.

Reports: After a request has been completely processed, the entry point sends a report that indicates whether the execution was successful or not. If the execution was successful, the entry point sends a completion report which contains a completion code. This is translated to a function return code of 0, for the purpose of using conditional plans and phase processing.

If the execution was unsuccessful, the entry point sends an exception report indicating the cause of the failure with an SNA Condition Report. This is translated to a function return code of 8, for the purpose of using conditional plans and phase processing.

Dynamic Substitution: Of special interest is the ability to perform dynamic substitution with several of the transmission functions. With the operand BYDESTID, you can decide to exchange the tokens for NETID and NETLU for their actual tokens when the transmission plan is run.

The following example illustrates this. In this example, you perform dynamic substitution for a function addressing a group of nodes (GROUP1), including these two nodes:

- **PSNXNOD1**
  - RGN=PSNXRGN1
  - REN=PSNXREN1
- **PSNXNOD2**
  - RGN=PSNXRGN2
  - REN=PSNXREN2
Here is an example of the control statement you would use:

```
DEF PHAS NAME=PHAS01, GROUP=GROUP1, STATUS=RELEASE
INSTALL RESOURCE NAME=IBM.NETID.1.PSNX.NETLU.PSNXREF.NETLU
   RESTYPE=0070,
   ACTUSE=P,
   SEND=YES,
   DISPOSITION=KEEP,
   POSTTEST=YES,
   REMOVABILITY=YES,
   AUTOACCEPT=YES,
   AUTOACCEPT=YES,
   ALTERTIVE=A,
   BYDESTID=Y
```

The actual resource sent and installed for the node PSNXNOD1 is:

```
IBM.PSNXRGN1.1.PSNX.PSNXREF.PSNXREN1
```

The actual resource sent and installed for the node PSNXNOD2 is:

```
IBM.PSNXRGN2.1.PSNX.PSNXREF.PSNXREN2
```
You can use the ACCEPT MICROCODE statement at a 3174 CMEP node type, to request a node to permanently install the microcode or microcode customization data previously sent to the node (specified in the preceding DEFINE PHASE statement), and installed in production removable state.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT MICROcode</td>
<td>NAME=microwcode-name</td>
<td>CMEP nodes only</td>
</tr>
<tr>
<td></td>
<td>GROUPID=microwcode-group-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[TRACK=Y[ES]</td>
<td>N[O]]</td>
</tr>
</tbody>
</table>

**NAME**

Specify the name of the microcode, or microcode customization data. You can specify up to 64 characters. You must fully qualify the names of resources, although you can dynamically substitute the NETLU and NETID values, as described below.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**GROUPID**

Specify the microcode group name. You can specify up to 8 alphanumeric characters. The first character must be alphabetic.

**BYDESTID**

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced.

BYDESTID is valid only if NETID and NETLU are not specified.

**NETLU**

Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**

Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

**TRACK**

Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro. You can also define this value when you define the actual resource.

**Example:** Here is an example of an ACCEPT MICROCODE statement:

```
ACC MICR NAME=MCUST.3174.NA.*.*.CFG.LOCALCHG.NEW,
            NETLU=5,
            NETID=4
```
## ACCEPT RESOURCE

You can use the ACCEPT RESOURCE statement at a CMEP node type, to request a node to permanently install a resource previously sent to the node (specified in the preceding DEFINE PHASE statement), and installed in production removable state.

### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT RESOURCE</td>
<td>NAME=resource-name</td>
<td>CMEP nodes only (LU 6.2)</td>
</tr>
<tr>
<td></td>
<td>.GROUPID=resource-group-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>.RESTYPE=resource-type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.BYDESTID=YES</td>
<td>NO]</td>
</tr>
<tr>
<td></td>
<td>[.NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TRACK=YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>

#### NAME

Specify the resource name. You can specify up to 64 characters. You cannot use partial matching for this operand. You must fully qualify the names of resources, although you can dynamically substitute the NETLU and NETID values, as described below.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

#### GROUPID

Specify the resource group name. You can specify up to 8 alphanumeric characters. The first character must be alphabetic.

#### RESTYPE

Specify the resource type name or type code, as described in “NetView DM for MVS Resource Types” on page 566, but only for CMEP resource types. This operand is mandatory for all resource types apart from microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.

#### BYDESTID

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU”. If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced.

BYDESTID is valid only if NETID and NETLU are not specified.

#### NETLU

Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

#### NETID

Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

#### TRACK

Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.
Example: Here is an example of an ACCEPT RESOURCE statement:

```
ACCEPT RESOURCE NAME=IBM.1234567.13.FIX.1.PROB1,
    RESTYPE=0070,
    TRACK=YES
```
ACTIVATE NODE

Use the ACTIVATE NODE statement to first shutdown and then reactivate a node (specified in the preceding DEFINE PHASE statement), by causing a node IML or IPL. The resource is moved from the trial area to the active area of the node during the shutdown process, and the node is restarted. The IML or IPL activates the node components altered by the installation of change files.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTivate NODE</td>
<td>FORCE=[Y</td>
<td>YES]</td>
</tr>
<tr>
<td></td>
<td>[DATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[TIME=hhmm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[CMACTUSE=P</td>
<td>T]</td>
</tr>
<tr>
<td></td>
<td>[TIMEIND=G[MT]</td>
<td>L[OCAL]]</td>
</tr>
</tbody>
</table>

FORCE Specify whether you want the node to proceed with the activation, based on its quiescent state. Specify Y if you want to continue. If the node is an AS/400 node, you must always specify FORCE=YES.

Specify N if you do not want to continue. The node will reject the activation request. If you specify FORCE=NO, the SNA Report Code detected when addressing a NetView DM/2 node should be 100B000C. Refer to Messages and Codes for a description of the SNA Registered Sense codes generated by NetView DM for MVS.

DATE TIME Specify the date and time when you want to activate the node. You must specify both the date and the time. If you do not specify any values, the default is for the request to be executed immediately.

CMACTUSE Specify which type of components altered by changes that you want to be used during the activation. You can specify:

T (Trial-and-production). The components altered by changes installed on trial will be used. These will supersede the components installed in production, and also the unchanged components.

Notes:

1. You cannot specify CMACTUSE=TRIAL if you are using an AS/400 system running OS/400 and SystemView Managed System Services/400. This option is not supported by the AS/400 system.

2. If you specify CMACTUSE=TRIAL, then the SNA Report Code detected when addressing a NetView DM/2 node should be 100B000C. Refer to Messages and Codes for a description of the SNA Registered Sense codes generated by NetView DM for MVS.

P (Production only). The components altered by changes installed in production will be used, but no changes will be installed on trial.

TIMEIND Specify if the date and time (DATE and TIME) that you specified for the activation at the node is expressed in local time at the node, or in Greenwich Mean Time (GMT).
Example: This is an example of an ACTIVATE NODE statement:

```
ACTIVATE NODE CMACTUSE=P,
    DATE=112390,
    TIME=1320,
    TIMEIND=L
```
DELETE MICROCODE

Use the DELETE MICROCODE statement to delete microcode or microcode customization data from the node, or group of nodes, as specified in the preceding DEFINE PHASE statement. The microcode that you specify must already have been sent to the node, but it must not have been installed at the node.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete MICROcode</td>
<td>NAME= microcode-name</td>
<td>CMEP and CMFP nodes only</td>
</tr>
<tr>
<td></td>
<td>[GROUPID= microcode-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.BYDESTID=Y[ES][NO]]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L,I,H),...)]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TRACK=Y[ES][NO]]</td>
<td></td>
</tr>
</tbody>
</table>

NAME

Specify the name of the microcode, or microcode customization data. You can specify up to 64 characters, either fully or partially qualified.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

GROUPID

Specify the microcode group name. You can specify up to 8 alphanumeric characters. The first character must be alphabetic.

BYDESTID

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU”. If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

MATCHIND

Specify the list of matching indicators to be used by the host for deleting microcode at the node. The values of the MATCHIND parameter can be:

- n: The qualifier ordinal number that the matching indicator refers to.
- L: Set low value.
- I: The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
- H: Set high value.

The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

NETLU

Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

NETID

Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.
DELETE MICROCODE

TRACK Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.

Example: This is an example of a DELETE MICROCODE statement:

```
DEL MICR NAME=MCUST.3174.NA.*.*.CFG.LOCALCHG.*, 
    NETLU=5, 
    NETID=4
```
DELETE RESOURCE (from node)

Use the DELETE RESOURCE statement to delete a resource from the node specified in the preceding DEFINE PHASE statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELETE RESOURCE</td>
<td>RESTYPE=resource-type</td>
<td>NDMT, CMEP and CMFP node types</td>
</tr>
<tr>
<td></td>
<td>[BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[NAME=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[GROUPID=resource-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[EXT=0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>[FCTLNAME=resource-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[KEYNUMB=key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[LIBNAME=library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[LMTYPE=library/member-type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[MEMBER=member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[ORIGIN=resource-origin]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[SUBLIB=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[MATCHIND=((n,L</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>[NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[TRACK=Y[ES]</td>
<td>N[O]]</td>
</tr>
</tbody>
</table>

Note: Table 39 on page 355 shows which operands are required, and which are optional.

**RESTYPE**

Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory for all resource types apart from microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.

You can specify LU 6.2 resource types for nodes with CMEP or CMFP functional capabilities.

For nodes with NDMT functional capabilities, you can only specify LU0 resource types.

**BYDESTID**

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU”. If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

**NAME**

Specify the name of the resource that you want to delete. You can specify up to 64 characters, either fully or partially qualified.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**GROUPID**

Specify the name of the group of resources. You can specify up to eight alphanumeric characters. The first character must be alphabetic. If you specify GROUPID, you must also specify RESTYPE. Depending upon the resource type you specify, you must also specify the FCTLNAME and ORIGIN operands. Refer to Table 39 on page 355.
DELETE RESOURCE

EXT
This operand applies to DSCB resources only. Specify an extent (0 or 1), or a key number in the KEYNUMB operand, but not both. If you do not specify an extent or a key number, the default is EXT=0.

FCTLNAME
Specify the name that the resources are queued under in the target repository. You can specify up to 8 characters. The first character must be alphabetic. Refer to Table 39 on page 355.

KEYNUMB
This operand applies to DSCB resources only. Specify a key number (from 1 to 8), or an extent in the EXT operand, but not both. If you do not specify a key number or an extent, the default is EXT=0.

LIBNAME
Specify the library name.

LMTYPE
Specify the library type:
C  Core image library
P  Procedure library
R  Relocatable library
S  Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

MEMBER
Specify the member name.

MOD
Specify the modification level. This is an integer from 0 to 99. You only use this for resources that are stored in the NetView DM for MVS library. MOD only applies to certain resource types (see Table 39 on page 355). If it applies to the resource type you are deleting, the default is 0. If MOD does not apply, NetView DM for MVS sets the value to blanks.

ORIGIN
Specify the name of the resource origin. Depending on the resource type you specify, you must also specify ORIGIN. There is no default.

SUBCAT
Specify the subcategory (from 1 to 16). If you do not enter a value, NetView DM for MVS assumes a blank.

SUBLIB
Specify the sublibrary name. The default is to delete the entire library.

Note: If the library type is S (LMTYPE=S), you must specify a sublibrary name.

VER
Specify the version level of the resource. This is an integer from 0 to 99. You only use this operand for resources that are stored in the NetView DM for MVS library. If VER applies to the resource type you are deleting (see Table 39 on page 355), the default is 0. If VER does not apply, NetView DM for MVS sets the value to blanks.

MATCHIND
Specify the list of matching indicators to be used by the host to delete a resource at the node. The values of the MATCHIND parameter can be:

n  The qualifier ordinal number that the matching indicator refers to.
L  Set low value.
I  The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
H  Set high value.
The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

**NETLU**
Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**
Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

**TRACK**
Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.

**Usage:** Table 39 shows which operands are required, and which are optional, for the DELETE RESOURCE control statement.

---

**Table 39 (Page 1 of 2). Applicability of DELETE RESOURCE operands**

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object</td>
<td>0158</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>containment structure</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>RESTYPE, NAME or GROUPID</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>RESTYPE, NAME or GROUPID</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>RESTYPE, NAME or GROUPID</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB, MEMBER</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>RESTYPE, FCTLNAME</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>NAME or GROUPID</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
</tbody>
</table>

---

Chapter 16. Preparing and Submitting Transmission Plans 355
<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td></td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td></td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td></td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td></td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>RESTYPE, NAME or GROUPID</td>
<td>MATCHIND, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Transaction</td>
<td>0026</td>
<td>RESTYPE, ORIGIN, FCTLNAME</td>
<td></td>
</tr>
</tbody>
</table>
INITIATE PROCEDURE

Use the INITIATE PROCEDURE statement to request the execution of a command procedure at an entry point node. After the request has been completely processed, the entry point sends a report that indicates whether the execution was successful or not. If the execution was successful, the entry point sends a completion report which contains a completion code. This is translated to a function return code of 0, for the purpose of using conditional plans and phase processing.

If the execution was unsuccessful, the entry point sends an exception report indicating the cause of the failure with an SNA Condition Report. This is translated to a function return code of 8, for the purpose of using conditional plans and phase processing.

These reports are all stored in the TCF.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INItiate PROCedure</td>
<td>NAME=resource-name</td>
<td>CMEP nodes only</td>
</tr>
<tr>
<td></td>
<td>[.BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[.SEND=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L)</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>[.NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.PROCDATA=' parameter-area'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TIME=hhmm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TIMEIND=LOCAL</td>
<td>GMT]</td>
</tr>
<tr>
<td></td>
<td>[.TRACK=Y[ES]</td>
<td>N[O]]</td>
</tr>
</tbody>
</table>

NAME Specify the name of the procedure that you want to execute at the node. You can specify up to 64 characters. You must fully qualify the name if you specify SEND=NO in the operand described below. You can partially qualify the name if you specify SEND=YES.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

Note: Resource names starting with $DELETE,$PENDING are reserved, and may not be used.

BYDESTID Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” BYDESTID is valid only if NETID and NETLU are not specified.

SEND Specify either that you want NetView DM for MVS to send the procedure to the node and then execute it, or that you do not want send the procedure, as the procedure is at the node and NetView DM for MVS only has to execute it. The values you can specify are:

Y Send and execute the procedure.

N Do not send the procedure; just execute it.

If you specify SEND=YES and a procedure with the same name already exists at the node, NetView DM for MVS overwrites it. The procedure is not deleted at the node after execution.
INITIATE PROCEDURE

MATCHIND  This operand is valid only in combination with SEND=YES. Specify a list of matching indicators to be used by the host for identifying the procedure to be sent and executed. The values of the MATCHIND parameter can be:

n  The qualifier ordinal number that the matching indicator refers to.
L  Set low value.
I  The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
H  Set high value.

The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

NETID  NETLU  Indicate that the network ID (NETID) and network LU (NETLU) name of the target node are to be inserted in the name at the specified positions. You must specify the two operands together, and the corresponding tokens in the name must be specified as "*," without matching indicators.

PROCDATA  Specify up to 60 characters of information related to the procedure that will be started at the node (for example, parameters required to run the procedure).

DATE  TIME  Specify the date and time when you want the procedure to be initiated at the node. You must specify both the date and the time. If you do not specify any values, the default is immediate execution.

TIMEIND  Specify if the date and time (DATE and TIME) that you specified for the execution of the command at the node is expressed in local time at the node, or in Greenwich Mean Time (GMT).

TRACK  Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro. You can overwrite this initial value. Specify Y (Yes) for unconditional tracking. Specify N (No) for no tracking.

Example:  Here is an example of a transmission plan definition with an INITIATE PROCEDURE using the PROCDATA operand:

```
DEFINE PLAN NAME=PLAN1
DEFINE PHASE NAME=PHASE01,NODE=NODE1
  INI PROC NAME=PROC1,
    PROCDATA='RECEIVE ULM/zerodot3 Y:/BTXFIX4/zerodot.126 DBPT.BTX.FIX.X4/zerodot.126'
  END
END
```

Note:  The first phase, which uses PROCDATA, passes a parameter area, while the procedure in the second phase does not require any parameter area.
INSTALL MICROCODE

Use the INSTALL MICROCODE statement to install microcode or microcode customization data at a node, specified in the preceding DEFINE PHASE statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTALL MICROCODE</td>
<td>NAME=microcode-name</td>
<td>GROUPID=microcode-group-name</td>
</tr>
<tr>
<td></td>
<td>[AUTOACCEPT=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[ACTUSE=TRIAL</td>
<td>[PRODUCTION]]</td>
</tr>
<tr>
<td></td>
<td>[COREQUISITE=microcode-group-name]</td>
<td>[COREQUISITE=microcode-group-name]</td>
</tr>
<tr>
<td></td>
<td>[DELMATCHIND=</td>
<td>(n,L]</td>
</tr>
<tr>
<td></td>
<td>[DELNAME=microcode-delete-name]</td>
<td>[DELNAME=microcode-delete-name]</td>
</tr>
<tr>
<td></td>
<td>[DESTRUCTION=</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>[MATCHIND=</td>
<td>(n,L]</td>
</tr>
<tr>
<td></td>
<td>[POSTTEST=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[PRETEST=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[REMOVABILITY=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[SEND=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[DATE=mmddyy]</td>
<td>[DATE=mmddyy]</td>
</tr>
<tr>
<td></td>
<td>[TIME=hhmm]</td>
<td>[TIME=hhmm]</td>
</tr>
<tr>
<td></td>
<td>[NETLU=token-number]</td>
<td>[NETLU=token-number]</td>
</tr>
<tr>
<td></td>
<td>[NETID=token-number]</td>
<td>[NETID=token-number]</td>
</tr>
<tr>
<td></td>
<td>[DISPOSITION=K[EEP]</td>
<td>D[DELETE]]</td>
</tr>
<tr>
<td></td>
<td>[TIMEIND=G[MT]</td>
<td>L[OCAL])</td>
</tr>
<tr>
<td></td>
<td>[TRACK=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[EXTENDFILE=Y[ES]</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>[FORCEIND=Y[ES]</td>
<td>D</td>
</tr>
</tbody>
</table>

**NAME**

Specify the name of the microcode to be installed at the node. You can specify up to 64 characters. If you specify **SEND=NO**, you must fully qualify the name, unless you are using the NETLU and the NETID operands. If you do specify these values, the two tokens you specify will be substituted by the related values. If you specify **SEND=YES**, you can partially qualify the name.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**GROUPID**

Specify the name of the group of microcode files to be installed at the node. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. Do not specify the COREQUISITE operand if you specify GROUPID.

**AUTOACCEPT**

Specify whether you want the microcode to be automatically accepted after it has been successfully installed and tested. Automatic acceptance frees the node resources that are used to provide the accept option. The AUTOACCEPT values can be:

- **Y** Yes (Accept)
- **D** Desired (Accept if possible)
- **N** No (Do not accept)
INSTALL MICROCODE

Note:

- If `REMOVABILITY=NO`, and you do not specify a value for `AUTOACCEPT`, the default for `AUTOACCEPT` is **NONE**.
- If `REMOVABILITY=YES`, and you do not specify a value for `AUTOACCEPT`, the default for `AUTOACCEPT` is **NO**.

**AUTOREMOVAL**

Specify whether you want the microcode to be automatically removed if it is unsuccessfully installed and tested. Automatic removal frees the node resources that are used to provide the removal option. The **AUTOREMOVAL** values can be:

- **Y** Yes (Remove)
- **D** Desired (Remove if possible)
- **N** No (Do not remove)

If you specify `REMOVABILITY=NO`, do not specify a value for `AUTOREMOVAL`, as the default value is **NONE**. If, however, you specify `REMOVABILITY=YES` and you do not specify a value for `AUTOREMOVAL`, the default value is **NO**.

**ACTUSE**

Specify whether the installed microcode can be used for trial or production purposes. The **ACTUSE** values can be:

- **T** Trial (Trial purposes only)
- **P** Production (Trial or production purposes)

**COREQUISITE**

Specify the name of a group of microcode resources that are corequisites of the microcode to be installed. The node must verify the availability of the corequisite resources. You must already have defined the group in the DRD. If any of the corequisite microcode resources are unavailable, the microcode installation request will fail.

You cannot have more than six elements in the group. You can send a seventh element by using the **SEND=** option.

If the corequisites have to be installed at the node in a specified order, then you must specify the resources in the same order when you define the resource group (using either GIX or the SUBMIT batch utility). The order that you define in the group cannot be changed. If you need to change the order, you must delete the whole group and redefine it with the new order.

**DELMATCHIND**

Specify the list of matching indicators used by the node to identify the microcode to be deleted. The **DELMATCHIND** values can be:

- **n** The qualifier ordinal number that the matching indicator refers to
- **L** Set low value
- **I** The relative token value is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
- **H** Set high value

You can only use the **DELMATCHIND** operand when you also specify **DESTRUCTION=ALLOWED**.
DELNAME Specify the name of the microcode to be deleted at the node. The name can be partially qualified. This is valid only if you also specify DESTRUCTION=ALLOWED and NAME. If you do not specify DELNAME, the current value of the NAME operand, or each corresponding name of the group, is kept.

DESTRUCTION Specify whether or not, if there is a lack of storage space, the microcode to be installed will be allowed to overwrite microcode already stored at the node. The DESTRUCTION values can be:

A This means that overwriting is allowed.
N This means that overwriting is not allowed. This is the default value.

The DESTRUCTION=ALLOWED option is valid only if you also specify SEND=YES. The DESTRUCTION operand is valid only if you also specify the NAME operand.

MATCHIND Specify the list of matching indicators to be used by the host for sending microcode to the node. The values of the MATCHIND parameter can be:

n The qualifier ordinal number that the matching indicator refers to.
L Set low value.
I The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
H Set high value.

The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

POSTTEST Specify whether the microcode is to be tested after installation. The POSTTEST values can be:

Y Yes (Test after installation)
D Desired (Test after installation if possible)
N No (Do not test after installation)

PRETEST Specify whether the microcode is to be tested before installation. The POSTTEST values can be:

Y Yes (Test before installation)
D Desired (Test before installation if possible)
N No (Do not test before installation)

REMOVABILITY Specify whether the microcode is to be installed removably. Removable installation means that the microcode you are replacing can be restored. If you choose not to install the microcode removably, you cannot restore the microcode even if the installed and tested microcode is unsuccessful. The REMOVABILITY values can be:

Y Yes (Install removably)
D Desired (Install removably if possible)
N No (Do not install removably)

If you specify ACTUSE=TRIAL you must also specify REMOVABILITY=YES.
INSTALL MICROCODE

SEND Specify whether the microcode is to be sent together with the install command. The SEND values can be:

Y Yes (Microcode is retrieved from the resource repository and sent to the node with the install command.)
N No (Microcode is not retrieved, and only the install command is sent to the node.)

DATE TIME Specify the date and time when you want to install the microcode at the node. You must specify both the date and the time. If you do not specify any values, the default is immediate execution.

NETLU Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

NETID Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

ALTERACTIVE Specify whether the target node can apply the installed changes to the components in the active system, or if a subsequent activation is required to apply the installed changes.

If you specify ALTERACTIVE=ALLOWED, the node will determine whether it is necessary for the node components to be activated in order for the microcode to be installed.

If you specify ALTERACTIVE=NO, the node cannot activate the affected components during the install process and they will be activated only when the node is reactivated. The node is reactivated with an ACTIVATE NODE command from NetView DM for MVS, or with a re-IPL or re-IML from the node operator.

If you do not enter a value for this operand, NetView DM for MVS does not send either of the settings to the entry point.

DISPOSITION Specify whether you want to delete change files after successful installation, or keep them for redistribution. If you choose the delete option, this has no effect if the installation fails.

If you do not specify a value for this operand, NetView DM for MVS does not send either of the settings to the entry point.

TIMEIND Specify if the date and time (DATE and TIME) that you specified for the installation at the node is expressed in local time at the node, or in Greenwich Mean Time (GMT).

TRACK Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.
EXTENDFILE  Specify whether you want the journaled file system at the target NetView DM/6000 or NetView DM for NetWare node to be extended if additional disk space is required for installation. The values of the EXTENDFILE parameter can be:

Y  Yes (Check whether there is enough space. If not, extend the journaled file system. If the information necessary to make the check is not available, terminate the installation.)
D  Desired (Check whether there is enough space. If not, extend the journaled file system. If the information necessary to make the check is not available, proceed with the installation.)
N  No (Do not make the check.)

FORCEIND  Specify whether you want to force the installation at the target NetView DM/6000 or NetView DM for NetWare node if the change file history at the target node forbids it. The values of the FORCEIND parameter can be:

Y  Yes (Force)
D  Desired (Force if possible)
N  No (Do not force)

Example:  This is an example of an INSTALL MICROCODE statement:

INS MICR NAME=MCUST.3174.NA.***.CFG.LOCALCHG,
  SEND=N,
  NETLU=5,
  NETID=4,
  REMOVABILITY=Y,
  AUTOREMOVAL=Y,
  POSTTEST=N,
  ACTUSE=P
INSTALL RESOURCE

Use the INSTALL RESOURCE statement to install resources at a node.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CMEP nodes only</td>
</tr>
</tbody>
</table>

NAME Specify the name of the resource to be installed at the node. You can specify up to 64 characters. If you specify SEND=NO, you must fully qualify the name, unless you are using the NETLU and the NETID operands. If you do specify these values, the two tokens you specify will be substituted by the related values. If you specify SEND=YES, you can partially qualify the name.

Note: You can specify the name of a single resource in the NAME operand, or the name of a group of resources in the GROUPID operand, but not both.

GROUPID Specify the name of the group of resources to be installed at the node. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. Do not specify the COREQUISITE operand if you specify GROUPID.

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory for all resource types apart from microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.
Notes:

1. You can only specify LU 6.2 resource types for this control statement because it is valid for CMEP nodes only.

2. AS/400 CMEP nodes do not support the Install Resource function for flat data resource types.

BYDESTID

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID or COREQUISITE is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

DASDCHECK

Specify whether you want the target node to check that the required DASD space is available before performing the Install function.

Note: This parameter is supported by NetView DM/2 Version 2 Release 1 target workstations only. NetView DM/6000 and NetView DM for NetWare products check the disk space automatically. Specify a value for the DASDCHECK parameter only if the resource is to be installed on a NetView DM/2 Version 2 Release 1 workstation.

You can specify one of the following values:

Y  This means that you want to perform the check. To successfully check the DASD space, you must have defined either one of the following:

- The processing size of the resource to be installed and stored in the catalog. This is specified in the PROCESIZE operand of the LOAD RESOURCE statement.
- The user-defined processing size of the resource to be installed and stored in the catalog. This is specified in the UDPROCESIZE operand of the Install request.

If the information needed to perform the check is not available, the function terminates unsuccessfully.

If the target node checks the DASD space and discovers that there is not enough space, the function is not performed and terminates unsuccessfully.

D  This means that you want to check the DASD space only if it is possible. If the information needed to perform the check is not available, the check is not performed but the function is still executed.

N  This means that you do not want to check the DASD space. No check is performed on the DASD space that is available.

If you do not enter a value for this operand, NetView DM for MVS does not send any of the settings to the entry point. The target node determines the default value.

UDPROCESIZE

Specify a user-defined processing size if you want to override the value that was specified in the PROCESIZE operand of the LOAD RESOURCE statement when the resource was loaded into the target catalog. Specify the size in kilobytes of the DASD space required on the target installation drive to install the resource. Enter an integer from 1 to 999 999.

Note: This parameter is supported by NetView DM/2 Version 2 Release 1 target workstations only.

You can only specify a user-defined processing size if you specified DASDCHECK=Y or DASDCHECK=D.
INSTALL RESOURCE

If you also specify a value for the COREQUISITES operand, the processing size for each change object that belongs to the corequisites group must be available in the target catalog before the install request is received. You cannot override the processing size of the corequisite change objects.

Attention

Do not specify a user-defined processing size in either of the following circumstances:

- With a resource name that is partially qualified. The result may not be what you expected, because the specified size is assigned to any resource that matches the partial name. This could be a different resource than the one you expected, or it could be multiple resources, if you specified the NETID and NETLU operands in the partial name.
- If you specified GROUPID rather than a single resource name.

AUTOACCEPT

Specify whether you want the resource to be automatically accepted after it has been successfully installed and tested. Automatic acceptance frees the node resources that are used to provide the accept option. The AUTOACCEPT values can be:

- **Y** Yes (Accept)
- **D** Desired (accept if possible)
- **N** No (do not accept)

Notes:

1. Do not specify AUTOACCEPT if REMOVABILITY=NO.
2. If you specify ACTUSE=TRIAL, you must also specify AUTOACCEPT=NO.
3. If REMOVABILITY=NO, and you do not specify a value for AUTOACCEPT, the default for AUTOACCEPT is NONE.
4. If REMOVABILITY=YES, and you do not specify a value for AUTOACCEPT, the default for AUTOACCEPT is NO.

AUTOREMOVAL

Specify whether you want the resource to be automatically removed after it has been unsuccessfully installed and tested. Automatic removal frees the node resources that are used to provide the removal option. The AUTOREMOVAL values can be:

- **Y** Yes (Remove)
- **D** Desired (Remove if possible)
- **N** No (Do not remove)

Do not specify AUTOREMOVAL if you specify REMOVABILITY=NO.

Note:

- If REMOVABILITY=NO, and you do not specify a value for AUTOREMOVAL, the default for AUTOREMOVAL is NONE.
- If REMOVABILITY=YES, and you do not specify a value for AUTOREMOVAL, the default AUTOREMOVAL is NO.

ACTUSE

Specify whether the installed resource can be used for trial or production purposes. The ACTUSE values can be:

- **T** Trial (Trial purposes only)
- **P** Production (Trial or production purposes).
**COREQUISITE** Specify the name of a group of resources that are corequisites of the resource to be installed. The node must verify the availability of the corequisite resources. You must already have defined the group in the DRD. If any of the corequisite resources are unavailable, the resource installation request will fail.

You cannot have more than six elements in the group. You can send a seventh element by using the **SEND=**YES option.

If the corequisites have to be installed at the node in a specified order, you must specify the resources in the same order when you define the resource group (using either GIX or the SUBMIT batch utility). The order that you define in the group cannot be changed. If you need to change the order, you must delete the whole group and redefine it with the new order.

**DELMATCHIND** Specify the list of matching indicators used by the node to identify the resource to be deleted. The DELMATCHIND values can be:

- **n** The qualifier ordinal number that the matching indicator refers to.
- **L** Set low value.
- **I** The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
- **H** Set high value.

You can only use the DELMATCHIND operand when you also specify **DESTRUCTION=**A (allowed).

**DELNAME** Specify the name of the resource to be deleted at the node. The name can be partially qualified. This is valid only if you also specify **DESTRUCTION=**A (allowed) and NAME. If you do not specify DELNAME, the current value of the NAME operand, or each corresponding name of the group, is kept.

If you specify a value for DELNAME that is different from the NAME operand, NetView DM for MVS checks that all the tokens of the NAME operand whose token attribute is **MUST MATCH** are equal to the corresponding tokens of the DELNAME operand. If they are not, NetView DM for MVS ends the functions with return code 8 and issues the sense code 08A40001.

**DESTRUCTION** Specify whether or not, if there is a lack of storage space, the resource to be installed will be allowed to overwrite a resource already stored at the node. The DESTRUCTION values can be:

- **A** This means that overwriting is allowed.
- **N** This means that overwriting is not allowed. This is the default value.

The **DESTRUCTION=**A (allowed) option is valid only if you also specify **SEND=**YES. The DESTRUCTION operand is valid only when you also specify the NAME operand.

**MATCHIND** Specify the list of matching indicators to be used by the host for sending one or more resources, to the node. The values of the MATCHIND parameter can be:

- **n** The qualifier ordinal number that the matching indicator refers to.
- **L** Set low value.
The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.

Set high value.

The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

**POSTTEST**
Specify whether the resource is to be tested after installation. The POSTTEST values can be:

- Y Yes (Test after installation)
- D Desired (Test after installation if possible)
- N No (Do not test after installation)

**PRETEST**
Specify whether the resource is to be tested before installation. The PRETEST values can be:

- Y Yes (Test prior to installation)
- D Desired (Test prior to installation if possible)
- N No (Do not test prior to installation)

**REMOVABILITY**
Specify whether the resource is to be installed removably. Removable installation means that the resource you are replacing can be restored, if the installed and tested resource is unsuccessful. If you choose not to install the resource removably, you cannot restore the resource even if the installed and tested resource is unsuccessful. The REMOVABILITY values can be:

- Y Yes (Install removably)
- D Desired (Install removably if possible)
- N No (Do not install removably)

If you specify ACTUSE=TRIAL, you must also specify REMOVABILITY=YES.

**SEND**
Specify whether the resource is to be sent together with the install command. The SEND values can be:

- Y Yes (The resource is retrieved from the resource repository and sent to the node with the install command.)
- N No (The resource is not retrieved, and only the install command is sent to the node.)

**DATE**
Specify the date and time when you want to install the resource at the node. You must specify both the date and the time. If you do not specify any values, the default is immediate execution.

**NETLU**
Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**
Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.
ALTERACTIVE Specify whether the target node can apply the installed changes to the components in the active system, or if a subsequent activation is required to apply the installed changes.

If you specify ALTERACTIVE=ALLOWED, the node will determine whether it is necessary for the node components to be activated in order for the resource to be installed.

If you specify ALTERACTIVE=NO, the node cannot activate the affected components during the install process and they will be activated only when the node is reactivated. The node is reactivated with an ACTIVATE NODE command from NetView DM for MVS, or with a re-IPL or re-IML from the node operator.

If you do not enter a value for this operand, NetView DM for MVS does not send either of the settings to the entry point. The target node will determine the default.

DISPOSITION Specify whether you want to delete change files after successful installation, or keep them for redistribution. If you choose the delete option, this has no effect if the installation fails.

Note: DISPOSITION=DELETE is not supported by NetView DM/2 Version 2 nodes.

If you do not enter a value for this operand, NetView DM for MVS does not send either of the settings to the entry point. The target node will determine the default.

TIMEIND Specify if the date and time (DATE and TIME) that you specified for the installation at the node is expressed in local time at the node, or in Greenwich mean time (GMT).

TRACK Specify whether tracking is to be performed for this function. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.

EXTENDFILE Specify whether you want the journaled file system at the target NetView DM/6000 or NetView DM for NetWare node to be extended if additional disk space is required for installation. The values of the EXTENDFILE parameter can be:

- Y Yes (Check whether there is enough space. If not, extend the journaled file system. If the information necessary to make the check is not available, terminate the installation.)
- D Desired (Check whether there is enough space. If not, extend the journaled file system. If the information necessary to make the check is not available, proceed with the installation.)
- N No (Do not make the check.)

FORCEIND Specify whether you want to force the installation at the target NetView DM/6000 or NetView DM for NetWare node if the change file history at the target node forbids it. The values of the FORCEIND parameter can be:

- Y Yes (Force)
- D Desired (Force if possible)
- N No (Do not force)
Example: This is an example of an INSTALL RESOURCE statement:

```
INSTALL RESOURCE NAME=IBM.1234567.13.FIX.1.PROB1,
RETYPE=0070,
REMOVABILITY=Y,
ACTUSE=P,
POSTTEST=NO,
DATE=011891,
TIME=1100,
ALTERATIVE=N,
DISPOSITION=KEEP
```
# REMOVE MICROCODE

Use the REMOVE MICROCODE control statement to remove microcode or microcode customization data from the node, if the microcode was installed in a production removable state.

## Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOVE MICROcode</td>
<td>NAME= microcode-name</td>
<td>CMEP nodes only</td>
</tr>
<tr>
<td></td>
<td>GROUPID= microcode-group-name</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[,POSTTEST=Y[ES]</td>
<td>D[N[O]]</td>
</tr>
<tr>
<td></td>
<td>[,NETLU= token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,NETID= token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,ALTERACTIVE=ALLOWED]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[,TRACK=Y[ES]</td>
<td>N[O]]</td>
</tr>
</tbody>
</table>

**NAME**
Specify the name of the microcode you want to remove. You can specify up to 64 characters.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names. You cannot use partial naming for this operand. You must fully qualify the name of the microcode, although you can substitute the NETID and NETLU values dynamically, described below.

**GROUPID**
Specify the name of the group of microcode to be removed. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. The group of microcode must already be defined in the DRD.

**BYDESTID**
Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

**POSTTEST**
Specify whether you want to test the microcode that remains at the node after other microcode has been removed. The POSTTEST values can be:

- **Y** Yes (Test remaining microcode)
- **D** Desired (Test remaining microcode if possible)
- **N** No (Do not test remaining microcode)

**NETLU**
Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**
Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

**ALTERACTIVE**
Specify whether the node can reactivate the node, or node components, during the removal process.

If you specify **ALTERACTIVE=ALLOWED**, the node can reactivate the node or node components while the microcode is being removed.
If you specify **ALTERACTIVE=NO**, the node cannot proceed with a removal that requires node components to be activated before the resource can be removed.

You can reactivate a node with an ACTIVATE NODE command from NetView DM for MVS, or the node operator can perform a re-IPL or re-IML.

If you do not enter a value for this operand, NetView DM for MVS does not send either of the settings to the entry point.

**TRACK**

Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.

**Example:** Here is an example of a REMOVE MICROCODE statement:

```
REM MICR NAME=MCUST.3174.NA.*.*.CFG.LOCALCHG.NEW,
   NETLU=5,
   NETID=4
```
## REMOVE RESOURCE

Use the REMOVE RESOURCE statement to remove resources, or resource customization data, from a node. The resources must previously have been installed in a production removable state.

### Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
</table>

**NAME** Specify the name of the resource to be removed. You can specify up to 64 characters. You cannot use partial matching for this operand. You must fully qualify the names of resources, although you can dynamically substitute the NETLU and NETID values, as described below.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**GROUPID** Specify the name of the resource group to be removed. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. The group of resources must already be defined in the DRD.

**RESTYPE** Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory for all resource types apart from microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.

If you specify a name and a resource type, NetView DM for MVS checks that the resource type is compatible with the first token specified in the resource name.

**Note:** You should generally use the REMOVE MICROCODE statement for microcode, but the capability to remove microcode using REMOVE RESOURCE is provided for compatibility with previous releases of NetView DM for MVS.

**DASDCHECK** Specify whether you want the target node to check that the required DASD space available before performing the Remove function.

**Note:** This parameter is supported by NetView DM/2 Version 2 Release 1 target workstations only. NetView DM/6000 products check the disk space automatically. Specify a value for this parameter only if the resource is to be removed from a NetView DM/2 Version 2 Release 1 workstation.

You can specify one of the following values:

**Y.** Yes. This means that you want to perform the check. To successfully check the DASD space, you must have defined the processing size for each change object to be restored from the back-level and stored in the catalog.
If the information needed to perform the check is not available, the function terminates unsuccessfully.

If the target node checks the DASD space and discovers that there is not enough space, the function is not performed and terminates unsuccessfully.

D This means that you want to check the DASD space only if it is possible. If the information needed to perform the check is not available, the check is not performed but the function is still executed.

N This means that you do not want to check the DASD space. No check is performed on the DASD space that is available.

If you do not enter a value for this operand, NetView DM for MVS does not send any of the settings to the entry point. The target node determines the default value.

BYDESTID Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

POSTTEST Specifies whether to test the resources that remain at the node when the other resources have been removed. The POSTTEST values can be:

Y Yes (Test remaining resources)
D Desired (Test remaining resources if possible)
N No (Do not test remaining resources)

NETLU Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

NETID Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

ALTERACTIVE Specify whether the node can reactivate the node, or node components, during the removal process.

If you specify ALTERACTIVE=ALLOWED, the node can reactivate the node or node components while the resource is being removed.

If you specify ALTERACTIVE=NO, the node cannot proceed with a removal that requires node components to be activated before the resource can be removed.

You can reactivate a node with an ACTIVATE NODE command from NetView DM for MVS, or the node operator can perform a re-IPL or re-IML.

If you do not enter a value for this operand, NetView DM for MVS does not send either of the settings to the entry point.

TRACK Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.
**Example:** Here is an example of a REMOVE RESOURCE statement:

```
REMOVE RESOURCE NAME=IBM.1234567.13.FIX.1.PROB1,
  ALTERACTIVE=A,
  RESTYPE=0070
```
RETIEVE MICROCODE

Use the RETRIEVE MICROCODE statement to retrieve microcode or microcode customization data from a node.

Note: This statement assumes DISP=(NEW) automatically for the microcode.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve MICROcode</td>
<td>{NAME= microcode-name</td>
<td>GROUPID= microcode-group-name</td>
</tr>
<tr>
<td></td>
<td>[.NETLU= token-number]</td>
<td>[.NETID= token-number]</td>
</tr>
</tbody>
</table>

**NAME**

Specify the name of the microcode, or microcode customization data, to be retrieved from the node. You can specify up to 64 characters, fully or partially qualified.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**GROUPID**

Specify the name of the group of microcode to be retrieved from the node. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. The group of microcode must already be defined in the DRD.

**BYDESTID**

Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

**MATCHIND**

Specify the list of matching indicators to be used by the host to identify the microcode at the node. The values of the MATCHIND parameter can be:

- **n** The qualifier ordinal number the matching indicator refers to.
- **L** Set low value.
- **I** The relative matching indicator is ignored.
- **H** Set high value.

This operand is valid only if you use partial naming in the NAME operand. If you partially qualify the NAME and you do not enter a value for the MATCHIND operand, the default value is **H** (set high value).

**NETLU**

Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**

Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.
Example: This is an example of a RETRIEVE MICROCODE statement:

```
RET MICR NAME=MCUST.3174.NA.*.*.CFG.LOCALCHG.*,  
    NETLU=5,  
    NETID=4
```
Use the RETRIEVE RESOURCE statement to retrieve resources from a node, specified in the preceding DEFINE PHASE statement. You cannot track this function because the subsequent presence of the resource in the resource repository is the tracking.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RETrieve RESOurce</td>
<td>.RESTYPE=resource-type</td>
<td>NDMT, CMEP and CMFP node types</td>
</tr>
<tr>
<td></td>
<td>[.BYDESTID=Y[ES][NO]]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NAME=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.GROUPID=resource-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DISP={NEW</td>
<td>OLD}</td>
</tr>
<tr>
<td></td>
<td>[.EXT=0</td>
<td>1]</td>
</tr>
<tr>
<td></td>
<td>[.KEYNUMB=key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.FCTLNAME=resource-queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.GROUPID=resource-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.LIBNAME=library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.LMTYPE=library/member-type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MEMBER=resource-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.MOD=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.ORIGIN=resource-origin]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.SUBLIB=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.VER=0</td>
<td>nn]</td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L,I,H),...]]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.RESSTAT=ASIS</td>
<td>COMP</td>
</tr>
</tbody>
</table>

Note: Table 40 on page 380 shows which operands are required, and which are optional.

**RESTYPE** Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory for all resource types other than microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.

You can specify LU 6.2 resource types for nodes with CMEP or CMFP functional capabilities.

For nodes with NDMT functional capabilities, you can only specify LU0 resource types.

**BYDESTID** Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced. BYDESTID is valid only if NETID and NETLU are not specified.

**NAME** Specify the name of the resource to be retrieved from the node. The name can be fully or partially qualified. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.
GROUPID Specify the name of the group of resources that you want to retrieve from the node. You can specify up to 8 characters. The first character must be alphabetic. The group of resources must already be defined in the DRD. If you specify GROUPID, you must also specify RESTYPE. Depending on the RESTYPE you specify, you must also specify the FCTLNAME and ORIGIN operands.

DISP Specify how the retrieved resources are to be managed at the target repository. If you manage node types with NDMT functional capability, you can specify any of these values:

- NEW This means that the retrieved resources will be stored in the target repository only if they do not already exist in the target repository.
- OLD This means that the retrieved resources will be stored in the target repository, regardless of whether they already exist in the target repository.
- KEEP This means that the retrieved resources will be kept in the repository after they have been processed. You can specify KEEP for only data set, dump, and print resource types.
- DELETE This means that the retrieved resources will be deleted from the repository after they have been processed.

EXT This operand applies to DSCB resources only. Specify an extent (0 or 1), or a key number, but not both. If you do not specify an extent or a key number, the default is EXT=0.

KEYNUMB This operand applies to DSCB resources only. Specify a key number (from 1 to 8), or an extent with the EXT operand, but not both. If you do not specify a key number or an extent, the default is EXT=0.

FCTLNAME Specify the name that the resources are queued under in the repository. You can specify up to 8 characters. The first character must be alphabetic. Refer to Table 40 on page 380.

LIBNAME Specify a library name.

LMTYPE Specify the library type:

- C Core image library
- P Procedure library
- R Relocatable library
- S Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

MEMBER Specify the member name.

MOD Specify the modification level of the resource. This is an integer from 0 to 99. This operand only applies to resources that are stored in the NetView DM for MVS library. If MOD applies to the resource types that you are retrieving, the default is 0. If MOD does not apply to the resource types you are retrieving, NetView DM for MVS sets the value to blanks.

ORIGIN Specify the name of the resource origin.
RETRIEVE RESOURCE

SUBCAT Specify the subcategory (from 1 to 16). If you do not enter a value, NetView DM for MVS assumes a blank.

SUBLIB Specify the sublibrary name. The default is to retrieve the entire library.

Note: If the library type is S (LTYPE=S), you must specify a sublibrary name.

VER Specify the version level of the resource. This is an integer from 0 to 99. This operand only applies to resources that are stored in the NetView DM for MVS library. If VER applies to the resource types you are retrieving, the default is 0. If VER does not apply to the resource types you retrieving, NetView DM for MVS sets the value to blanks.

MATCHIND Specify the list of matching indicators to be used by the host for retrieving the resource from the node. The values of the MATCHIND parameter can be:

- n The qualifier ordinal number that the matching indicator refers to.
- L Set low value.
- I The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria it is assumed to be successful, whatever the value of the related token.
- H Set high value.

The MATCHIND operand is valid only if you also specify the NAME and SEND=YES operands. If the NAME operand is partially qualified, H default values are provided.

NETLU Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

NETID Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

RESSTAT Specify the status that the node must send the resource to the host, as a result of a RETRIEVE function.

- ASIS The node must send the resource as it is at the node itself. This is the default.
- COMP The node must send a compressed resource.
- NOTCOMP The node must send a non-compressed resource.

Only syntax errors are managed by NetView DM for MVS.

Usage: Table 40 shows which operands you can use with which resource types.

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td>0158</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
</tbody>
</table>
### Table 40 (Page 2 of 3). Applicability of RETRIEVE RESOURCE operands

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME, MEMBER</td>
<td>DISP</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>RESTYPE, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>NAME or GROUPID</td>
<td>RESTYPE, MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME, MEMBER</td>
<td>DISP</td>
</tr>
<tr>
<td>Print</td>
<td>0038</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME, MEMBER</td>
<td>DISP</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
</tbody>
</table>
Retrieve Resource

Example: This is an example of a RETRIEVE RESOURCE statement:

```plaintext
DEFINE PLAN NAME=PLAN1
DEFINE PHASE NAME=PHASE01, NODE=NODE1
   RET RESO NAME=PROC1,
      RESSTAT=COMP
DEFINE PHASE NAME=PHASE02, NODE=NODE2
   RET RESO NAME=PROC2
END
```

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
<td>0240</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, BYDESTID, NETID, NETLU, RESSTAT</td>
</tr>
<tr>
<td>Transaction</td>
<td>0026</td>
<td>RESTYPE, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
</tbody>
</table>

Table 40 (Page 3 of 3). Applicability of RETRIEVE RESOURCE operands
SEND MICROCODE

Use the SEND MICROCODE statement to send microcode or microcode customization data to the node specified in the preceding DEFINE PHASE statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND MICROCODE</td>
<td>(NAME= microcode-name</td>
<td>GROUPID= microcode-group-name)</td>
</tr>
<tr>
<td></td>
<td>[.BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[.DELMATCHIND=((n,L</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DELNAME= microcode-delete-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DESTRUCTION=A</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[.MATCHIND=((n,L</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TRACK=YES</td>
<td>NO]</td>
</tr>
</tbody>
</table>

NAME  
Specify the name of the microcode to be sent to the node. You can partially qualify the name. For an explanation of full and partial qualification.
Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

GROUPID  
Specify the name of the group of microcode to be sent to the node. You can specify up to 8 alphanumeric characters. The first character must be alphabetic. The group of microcode must already be defined in the DRD.

BYDESTID  
Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced.
BYDESTID is valid only if NETID and NETLU are not specified.

DELMATCHIND  
Specifies the list of matching indicators that the node must use to identify the microcode to be deleted. The qualifier number referred to by the DELMATCHIND is specified by \( n \).
You can specify the following values:

L  Set low value
I  The relative matching indicator is ignored. This means that if only one name is found matching the partial match criteria, it is assumed to be successful, whatever the value of the related token.
H  Set high value.

The DELMATCHIND operand is optional. The default is H. You must also specify DESTRUCTION=A (allowed) for the DELMATCHIND operand to be valid. The “must match” tokens must be equal to the same tokens specified in the MATCHIND parameter. If you do not specify values, they are set to the values of the tokens specified in the MATCHIND parameter.
SEND MICROCODE

DELMICR  Specify the name of the microcode to be deleted at the node. You can partially qualify
the name. The DELMICR operand is valid only if you also specify DESTRUCTION=A
(allowed) and NAME. If you do not specify DELMICR, the current value of the NAME
operand is kept.

DESTRUCTION  Specify whether the microcode to be installed can overwrite microcode already stored at
the node, if there is a lack of storage space. The DESTRUCTION values can be:

A  Overwriting is allowed.
N  Overwriting is not allowed.

The DESTRUCTION operand is valid only when you also specify the NAME operand.

MATCHIND  Specify the list of matching indicators that the host will use to identify the microcode to be
sent to the node. The values of the MATCHIND parameter can be:

n  The qualifier ordinal number the matching indicator refers to.
L  Set low value.
I  The relative matching indicator is ignored.
H  Set high value.

This operand is valid only if you use partial naming in the NAME operand. If you partially
qualify the NAME and you do not enter a value for the MATCHIND operand, the default
value is H (set high value).

NETLU  Specify the ordinal number of the token to be automatically substituted by the actual
network logical unit name (NETLU) of the target node. Do not use matching indicators for
this token. You must set the token in the resource name to an asterisk (*). You must
also specify a value for the NETID operand.

NETID  Specify the ordinal number of the token to be automatically substituted by the actual
network ID (NETID) of the target node. You must set the token in the resource name to
an asterisk (*). You must also specify a value for the NETLU operand.

TRACK  Specify whether tracking is to be performed for this transmission. The person who
installs NetView DM for MVS defines the default value at installation time, using the
NDMRES generation macro.

Example:  Here is an example of a SEND MICROCODE statement:

SEN MIRC NAME=MCUST.3174.NA.*.*.CFG.LOCALCHG.*,  
NETLU=5,  
NETID=4
SEND RESOURCE

Use the SEND RESOURCE statement to send resources to the node specified in the preceding DEFINE PHASE statement.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEND RESourse</td>
<td>.RESTYPE=resource-type</td>
<td>NDMT, CMEP, and CMFP node types</td>
</tr>
<tr>
<td></td>
<td>[ ,NAME=resource-name ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,BYDESTID=Y[ES]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,GROUPID=resource-group-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,DISP={ NEW</td>
<td>OLD }</td>
</tr>
<tr>
<td></td>
<td>[ ,EXT=0</td>
<td>1 ]</td>
</tr>
<tr>
<td></td>
<td>[ ,KEYNUMB=key-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,FCTLNAME=resource-queue-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,LIBNAME=library-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,LMTYPE=library/member-type]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,MEMBER=member-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,MOD=0</td>
<td>nn ]</td>
</tr>
<tr>
<td></td>
<td>[ ,ORIGIN=resource-origin]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,SUBCAT=subcategory]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,SUBLIB=sublibrary-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,VER=0</td>
<td>nn ]</td>
</tr>
<tr>
<td></td>
<td>[ ,DELMATCHIND=(( n ,L</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>[ ,DELNAME=resource-delete-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,DESTRUCTION=A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,MATCHIND=(( n ,L</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>[ ,NETLU=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[ ,TRACK=Y[ES]</td>
<td></td>
</tr>
</tbody>
</table>

Note: Table 41 on page 389 shows which operands are required, and which are optional.

RESTYPE Specify the resource type name or type code, described in “NetView DM for MVS Resource Types” on page 566. This operand is mandatory for all resource types apart from microcode and microcode customization data. You can omit this operand if you specify a resource name and the first token of the resource name is MCUST or MCODE.

You can specify LU 6.2 resource types for nodes with CMEP or CMFP functional capabilities.

For nodes with NDMT functional capabilities, you can only specify LU0 resource types.

BYDESTID Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the tokens named “NETID” and “NETLU.” If GROUPID is also specified, the substitution is performed for each resource of the group; in this case all the tokens set to “NETID” or “NETLU” when defining the group are replaced.

BYDESTID is valid only if NETID and NETLU are not specified.

NAME Specify the name of the resource to be sent to the node. You can partially qualify the name using one or more wildcard characters specifying one of the following operands:

- MATCHIND
- NETLU
- NETID
The wildcard replaces completely the token you omit. The first token must be specified. It cannot be replaced by a wildcard. Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

GROUPID
Specify the name of the group of resources to be sent. You can specify up to 8 characters. The first character must be alphabetic. The group of resources must already be defined in the DRD. If you specify GROUPID, you must also specify RESTYPE. Depending on the RESTYPE you specify, you must also specify the FCTLNAME and ORIGIN operands.

DISP
Specify how the resources you send are to be managed at the target repository. If you manage node types with NDMT functional capability, you can specify these values:

NEW
Resources that you send are stored in the target repository only if they do not already exist in the target repository.

OLD
Resources that you send are stored in the target repository regardless of whether they already exist in the target repository.

KEEP
Resources that you send are kept in the repository after they have been processed. You can only specify KEEP for data set, dump, and print resource types.

DELETE
Resources that you send will be deleted from the repository after they have been successfully completed and the phase terminated with return code 0.

If you manage node types with CMEP or CMFP functional capability, do not specify the first value in the syntax of the DISP parameter; it is ignored. The syntax of the DISP parameter must be [DISP=KEEP|DELETE]

EXT
This operand applies to DSCB resources only. Specify an extent (0 or 1), or a key number, but not both. If you do not specify an extent or a key number, the default is EXT=0.

KEYNUMB
This operand applies to DSCB resources only. Specify either a key number (from 1 to 8), or an extent with the EXT operand, but not both. If you do not specify a key number or an extent, the default is EXT=0.

FCTLNAME
Specify the file control name (up to 8 alphanumeric characters, first character alphabetic) under which the resource to be transmitted is stored at the host. Refer to Table 41 on page 389.

LIBNAME
Specify a library name.

LMTYPE
Specify the library type:

C Core image library
P Procedure library
R Relocatable library
S Source library

For member resource types, you can specify up to eight alphanumeric characters.

Note: If you specify a library type of S, you must also specify a sublibrary name in the SUBLIB operand.

MEMBER
Specify the member name.
**MOD** Specify the modification level of the resource. This is an integer from 0 to 99. This operand only applies to resources that are stored in the NetView DM for MVS library. If MOD applies to the resource types you are sending (see Table 41 on page 389), the default is 0. If MOD does not apply to the resource types you are sending, NetView DM for MVS sets the value blanks.

**ORIGIN** Specify the name of the resource origin.

**SUBCAT** Specify the subcategory (from 1 to 16). If you do not enter a value, NetView DM for MVS assumes a blank.

**SUBLIB** Specify the sublibrary name. The default is to send the entire library.

**VER** Specify the version level. This is an integer from from 0 to 99. This operand only applies to resources that are stored in the NetView DM for MVS library. If VER applies to the resource types that you are sending (see Table 41 on page 389), the default is 0. If VER does not apply to the resource types you are sending, NetView DM for MVS sets the value to blanks.

**DELMATCHIND** Specify the list of matching indicators that the node will use to identify the resource to be deleted. The qualifier number referred to by the DELMATCHIND is specified by \( n \).

You can specify any of these values:

- **L** Set low value.
- **I** The relative matching indicator is ignored. This means that if only one name matches the partial match criteria, it is assumed to be successful, whatever is the value of the related token.
- **H** Set high value.

The DELMATCHIND operand is optional. The default value is **H**. The DELMATCHIND operand is valid only if you also specify **DESTRUCTION=A** (allowed). The “must match” tokens must be equal to tokens specified in the MATCHIND parameter. If you do not specify values, they are set to the tokens specified in the MATCHIND parameter.

**DELNAME** Specify the name of the resource to be deleted at the node. You can partially qualify the name. The DELNAME operand is valid only if you also specify **DESTRUCTION=A** (allowed) and **NAME**. If you do not specify DELNAME, the current value of the NAME operand is kept.

If you specify a value for DELNAME that is different from that of the NAME operand, NetView DM for MVS checks that all the tokens of the NAME operand whose token attribute is **MUST MATCH** are equal to the corresponding tokens of the DELNAME operand. If they are not, NetView DM for MVS ends the functions with return code 8 and issues the sense code 08A40001.

**DESTRUCTION** Specify whether the resource to be stored can overwrite the resource already stored at the node, if there is a lack of storage space. The DESTRUCTION values can be:

- **A** Overwriting is allowed.
- **N** Overwriting is not allowed.

**MATCHIND** Specify the list of matching indicators that the host will use to identify the resource to be sent to the node. The values of the MATCHIND parameter can be:
The qualifier ordinal number the matching indicator refers to. It cannot be equal to 1.

L  Set low value.

I  The relative matching indicator is ignored.

H  Set high value.

A  Send all the resources of the holding file whose name mask matches the name you specified in the NAME operand to the target for processing.

If you specify the A value for the MATCHIND operand, do not specify the other values that MATCHIND can take.

For example, if the name of the resource of the holding file that you specify in the NAME operand is:

```
NAME=IBM.*.RES01.*
```

the value that the match indicator can assume is:

```text
MATCHIND=((2,A),(4,A))
```

You cannot specify the following values for the match indicator:

```text
MATCHIND=((2,A),(4,L))
MATCHIND=((2,A),(4,H))
```

No combination of the NETID and NETLU operands is possible when you specify the A value for the MATCHIND operand.

Do not specify the BYDESTID operand when you specify the A value for the MATCHIND operand.

Do not specify the DELNAME operand when you specify the A value for the MATCHIND operand.

If you specify the DESTRUCTION operand, the name of the resource that you specify in the DELNAME operand is assumed as the resource name.

This operand is valid only if you use partial naming in the NAME operand. If you partially qualify the NAME and you do not enter a value for the MATCHIND operand, the default value is H (set high value).

**NETLU**

Specify the ordinal number of the token to be automatically substituted by the actual network logical unit name (NETLU) of the target node. Do not use matching indicators for this token. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETID operand.

**NETID**

Specify the ordinal number of the token to be automatically substituted by the actual network ID (NETID) of the target node. You must set the token in the resource name to an asterisk (*). You must also specify a value for the NETLU operand.

**TRACK**

Specify whether tracking is to be performed for this transmission. The person who installs NetView DM for MVS defines the default value at installation time, using the NDMRES generation macro.

**Usage:** Table 41 on page 389 shows which operands are required, and which are optional, for the SEND RESOURCE control statement.
### Table 41 (Page 1 of 2). Applicability of SEND RESOURCE operands

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure</td>
<td>0158</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>CLIST</td>
<td>0040</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Data set</td>
<td>0030</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Errorlog</td>
<td>0034</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Error log (CMEP)</td>
<td>0250</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Job</td>
<td>0041</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Library</td>
<td>0052</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB</td>
</tr>
<tr>
<td>Member</td>
<td>0050</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB, MEMBER</td>
</tr>
<tr>
<td>Message</td>
<td>0017</td>
<td>RESTYPE, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Microcode</td>
<td>0060</td>
<td>NAME or GROUPID</td>
<td>RESTYPE, MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Panel (not VSE)</td>
<td>0036</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Panel (VSE)</td>
<td>0036</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB, MEMBER</td>
</tr>
</tbody>
</table>
### Table 41 (Page 2 of 2). Applicability of SEND RESOURCE operands

<table>
<thead>
<tr>
<th>Resources</th>
<th>Type</th>
<th>Required Operands</th>
<th>Optional Operands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>0038</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Program (not VSE)</td>
<td>0042</td>
<td>RESTYPE, NAME or GROUPID, VER, MOD</td>
<td>DISP</td>
</tr>
<tr>
<td>Program (VSE)</td>
<td>0042</td>
<td>RESTYPE, LIBNAME or GROUPID, LMTYPE, VER, MOD, SUBLNAME</td>
<td>SUBLIB, MEMBER</td>
</tr>
<tr>
<td>PTF</td>
<td>0044</td>
<td>RESTYPE, NAME or GROUPID, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Software</td>
<td>0070</td>
<td>NAME or GROUPID, RESTYPE</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Storage dump</td>
<td>0032</td>
<td>RESTYPE, NAME or GROUPID, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td>RESTYPE, NAME</td>
<td>MATCHIND, DELNAME, DELMATCHIND, DESTRUCTION, NETID, NETLU, TRACK, BYDESTID</td>
</tr>
<tr>
<td>Transaction</td>
<td>0026</td>
<td>RESTYPE, ORIGIN, FCTLNAME</td>
<td>DISP</td>
</tr>
</tbody>
</table>
Example: This is an example of a SEND RESOURCE statement:

```
SEND RESOURCE NAME=IBM.NDM.FIX.13.FIX1.*.*,
    NETLU=6,
    NETID=7,
    TRACK=YES,
    DESTRUCTION=A,
    RESTYPE=0070
```
UNINSTALL COMPONENT

Use the UNINSTALL COMPONENT statement to remove a component from a node. A component is removed by removing a refresh change file and associated update and fix change files that share the same component name as the refresh change file.

**Note:** The Uninstall function can only be performed on a RISC System/6000 server.

**Syntax**

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNInstall COMPONENT</td>
<td>NAME=component-name</td>
<td>CMEP nodes only</td>
</tr>
<tr>
<td></td>
<td>,RESTYPE=resource-type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,LEVEL=level]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,DATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,TIME=hhmm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,TIMEIND=G[MT]</td>
<td>L[OCAL]]</td>
</tr>
<tr>
<td></td>
<td>[,TRACK=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[,BYDESTID=Y[ES]</td>
<td>N[O]]</td>
</tr>
<tr>
<td></td>
<td>[,NETID=token-number]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,NETLU=token-number]</td>
<td></td>
</tr>
</tbody>
</table>

**NAME**

Specify the name of the component to be uninstalled at the node. This operand is mandatory.

You can specify up to 57 characters. The component name is the part of an FS global name that precedes the REF token in a refresh change file name.

Appendix C, “Resource Naming Conventions” on page 569 describes the syntax of resource names.

**RESTYPE**

Specify the resource type name or type code of the change files associated with the component. This operand is mandatory. “NetView DM for MVS Resource Types” on page 566 shows the resource types for which the Uninstall function can be performed.

**Note:** Remember that you can only specify LU 6.2 resource types for this control statement, as it is valid for CMEP nodes only.

**LEVEL**

Specify the refresh level of the component to be uninstalled. You can specify up to 16 numeric characters, but not all zeroes. The node checks that the level you expect to uninstall is the level that is actually installed.

This operand is optional. If not specified, the node will uninstall whatever level of the component is installed at the node.

**DATE**

Specify the date and time when you want to uninstall the resource at the node. You must specify both the date and the time. If you do not specify any values, the default is immediate execution.

**TIME**

Specify if the date and time (DATE and TIME) that you specified for the uninstallation at the node are expressed in local time at the node, or in Greenwich mean time (GMT).

**TRACK**

Specify whether tracking is to be performed for this function. The person who installs NetView DM defines the default value at installation time, using the NDMRES generation macro.
BYDESTID  Specifies if the automatic substitution of tokens with the actual network LU name of the target node is to be performed for the NETID and NETLU tokens. BYDESTID is valid only if you do not specify the NETID and NETLU operands.

NETID
NETLU  Indicate that the network ID (NETID) and network LU (NETLU) names of the target node are to be inserted in the name at the specified positions. You must specify these two operands together. You must also specify the corresponding tokens in the name as asterisk (*) without matching indicators.

ALTERACTIVE  Specify whether the target node can uninstall the component in the active system, or if a subsequent activation is required to uninstall the component.

If you specify ALTERACTIVE=ALLOWED, the node will determine whether it is necessary for node elements to be activated in order for the component to be uninstalled.

If you specify ALTERACTIVE=NO, the node cannot activate the affected elements during the uninstall process and they will be activated only when the node is reactivated. The node is reactivated with an ACTIVATE NODE command from NetView DM, or with a re-IPL or re-IML from the node operator.

If you do not enter a value for this operand, NetView DM does not send either of the settings to the entry point. The target node will determine the default.

Example: Here is an example of an UNINSTALL COMPONENT statement. This statement removes the refresh change file called IBM.1234567.13.REF./zerodot2 and associated update and fix change files that share the same component name as this refresh change file.

UNINSTALL COMPONENT NAME=IBM.1234567.13,
    LEVEL=2,
    DATE=/zerodot11891,
    TIME=11/zerodot/zerodot,
    ALTERACTIVE=N,
    RESTYPE=SOFTW
Chapter 17. Managing Submitted Plans

This chapter describes how to manage transmission plans that have been submitted to the TCF. The plans that you submit for execution are stored in the transmission control file (TCF). After you have submitted a plan, you can look at the information about that plan in the TCF. The TCF contains the following types of information for each plan:

- The contents and the schedule of the plan, defined when you created the plan
- The status of the plan, its phases and phase-by-node records, and functions
- The transmission function return codes

You can use TCFMAINT to:

- Delete a plan
- Erase a plan
- Print a plan
- Save a plan
- Reset a plan

The chapter also explains how you can erase, print, or save node-solicited requests (NSRs) using TCFMAINT.

Printing the Contents of a Plan

You can use the PRINT PLAN statement to print the contents of a transmission plan (see “PRINT PLAN” on page 416). This section describes the information contained in the TCF for each plan. This information is divided into:

- Plan status information
- Phase status information
- Phase by node information
- Function information for LU0 node types
- Function information for LU 6.2 node types

The information is ordered alphabetically.

Plan Status Information

Deleted by: If the plan has been deleted, the user ID of the person who deleted it.

Highest return code: The highest return code of all phases contained in the plan.

Max delay: If the plan is recursive, the start time delay specified when the plan was created.

Number of completed/deleted phases: The number of phases already transmitted or deleted in the plan.

Planner Id: The user ID of the person who created the plan.

Plan notes: The notes that were entered when the plan was created.

Priority: The priority that was specified when the plan was created.

Recursion schedule: If the plan is recursive, the date of the recursion being displayed.
Phase Information (End User)

**Recursive plan:** Whether the plan is recursive or nonrecursive.

**Status:** The transmission status of the plan, which can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>All the functions in the plan have been processed. Some functions could have been skipped, due to the conditioning criteria that were specified between the phases and/or functions in the plan.</td>
</tr>
<tr>
<td>Deleted</td>
<td>The plan has been deleted, either by a user of GIX or the TCFMAINT batch utility. Recursive plans are also deleted by the TCP if the plan does not start within the time interval specified by the plan's start time delay. The plan remains in the TCF but is flagged as deleted.</td>
</tr>
<tr>
<td>Deleting</td>
<td>A deletion request for the plan has been entered. The TCP is going to delete the plan, but the plan is temporarily flagged as deleting.</td>
</tr>
<tr>
<td>Submitted</td>
<td>The plan has been submitted to the TCF.</td>
</tr>
</tbody>
</table>

*Table 42: Plan status*

**Submission date and time:** When the plan was submitted to the TCF.

**Submitted by:** The user ID of the person who submitted the plan.

**Total number of phases:** The number of phases scheduled in the plan.

**Phase Information**

**Completed/deleted phases by node:** The number of completed and/or deleted phase-by-node records. If the phase addresses a single node, this value can be either 0 or 1.

**Conditioning phase name:** The name of the phase whose execution conditions the execution of this one.

**Conditioning return code:** The return code that the phase is conditional upon. See “Conditioning within a Phase” on page 315.

**Cut-off date:** This is the cutoff date when execution of a phase containing transmission windows must stop.

**Cut-off time:** This is the cutoff time when execution of a phase containing transmission windows must stop.

**Cut-off time interval:** The cut-off time interval is a time interval after the scheduled start date and time. If the phase is not completed within this time, the phase is terminated with return code 24 (see Table 46 on page 405), and NetView DM for MVS issues a message.

**Deleted by:** If the plan has been deleted from the TCF, the user ID of the person who deleted it.

**Highest return code:** If the functions in the phase have been executed, this field shows the highest of all function termination codes (see Table 46 on page 405).

**Highest return code to invoke procedure:** The threshold that determines whether the batch procedure should start.
Max start date/time:  The latest date and time at which the phase can be started.

Network change effect:  Indicates whether the phase by node was deleted by the TCP because of a network change, for example, the definition of the associated node was deleted or the associated logical unit was changed.

Node name:  The name of the node, or group of nodes, that the phase addresses.

Node or group of nodes:  Whether the phase addresses a single node, or a group of nodes.

Node type:  The type of node the phase addresses.

Number of nodes:  If the phase addresses a group of nodes, the number of nodes in the group.

Phase notes:  The notes entered when the phase was created.

Phase status:  The status of the phase can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>There are no more functions in the phase to be executed. This can be because they have all been executed, or because the IOF operator has canceled the phase, or because of “Conditioning within a Phase” or “Conditioning between Phases.”</td>
</tr>
<tr>
<td>Deleted</td>
<td>A user has deleted the phase. Phases are also deleted if the cut-off time or interval was reached before the phase completed. Recursive plans are also deleted when the phase does not start within the time interval specified by the plan’s start time delay. The phase remains in the TCF but is flagged as deleted until the plan to which it belongs has been erased.</td>
</tr>
<tr>
<td>Scheduled</td>
<td>The phase belongs to a plan which has been submitted, but not yet completed or deleted.</td>
</tr>
</tbody>
</table>

Table 43. Phase status

Phase termination return code:  If a function terminates with this return code, the phase will terminate immediately.

Procedure invoked at end of phase:  The batch procedure invoked at the end of the phase.

Scheduled start date/time:  This is the date and time that the phase is scheduled to start. The TCP can begin to execute the phase at this time.

TCP error message:  The number of the message issued by the TCP, if an error occurred during the submission of a job at the end of the phase. If the value is 0, there were no errors. Refer to the Messages and Codes book for information about error messages.

Windows:  Details of transmission windows that are defined for the phase. See “DEFINE PHASE” on page 321 for details of defining transmission windows for a phase.

Phase by Node Information

Cut-off date:  This is the cutoff date when execution of a phase containing transmission windows must stop.
Phase By Node Information (End User)

**Cut-off time:** This is the cutoff time when execution of a phase containing transmission windows must stop.

**Cut-off time interval:** The cut-off time interval is an interval after the scheduled start date/time. If the phase is not completed within this time, the phase is terminated with return code 24 (see Table 15 on page 180), and NetView DM for MVS issues a message.

**DRD updated:** This field indicates whether the resource history records in the DRD have been updated. These records are updated when the transmission of the phase produces results that change the status of resources at the nodes.

**End date/time:** The time that the last attempt to transmit the phase ended.

**First start date/time:** The first time the transmission of the phase was attempted.

**Highest phase return code:** The highest function return code of all the functions that were actually executed.

**Last start date/time:** If the phase is now active, this is the time that it started.

**Max start date/time:** The latest date and time the phase can be started.

**Node type:** The type of node the phase addresses.

**No. of completed/deleted functions:** The number of functions in the phase that have been either completed or deleted (for all nodes).

**Scheduled start date/time:** This is the date and time when the phase is scheduled to start.

**Sequence termination return code:** The return code resulting from the transmission of a function that inhibits the transmission of other functions in the sequence (see Table 15 on page 180).

**Status:** The status of the phase by node record can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>A phase by node is flagged as completed when all of the functions in it have been completed, either successfully or with an unrecoverable error, or when the phase is terminated due to “Conditioning within a Phase.”</td>
</tr>
</tbody>
</table>
### Table 44 (Page 2 of 3). Phase by node status

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Deleted | A phase by node is flagged as deleted because:  
- An IOF operator entered a delete command  
- The cut-off time or interval was reached before the phase completed  
- "Conditioning between Phases"  
- An unrecoverable error was encountered while executing the phase (VSE node types only)  
- In the case of recursive plans, the phase did not start within the interval specified by the plan's start time delay  
- A network change request was issued, for example, a request to delete a node or change a logical unit name. |
| Executing | NetView DM for MVS and the remote node are currently exchanging information on the transmission line. If the node is connected indirectly to the host, the TCP actually communicates with the intermediate node. |
| HCOND | Held by Condition. This means that the phase was held because the phase which conditions it is not yet completed. |
| HDEST | Held by Destination. This means that the node was held by an IOF HOLD or IMMEDIATE HOLD command. |
| HSYST | Held by TCP. This means that the phase was held either because of a line problem, the node is down, or because of a logic problem, and the retry counter is exhausted. |
| HRETR | Held/retrying. This is either the same as for HSYST, or because an SNA Request Shutdown command was received from the node. The difference is that the retry counter is not yet exhausted. |
| HWIN | A phase with transmission windows defined was held by the TCP, either because the phase was submitted outside of a window, or the phase did not reach pending status before the window ended. |
| Held | Held by the IOF operator, or belongs to a phase defined as held. |
| Pending | At least one function has been interrupted, and NetView DM for MVS is waiting for a delayed acknowledgement or data from the node. Execution will continue when this has been received from the node. |
| Ready | The scheduled date and time have been reached, and no other condition inhibits the phase from starting (neither the phase nor the node has been held, and the line is active). The TCP has not yet started executing functions in the phase. |
Function Information for LU0 Resources (End User)

The function information contained in the TCF consists of three types of information:

- Information that identifies the function record. This includes the plan name, phase name, node name, the transmission function verb and modifier (for example, "SEND DATASET"), and the resource name (for example, "DATA1.MINE").

- Information defined when the function was defined in the plan library, before you submitted the plan to the TCF. If the function addresses a group of resources, this information is repeated for each resource in the group. The information varies according to the type of node, the type of transmission function, and the type of resource. This information refers to resource attributes, such as DISPOSITION, COMPRESSED, and PROTECTION.

- Information that describes what has happened to the function record since the plan was submitted to the TCF. These are the fields that are described below.

Bytes transmitted: The number of bytes transmitted.

DRD updated: This field indicates whether the resource history records in the DRD have been updated. These records are updated when the transmission of the phase produces results that change the status of resources at the nodes.

Highest return code: The highest return code (see Table 15 on page 180).

Status: Table 45 on page 401 shows the possible statuses of each transmission function record.

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restartable</td>
<td>The transmission of at least one function in the phase was interrupted due to a recoverable error. No other functions or resources are in pending status. An IOF operator START command, or a TCP start/restart, will cause the execution to be resumed.</td>
</tr>
<tr>
<td>Waiting</td>
<td>Waiting for the scheduled date and time.</td>
</tr>
</tbody>
</table>
Function Information for LU0 Resources (End User)

<table>
<thead>
<tr>
<th>Status</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed</td>
<td>Completed, either successfully (return code=0), or unsuccessfully.</td>
</tr>
<tr>
<td>Not started</td>
<td>The TCP has not yet started executing the functions, or the function/resource was not executed at the node (VSE nodes only).</td>
</tr>
<tr>
<td>Pending</td>
<td>This only applies to VSE, DPPX, and SSP nodes that deal with compressed resources. One of the following may be true:</td>
</tr>
<tr>
<td></td>
<td>• The function/resource has been delivered to the node. There is no transmission activity while the TCP is waiting for an acknowledgement from the node. Once the TCP receives an acknowledgement, the function status becomes completed (successfully or not).</td>
</tr>
<tr>
<td></td>
<td>• The TCP can be receiving data from the node. Once the TCP receives the data, the function status becomes completed (successfully or not), or the function can be resumed until transmission activity is completed.</td>
</tr>
<tr>
<td>Restartable</td>
<td>This does not apply to VSE nodes. The function was interrupted by a restartable error.</td>
</tr>
<tr>
<td>Running</td>
<td>The host and the remote node are currently exchanging information on the transmission line.</td>
</tr>
</tbody>
</table>

Table 45. Function status

TCP error message: The number of the message issued by the TCP, if an error occurred during the submission of a job at the end of the phase. If the value is 0, there were no errors. Refer to the Messages and Codes book for information about error messages.

Primary and secondary return codes: These are described in “Recovering from Transmission Failures” on page 180.

Node error message: These are described in “Recovering from Transmission Failures” on page 180.

Transmission Error Identification: This field identifies transmission errors that have occurred. The information can be:

• The ID of the last error message issued by the TCP.
• The ID of the error messages issued at the node, if applicable. These are either generated at the end node, or at one of the intermediate nodes.
• The primary and secondary return code from the node, if applicable. These are either generated at the end node, or at one of the intermediate nodes.
• Return and feedback codes from TPAM.
• TPAM system and user senses, if applicable.
Function Information for LU 6.2 Resources

There can be some extra fields of information for those resource types that you use at LU 6.2 node types. These are described below.

**Function:** The function name.

**Function termination rc:** See Table 15 on page 180.

**Function rc:** The function return code.

**Function status:** See Table 45 on page 401.

**Resource Name:** The name of the resource.

**Matching Indicators:** These specify the list of matching indicators used at the host to identify the resource. These can be “-” (which means “must match”), HIGH, IGNORE, or LOW. These apply only to the INSTALL, SEND RETRIEVE, DELETE, and INITIATE functions.

**Transmission return code:** The return code resulting from the transmission.

**Status:** This indicates the current condition of a function. It can be one of: COMPLETED, STARTED, RUNNING, PENDING, RESTARTABLE, or NOT RUNNING.

**Bytes transmitted:** The number of bytes transmitted.

**Node Reports:** These are a set of responses from the node, which include the following information:

**Process at the Node:** The result of the function, which can be:
- SUCCESSFUL
- NOT SUCCESSFUL
- NOT ATTEMPTED
- WILL NOT BE ATTEMPTED

**When effective:** The effectiveness of the function, which can be:
- IN USE
- ACTIVATION REQUIRED
- NOT APPLICABLE

**Initiate completion code:** The return code initiated at a NetView DM/2 node when executing the procedure, and then reported to the host. The actual return code, and its meaning, is dependent upon the specific procedure at the node.

**Activation use:** Indicates which type of activation will cause components to be canceled automatically if the installation or a test are successful, in order to release resources required to maintain removability. These can be:
- TRIAL: the altered components are used only for an activation specifying trial use superseded by production components.
- PRODUCTION: the altered components are used for any activation, unless superseded by trial components.
- NOT INSTALLED
Removability: Indicates whether the microcode will be installed in a removable state. This can be YES, NO, or DESIRED.

Side Effects: This is the effect that this transmission function can have on other resources. This applies to the ACCEPT, REMOVE, and INSTALL transmission functions only. The effect can be DELETED, REINSTALLED, or BACK-LEVEL.

Side Effect Resource Names: The names of the resources effected as a result of the transmission function, in addition to the main resource. See “Side Effects” on page 403.

Effect: The effect of the function, for example, DELETED.

Snads report code: Specifies an SNA/Distribution Services report code. These codes are listed in SNA Formats GA27-3136.

Agent report code: Specifies an SNA report code. These codes are listed in SNA Formats GA27-3136.

Server report code: Specifies an SNA report code. These codes are listed in SNA Formats GA27-3136.

TCP Error Message number: If there is an error, this is the ID of the message issued by the TCP. See “Recovering from Transmission Failures” on page 180.

Details on Process Performed: This can include the following items:

Pretest: Indicates whether changes are to be tested prior to installation. This field can be YES, NO, or DESIRED.

Posttest: Indicates whether changes are to be tested after installation. This field can be either YES, NO, or DESIRED.

Automatic removal: Indicates whether changes will be removed automatically if the installation or test fails. This field can be either YES, NO, or DESIRED.

Automatic acceptance: Indicates whether the removability of changes will be canceled automatically if the installation or a test is successful. This releases the resources required to maintain removability. This field can be either YES, NO, or DESIRED.

Destruction: Indicates whether existing changes will be explicitly overwritten at the node if storage constraints occur. This field can be either ALLOWED or NOT ALLOWED. It applies only to the INSTALL and SEND transmission functions.

Corequisites: This applies only to the INSTALL transmission function. It specifies a list of resources to be installed together with the main resource referred to.

To be deleted resource name: The global name of the resource to be deleted at the node. This is indicated only if DESTRUCTION=YES. If there is no name, the field displays NONE.

To be deleted Matching Indicator: Specifies the list of matching indicators to be used at the resource to identify the resource to be deleted at the node. Indicators can be " - " (which means “must match”), HIGH, IGNORE, or LOW. This is displayed only if DESTRUCTION=YES.

Command Execution: The result of the function. This field can be either SUCCESSFUL, NOT SUCCESSFUL, NOT ATTEMPTED, or WILL NOT BE ATTEMPTED. For the ACTIVATE function, SUCCESSFUL actually means “Will_Be_Attempted.”
Function Information for LU 6.2 Resources (End User)

**Pre Test:** The result of the pretest. This field can be either SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

**Post Test:** The result of the post-test. This field can be either SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

**Auto Remove status:** The result of automatic removal. This field can be either SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

**Auto Accept:** The status of the automatic acceptance. This field can be either SUCCESSFUL, NOT SUCCESSFUL, or NOT ATTEMPTED.

**Removability (status):** The status of removability. This field can be either INSTALLED-REMOVABLE, INSTALLED-NOT REMOVABLE, or NOT INSTALLED.

**Resources:** The status of resources can be either DELETED, BACK-LEVEL, or REINSTALLED.

**Activation:** The effectiveness of the function. This field can be either IN USE, ACTIVATION REQUIRED, or NOT APPLICABLE.

**Detailed Data:** Additional information from the node, describing conditions at the node. This can be up to four lines, each line containing 64 bytes.

**Accept if no error:** This field can be either YES or NO.

**Remove if error:** This field can be either YES or NO.

**Send:** Indicates whether the resource has to be sent. This field can be either YES or NO. It applies only to the INSTALL and INITIATE functions.

**Initiate Completion Code:** This field shows the initiate return code reported by the node.
Recovering From Transmission Failures

A transmission error occurred if the status of a plan, phase, or function is COMPLETED but with a return code other than zero, or if the status is RESTARTABLE. Follow the steps outlined below to solve the problem:

**Step 1.** Identify the functions that did not execute successfully. You can either check the functions of a single plan, or you can list all the functions that were not executed successfully.

**Step 2.** Correct whatever the cause of the transmission failure was. The rest of this section shows how to correct the three main types of error: host errors, node errors, and VTAM errors.

**Step 3.** Resubmit transmission plans to the TCF. You can either submit the entire plan again (you will probably need to specify a new phase “Scheduling date and time”). Or, if most of the functions in the plan or phase executed successfully and you do not want to execute them again, you can use the old plan in the library to build a new plan. Delete all the functions that were executed successfully from the plan, and then submit the new plan to the TCF.

Table 46 shows the transmission function return codes that you might receive:

<table>
<thead>
<tr>
<th>RC</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Function completed successfully.</td>
</tr>
<tr>
<td>04</td>
<td>Function completed, but warning messages were issued (for example, a request to delete a data set for a data set that does not exit, or a request to send an existing resource to the node). The requested operation was performed.</td>
</tr>
</tbody>
</table>
| 06 | Function was not completed due to a temporary error. For example:  
| | • An LU0 resource was not found in the holding file. If an LU 6.2 resource is not found in the holding file, it will cause transmission to end with return code 08.  
| | • A VSAM error occurred in the holding file  
| | • A printer was not available at a node for a SEND PRINT function.  
| | The function is restartable. If the phase by node that the function belongs to is marked RESTARTABLE, an IOF restart command applied to this phase by node will cause the re-execution of the failing function.  
| | If the phase by node that the function belongs to is not marked COMPLETED or DELETED, then the failing function will be re-executed the next time the TCP is started, and the function would then be completed successfully. For example:  
| | • Data set was not found in the holding file.  
| | • A VSAM error occurred in the holding file.  
| | • Printer was not available at a node for a send print function. |
| 08 | Function completed with an error that is not temporary. The function will not be run again, and is therefore not restartable, for example, a logical error exists in the plan. This return code is also issued when an LU 6.2 resource is not found in the holding file. |
| 12 | This return code is for phase by node functions. It means that the phase by node was deleted because the phase conditioning criteria were not met. |
| 20 | This return code is for phases only. It means that the phase was deleted, either using GIX or with an IOF command. This return code is also issued when NETCHNG=IMMEDIATE and the phase is addressing a node that has been deleted, or the logical unit or connection profile associated with the node has been changed. |
| 24 | This return code is for phases only. It means that the phase was deleted because the cut-off time or interval has passed. |
You can find these transmission function return codes in various fields, according to the type of node that the phase addresses:

- For DPPX, PDOS, POSC, RPS, and SSP nodes, look at:
  - The TCP error message
  - Possibly an error message from the node
  - Primary and secondary return codes (either from VTAM or the node).

- For VSE nodes, look at:
  - The TCP error message
  - Primary and secondary return codes (either from VTAM or the node)
  - The user and system sense codes from the node.

As you can see, the error message issued by the TCP is the most useful information. The three main types of transmission error are:

- NetView DM for MVS host errors
- Node errors
- VTAM errors.

You should be able to correct NetView DM for MVS errors at the host by consulting the *Messages and Codes* book. You will have to consult other sources of information for VTAM and nodes errors. Here are some examples.

### NetView DM for MVS Host Errors

Transmission errors can be the result of a problem at the host. Consult the *Messages and Codes* book to see if the information there helps you to correct the error at the host. For example, if the TCP error message is 26, you will find this description in the *Messages and Codes* book:

```
NDM0026E  SEND WITHOUT REPLACE WAS SPECIFIED FOR RESOURCE resource name
BUT THE RESOURCE ALREADY EXISTS AT NODE node identifier
```

In this case, the book tells you to either delete the resource that already exists at the node, or specify the replace option for the resource you want to send. This is all the information you need to correct this error.

### VTAM Errors

If you specified a group model, you can specify whether you want to delete all the nodes that you want excluded from the group, or select all the nodes that you want to include in the group. Enter 2 in the **Group creation criteria** field for the exclusion option. This is the default value. Enter 1 in this field for the inclusion option.

Errors caused by problems with VTAM, or at the node, are identified by return codes as well as error messages. The TCP error message identifies whether the primary and secondary return codes are from the node, or from VTAM. For example, this feedback is for a VTAM error. You can tell this by looking up the message in *Messages and Codes*. The TCP message, NDM0020E is:
NODE INFORMATION (END USER)

NDM0020E  CONNECTION LOST WITH LOGICAL UNIT unit name,
          TP ACCESS METHOD RPL RETURN CODE value, RPL FEEDBACK value.

Consult the VTAM books called Advanced Communication Function for VTAM Programming, SC27-0449
and SC23-0115, for more information.

The NetView DM for MVS operator can use this information to restart transmission, and resend the
phases that were not completed.

NODE INFORMATION

Nodes send different types of information when an error occurs, depending on the type of node:

- All nodes send primary and secondary return codes
- DPPX nodes also issue error messages
- Other node types also issue sense codes.

This is an example of information about an error at a node: The TCP message is:

NDM0099E  WHILE PROCESSING function-type FOR resource, THE TCP
          RECEIVED FROM LOGICAL UNIT unit-name, A PRIMARY RETURN CODE 020C,
          SECONDARY RETURN CODE 0F56, MESSAGE NUMBER HDKF0039

which tells you that the return codes refer to a node error.
Managing Node Solicited Requests and Node Unsolicited Reports

A NetView DM for MVS user-defined node with CMEP functional capabilities can send commands to NetView DM for MVS for processing. The command that the entry point sends is called a **node-solicited request (NSR)**. CMEP nodes can send the following commands, if they are authorized to do so:

- Send a resource to the NetView DM for MVS resource repository
- Retrieve a file from the repository
- Delete a file from the repository

When NetView DM for MVS receives an NSR from a CMEP node, NetView DM for MVS performs a node authorization check before accepting the NSR.

If the node is sending a resource and NetView DM for MVS accepts the request, the resource is stored in the repository. NetView DM for MVS send a report to the node, describing the result of the operation.

If the node is requesting a resource from the NetView DM for MVS repository, NetView DM for MVS sends the resource to the node, if it can find the resource. If NetView DM for MVS cannot find the resource, NetView DM for MVS tells the node.

If the node is requesting to delete a resource, NetView DM for MVS deletes the object if it exists in the repository, and sends a report to the node.

**Node unsolicited reports** are reports that are received from a node for functions that the node has executed locally. The following reports for functions not solicited by NetView DM for MVS at the host can be received:

- Local Install of an object at the entry point
- Local Remove of an object previously installed at the entry point
- Local Accept of an object previously installed at the entry point
- Local execution of a procedure (Initiate) at the entry point
- Local Activate at the entry point

When NetView DM for MVS receives a report for which no correlation information is found in the TCF, it is considered to be an “unsolicited report” and is processed as follows:

1. All of the report data is received and stored in the TCF, in a structure which is marked completed as soon as it is received.
2. The node unsolicited report is processed by updating the tracking in the DRD related to the involved node and resource (if tracking is required for that resource type).
3. The event notification user exit is invoked, if it was specified at installation time in the NDMTCP macro. All data related to the received Report is passed to this exit.

The node unsolicited report is not deleted automatically from the TCF. The system administrator is responsible for maintaining the TCF (see “Maintaining the TCF” on page 452).

You can keep track of NSRs and node unsolicited reports using TCFMAINT, in the same way as for tracking resources at nodes. Use the following TCFMAINT control statements to erase, print, and save node-solicited requests and node unsolicited reports:

- ERASE REQUEST
- PRINT REQUEST
- SAVE REQUEST
TCFMAINT Control Statements

You use the TCFMAINT batch utility to control and maintain the information stored in the transmission control file (TCF).

You can delete, erase, print, save, and reset transmission plans. You can erase, print, and save node-solicited requests (NSRs) and node unsolicited reports. You can also use TCFMAINT to erase, save, and print Distribution and Change Control Application Programming Interface (D&CC API) requests that are queued in the TCF. D&CC API queue entries are usually erased from the TCF by a user program. If they are not erased for some reason, you can erase, print, and save the entries yourself using TCFMAINT.

The syntax and operands of the statements you can use are described on the following pages, together with examples of how to use them. The statements are described in alphabetical order.

DELETE SPLAN

Use the DELETE SPLAN statement to delete a transmission plan. There is a difference between deleting a plan and erasing a plan. Deleting a plan does not erase the plan from the TCF. It inhibits the transmission of a plan, or, if it is already being executed, it interrupts transmission. If the TCP is not actually running when you delete a plan, the plan status is marked as DELETING instead of DELETED.

If the plan contains one or more phases which have been sent to a remote server, but have not yet been reported as successfully completed, you can choose to send a Delete request to the remote server(s). NetView DM for MVS will then carry out the following:

- If the Delete at Server request includes any phases/destinations for which the original request has not yet been transmitted, the original request will be deleted at the host.
- If the Delete at Server request includes any phases/destinations for which the original request has already been successfully executed at the remote server, no action will be taken with respect to these destinations.
- For all other phases&sr1.destinations, a Delete at Server request will be transmitted to the server.

If the remote server is running TME 10 Software Distribution Version 3.1.5 or later, it will attempt to delete the original request, and will send a confirmation report to the host if it is successful. If the host receives confirmations from all remote destinations that the requests in PENDING status have been cancelled, the plan at the host will be cancelled automatically. In the event that completed statuses are not received for all of the nodes, you can optionally Force Delete the plan, thereby ignoring the presence of requests still in pending status; however, you should note that this Force Delete option must be used from within the GIX, as it is not available from the Batch Utilities.

If the TME 10 Software Distribution Version 3.1.5 node is processing a function when it receives the Delete Pending request for that same function, it will complete the processing and return a completed status to the host for the original function, ignoring the Delete Pending request.

To erase a plan completely from the TCF, use the ERASE PLAN statement described in “ERASE PLAN” on page 413.

When You Delete a Nonrecursive Plan

- Functions that have already been executed are not affected.
- Functions that have not yet started will not be started.
- Functions that are currently being executed are interrupted.
• The plan and all its phases (phases by node) are marked as deleted.

When You Delete a Recursive Plan
• All the recursions of the plan already completed are not affected.
• The current recursion is deleted.
• No more plan recursions are generated.

When You Delete a Plan Recursion
• The recursion with the date you specify is deleted.
• The recursion for the next day is generated as usual.

When You Delete a Specific Phase
• Functions in the phase that have already been executed are not affected.
• Functions in the phases that have not yet started will not be started.
• Functions in the phase currently being executed are interrupted.
• The phase (and related phases by node) are marked as deleted.

When You Delete a Plan or Phase Using the Delete at Server Option
• In addition to the information shown for the various types of delete command above, NetView DM for MVS will send a delete request for any nodes that have functions that have been started, but for which a completed status has not been received.
• The plan or phase will only be marked for deletion at the host if confirmations are received that all pending functions for all nodes were successfully deleted.
Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete SPLAn</td>
<td>NAME=plan-name</td>
<td>All node types</td>
</tr>
<tr>
<td></td>
<td>[,RECDATE=mmddyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[,PENDING=Y[ES][N][O]]</td>
<td></td>
</tr>
</tbody>
</table>

**NAME**  
Specify the name of the transmission plan to be deleted from the TCF.

**RECDATE**  
Specify the recursion date (only for recursive plans). If you do not enter a value, the current recursion is deleted and the next recursion will not be generated. If you specify a value, the specified recursion is deleted, and the recursion for the next day is generated.

**PENDING**  
If PENDING=YES, NetView DM for MVS will behave as follows:

- All phases/destinations for which a plan has been prepared but not transmitted, the plan will be deleted at the host.
- For all phases/destinations for which a plan has already been processed at the remote server, no action will be taken.
- For all other phases/destinations, a delete request is sent to the server. If the server is running TME 10 Software Distribution Version 3.1.5 and it is able to cancel the original request before it is executed, it will do so, and report its success to the host.

The plan will only be marked for deletion at the host if confirmations are received that all requested phases for all destinations were successfully deleted.

**Example:**  
Here is an example of a DELETE SPLAN statement to delete a plan from the TCF after successfully requesting the deletion of all phases that are pending implementation at the remote destinations identified in the plan.

```plaintext
DEL SPLA NAME=MYPLAN,RECDATE=110290,SERVER=Y
```
The ERASE APIQENT statement erases all of the D&CC API queue entries that have a status of COMPLETED or DELETED from the TCF. No information is saved for these queue entries.

Normally these entries are purged by user programs. This statement gives you the opportunity to delete requests that are not successfully purged, but a better long term solution is to correct your user programs. You can only use the ERASE APIQENT statement by running a batch job when the TCP is not active.

Syntax

<table>
<thead>
<tr>
<th>Statement</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERAse APIQent</td>
<td></td>
<td>All node types</td>
</tr>
</tbody>
</table>

There are no operands for this control statement.
ERASE PLAN

Use the ERASE PLAN statement to physically remove a plan from the TCF, making space for something else. You can only erase plans that have a status of COMPLETED or DELETED.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERASE PLAN</td>
<td>[NAME=plan-name] [,RECDATE=mmddyy]</td>
<td>All node types</td>
</tr>
</tbody>
</table>

**NAME**  
Specify the name of the transmission plan to be erased from the TCF. If you do not enter a name, all the plans whose status is COMPLETED or DELETED are erased.

**RECDATE**  
Specify the recursion date of the plan recursion to be erased (for recursive plans only). If you do not enter a date and the plan is recursive, the plan is deleted. You can only specify this operand if you also specify NAME.

You can also use the ERASPLAN job to erase plans. See “Erasing Plans Using the DB2 Environment” on page 423.
ERASE REQUEST

Use the ERASE REQUEST statement to erase all of the node-solicited requests that have the status of COMPLETED or DELETED from the TCF. You can also erase node unsolicited reports. No information is saved regarding the node-solicited request.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERAse REQUEST</td>
<td></td>
<td>All node types</td>
</tr>
</tbody>
</table>

There are no operands for this statement.
PRINT APIQENT

Use the PRINT APIQENT statement to print all of the information in the DSXPRINT file, regardless of its status. This file contains information about each D&CC API queue entry. The information for each request is printed in two sections. The first section contains information about the resources, the second about the functions.

Syntax

<table>
<thead>
<tr>
<th>Statement</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT APIQENT</td>
<td></td>
<td>All node types</td>
</tr>
</tbody>
</table>

There are no operands for this statement.
PRINT PLAN

Use the PRINT PLAN statement to print all of the information related to plans or plan recursions. You can print all plans, regardless of their status. The print output is stored in the DSXPRINT file.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT PLAN</td>
<td>[ERRPLAN={NO</td>
<td>YES}][,NAME=plan-name][,RECDATE=mmddyy]</td>
</tr>
</tbody>
</table>

**ERRPLAN** Specify YES if you want to print the information related to all of the plans that are currently in the TCF and are not successfully completed. The default value is NO. You cannot specify this operand if you specify a value for NAME, described below.

**NAME** Specify the name of a transmission plan to be printed from the TCF. If you do not enter a name, all of the plans in the TCF are printed.

**RECDATE** Specify the recursion date of the plan to be printed (for recursive plans only). If you do not enter a date and the plan is recursive, today’s date is used. You must also specify NAME before you can use this operand.
PRINT REQUEST

Use the PRINT REQUEST statement to print all of the information in the DSXPRINT file, regardless of its status. You can print information about node-solicited requests, and node unsolicited reports.

The information on each node-solicited request and node unsolicited report is printed in two sections. The first section contains information about resources, the other about functions.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT REQUEST</td>
<td></td>
<td>All node types</td>
</tr>
</tbody>
</table>

There are no operands for this statement.
RESET PLAN

Use the RESET PLAN statement to reset a plan or plan recursion in the TCF. You can also reset specific phases or phase-by-node records within the plan, in which case the plan must contain:

- One or more unsuccessful functions. A function is defined as unsuccessful if it completes with return code 8 and one of the following is true:
  - The function status is COMPLETED.
  - The phase-by-node status is COMPLETED.
  - The phase status is COMPLETED or SCHEDULED.
  - The plan status is SUBMITTED or COMPLETED.
- One or more phase-by-node records with a status of HSYST or HRETR.

Attention

Your system administrator may have assigned the same logical unit name (LUNAME) to more than one node. All of the phase-by-node records that are in HSYST or HRETR status, and which address nodes with the same logical unit name, are released, even if they do not belong to the plan that you are resetting but to a different plan.

You will receive a warning message if one of these conditions is not met and you try to reset phases or phase by node records in a plan. When you perform a reset, you can also specify a new scheduling time and date. The new time and date only apply to phase-by-node records with a status of COMPLETED, not HSYST or HRETR.

You can also remove transmission windows that have been defined for a phase by resetting the plan. In this case, you can reset the plan even if the phase is not in HSYST or HRETR status or there are no unsuccessful functions.

For recursive plans, you can reset either the current recursion, or the last completed recursion, if the scheduled date and time of the current plan recursion have not yet been reached.

When you reset the last completed recursion and specify a new scheduling date and time, the date and time you specify must be earlier than that of the current recursion.

If the plan is recursive, the scheduling date you specify must be either the date of the current recursion or the date of the previous recursion. If the date is that of the previous recursion, you must specify a scheduling time that is earlier than the current recursion. If you do not specify a date and time as described, you will receive an error message (NDM8200E).

Note: The S/36 fan-out capability is not used when you reset a plan.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESet PLAN</td>
<td>NAME=plan-name</td>
<td>All node types</td>
</tr>
<tr>
<td></td>
<td>[.RECDATE=recursion-date]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.PHASE=phase-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.NODE=node-name]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.DATE=scheduling-date][mmdyy]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.TIME=scheduling-time][hhmm]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[.RESETWIN=Y[ES]N[O]]</td>
<td></td>
</tr>
</tbody>
</table>
**NAME**
Specify the name of the plan to be reset (from 1 to 8 characters). This operand is mandatory.

**RECDATE**
For recursive plans, specify the date of the plan recursion to be reset, in the format *mmddyy*. The default is the date of the current plan recursion.

**PHASE**
Specify the name of the phase to be reset (from 1 to 8 characters). If this operand is specified, only the phase by node records belonging to this phase are processed. If it is not specified, all the phase by node records in the plan will be processed.

**NODE**
Specify the name of a node (from 1 to 8 characters). This operand is optional. You can only specify this operand if you also specify PHASE. If you do specify this operand, only the phase by node record for this node will be reset.

**DATE**
Specify the scheduling date (in the format *mmddyy*). The date specified by this operand replaces the scheduling date of all the phase by node records that contain any functions eligible for reset. If it is not specified, the scheduling date is left unchanged.

**TIME**
Specify the scheduling time (in the format *hhmm*). The time specified by this operand replaces the scheduling time of all the phase by node records that contain any functions eligible for reset. If it is not specified, the scheduling time is left unchanged.

**RESETWIN**
Specify **YES** to remove transmission windows for a specific phase belonging to a plan.
Use this operand only if the phase has been submitted with transmission windows defined, or you will receive an error message.

You must specify the name of the plan with the **NAME** operand, and the name of the phase with the **PHASE** operand. Do not specify any of the other operands.

You can reset the phase to remove windows even if the phase is not in HSYST or HRETR status or there is no function that was unsuccessfully completed. The windows are removed from the phase definition for all of the destinations involved in the phase. The phase changes to the READY status and is ready to be executed.

**Example:**
Here is an example of a **RESET PLAN** statement to reset a plan in the TCF:

```
RESET PLAN NAME=PLAN01,
    PHASE=PHASE01,
    NODE=NAVIG1
```

Here is an example of a statement to remove transmission windows defined for the phase:

```
RESET PLAN NAME=PLAN01,
    PHASE=PHASE01,
    RESETWIN=YES
```
SAVE APIQENT

SAVE APIQENT

Use the SAVE APIQENT statement to save all of the D&CC API queue entry information in the TCF to the DSXSTAT file, regardless of its status.

Syntax

<table>
<thead>
<tr>
<th>Statement</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVe APIQent</td>
<td>ERASE=Y[ES]</td>
<td>N[O]</td>
</tr>
</tbody>
</table>

ERASE      Specify whether you want to erase all D&CC API requests from the TCF, after they have been saved in the DSXSTAT file. If you specify **YES**, all D&CC API requests with a status of COMPLETED or DELETED are saved in the DSXSTAT file and then erased from the TCF. This operand only works when the TCP is *not* active.
SAVE PLAN

Use the SAVE PLAN statement to save information about plans or plan recursions. All plans can be saved, regardless of their status. Saved plans are stored in a sequential file called DSXSTAT. The record layout of the saved transmission plans is described in the Base Application Programming book.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE PLAN</td>
<td>[NAME=plan-name] [,RECDATE=mmddyy] [,ERASE={Y[ES]</td>
<td>N[O]}]</td>
</tr>
</tbody>
</table>

**NAME**
Specify the name of the plan to be saved. If you do not enter a name and you specify **ERASE=NO** (which is the default value), then all of the plans in the TCF are saved regardless of their status. If you do not enter a name and you specify **ERASE=YES**, only those plans that have a status of COMPLETED or DELETED are saved and then erased from the TCF.

**RECDATE**
For recursive plans, specify the recursion date. If you do not enter a date and the plan is recursive, today’s date is used. You must also specify NAME before you can use this operand.

**ERASE**
Specify whether the plan is to be erased from the TCF after it has been saved. The plan status must be COMPLETED or DELETED.
SAVE REQUEST

SAVE REQUEST

Use the SAVE REQUEST statement to save all of the node-solicited request and node unsolicited report information contained in the TCF to the DSXSTAT file, regardless of its status. The saved information is in two parts. The first contains resource information, the other function information. The record layout of the saved request is described in the *Base Application Programming* book.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAVE REQUEST</td>
<td>ERASE=Y[ES]</td>
<td>N[O]</td>
</tr>
</tbody>
</table>

ERASE Specify whether node-solicited requests or node unsolicited reports are to be erased from the TCF after being saved in the DSXSTAT file. The status of node-solicited requests must be COMPLETED or DELETED. The ERASE operand values can be:

- Y Erase the node-solicited information from the TCF.
- N Do not erase the node-solicited information from the TCF.
Erasing Plans Using the DB2 Environment

You can also use the ERASPLAN job to erase plans. Using the ERASPLAN job you erase all the plans whose status is COMPLETED from the TCF. The difference between the ERASE PLAN statement of the TCFMAINT utility and the ERASPLAN job is that if you use the ERASPLAN job the plan records are not put in the DB2 log.

The ERASPLAN job consists of the following steps:

1. The DELETE step, which deletes the temporary datasets.
2. The UNLOAD step, which allocates the datasets, reads DB2, and writes in the datasets of the TCF tables the record plans whose status is not COMPLETED or DELETED.
3. The LOAD step, which writes the records plans whose status is not COMPLETED or DELETED in DB2 and deletes the COMPLETED or DELETED plans from the DB2 without putting the records plans in the DB2 log.
4. The UTIL step, which restores the DB2 tables of TCF in the database.

Before running the ERASPLAN job you must customize the following fields:

- **XX_NDMLOAD**: The name of the NVDM LOAD library.
- **XX_DB2LOAD**: The name of the DB2 LOAD library.
- **XX_OWNER**: The name of the table qualifier. It matches the owner parameter in the SPMF Parameters panel in the NetView DM for MVS installation dialog. See the Installation and Customization manual for additional information.
- **XX_QUALDS**: The primary qualifier for the datasets.
- **XX_PLANNAME**: The name of the plan in the form FIYnUTIL where n is the NVDM table instance number.
- **XX_DBNAME**: The name of the database where the TCF tables are defined. It matches the TCF SPMF DB name in the SPMF Parameters panel in the NetView DM for MVS installation dialog (for example, NVDMTCF1). See the Installation and Customization manual for additional information.

Figure 101 on page 424 shows the ERASPLAN job.
Erasing Plans Using the DB2 Environment

//ERASPLAN JOB

**********************************************************************************************************************************************
/** LICENSED MATERIALS - PROPERTY OF IBM
/**
/** **
/** ** 5685-016 FOR MVS
/** ** (C) COPYRIGHT IBM CORP. 1988, 1998.
/** ** ALL RIGHTS RESERVED.
/** ** US GOVERNMENT USERS RESTRICTED RIGHTS
/** ** - USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
/** ** GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
/** **
/** **********************************************************************************************************************************************
/**
/** THIS JOB DELETES THE NVDM PLANS WHICH ARE COMPLETED,
/** SELECTING THE OTHER AND COPIING THEM ON TEMPORARY DATASETS.
/** IT REPLACES THE OLD DB2 PLAN TABLES WITH THE RECORDS WHICH
/** HAVE BEEN SELECTED IN THE ABOVE DATASETS WITHOUT FILLING
/** THE DB2 LOG.
/**
/** THIS JOB CONSISTS OF THE FOLLOWING STEPS:
/**
/** 1. THE DELETE STEP CLEANS THE TEMPORARY DATASETS.
/** 2. THE UNLOAD STEP ALLOCATES THE DATASETS,
/** READS DB2 AND WRITES IN THE DATABASE THE RECORDS PLANS
/** WHOSE STATUS IN NOT COMPLETED OR NOT DELETED.
/** 3. THE LOAD STEP WRITES THE RECORDS PLANS WHOSE STATUS
/** IS NOT COMPLETED OR NOT DELETED IN DB2 AND DELETES THE
/** COMPLETED OR DELETED PLANS FROM THE DB2.
/** 4. THE UTIL STEP RESTORES THE DB2 TABLES OF TCF IN THE
/** DATABASE.
/**
/** BEFORE RUNNING THIS JOB, CUSTOMIZE THE FOLLOWING PARAMETERS
/** - XX_QUALDS : PRIMARY QUALIFIER FOR DATASETS
/** - XX_NDMLOAD : NVDM LOAD LIBRARY
/** - XX_DB2LOAD : DB2 LOAD LIBRARY
/** - XX_OWNER : TABLE QUALIFIER
/** - XX_PLANAME : NAME OF THE PLAN, IT MUST BE: FIYnUTIL
/** WHERE n IS THE NVDM TABLES INSTANCE NUMBER
/** - XX_DBNAME : DATABASE NAME WHERE TCF TABLES ARE DEFINED
/** **********************************************************************************************************************************************
/**
/** DELETE WORK DATASET
/** DELETE EXEC PGM=IDCAMS
/** SYSPRINT DD SYSOUT=*s
/** SYSOUT DD SYSOUT=*s
/** SYSSIN DD *
/** DELETE 'XX_QUALDS.ERR'
/** DELETE 'XX_QUALDS.PLAN'
/** DELETE 'XX_QUALDS.PHASE'
/** DELETE 'XX_QUALDS.PHNODE'
/** DELETE 'XX_QUALDS.FUNC'
/** DELETE 'XX_QUALDS.RES5'
/** DELETE 'XX_QUALDS.RES6'
/** DELETE 'XX_QUALDS.RES7'
/** DELETE 'XX_QUALDS.RES8'
/** DELETE 'XX_QUALDS.RES9'
/** DELETE 'XX_QUALDS.RESC'
/** DELETE 'XX_QUALDS.RESD'
/** SET MAXCC=0

Figure 101 (Part 1 of 5). The ERASPLAN job
Erasing Plans Using the DB2 Environment

*TESTRC IF (RC<=4) THEN

*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
Erasing Plans Using the DB2 Environment

//FIYMPRTC DD DSN=XX_QUALDS.RESC,
// DISP=(NEW,CATLG),
// DCB=(DSORG=PS,LRECL=3633,BLKSIZE=3633,RECFM=FB),
// SPACE=(CYL,(2,2),RLSE),UNIT=SYSDA
//FIYMPRTD DD DSN=XX_QUALDS.RESD,
// DISP=(NEW,CATLG),
// DCB=(DSORG=PS,LRECL=1603,BLKSIZE=1603,RECFM=FB),
// SPACE=(CYL,(2,2),RLSE),UNIT=SYSDA
//SYSPRINT DD SYSOUT=A
//SYSTSPRT DD SYSOUT=*
//SYSUDDUMP DD SYSOUT=*
//SYSTGIN DD *
// DSN SYSTEM(DSN1)
// RUN PROGRAM(FIYUEPLA) PLAN(XX_PLANAME)
// ENDTEST ENDIF

//TESTRC IF (RC<=4) THEN
//***********************************************************************
//*********************************************************************** LOAD DB2 TABLES
//***********************************************************************
//***********************************************************************
//LOAD EXEC DSNUPROC,SYSTEM=DSN1,UID='TEMP',UTPROC=''
//STEPLIB DD DSN=XX_DB2LOAD,DISP=SHR
//***********************************************************************
//DSNUPROC.SORTWK01 DD DSN=XX_QUALDS.SORTWK01,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,ROUND),
// UNIT=SYSDA
//DSNUPROC.SORTWK02 DD DSN=XX_QUALDS.SORTWK02,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,ROUND),
// UNIT=SYSDA
//DSNUPROC.SORTWK03 DD DSN=XX_QUALDS.SORTWK03,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,ROUND),
// UNIT=SYSDA
//DSNUPROC.SORTWK04 DD DSN=XX_QUALDS.SORTWK04,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,ROUND),
// UNIT=SYSDA
//DSNUPROC.SYSREC00 DD DISP=SHR,DSN=XX_QUALDS.PLAN
//DSNUPROC.SYSREC01 DD DISP=SHR,DSN=XX_QUALDS.PHASE
//DSNUPROC.SYSREC02 DD DISP=SHR,DSN=XX_QUALDS.PHNODE
//DSNUPROC.SYSREC03 DD DISP=SHR,DSN=XX_QUALDS.FUNC
//DSNUPROC.SYSREC04 DD DISP=SHR,DSN=XX_QUALDS.RES5
//DSNUPROC.SYSREC05 DD DISP=SHR,DSN=XX_QUALDS.RES6
//DSNUPROC.SYSREC06 DD DISP=SHR,DSN=XX_QUALDS.RES7
//DSNUPROC.SYSREC07 DD DISP=SHR,DSN=XX_QUALDS.RES8
//DSNUPROC.SYSREC08 DD DISP=SHR,DSN=XX_QUALDS.RES9
//DSNUPROC.SYSREC09 DD DISP=SHR,DSN=XX_QUALDS.RESC
//DSNUPROC.SYSREC10 DD DISP=SHR,DSN=XX_QUALDS.RESC
//DSNUPROC.SYSREC11 DD DISP=SHR,DSN=XX_QUALDS.RESC
//DSNUPROC.SYSREC1 DD DSN=XX_QUALDS.SYSUT1,

Figure 101 (Part 3 of 5). The ERASPLAN job
Erasing Plans Using the DB2 Environment

// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,,ROUND),
// UNIT=SYSDA

//DSNUPROC.SORTOUT DD DSN=XX_QUALDS.SORTOUT,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,,ROUND),
// UNIT=SYSDA

//DSNUPROC.SYSERR DD DSN=XX_QUALDS.SYSERR,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,,ROUND),
// UNIT=SYSDA

//DSNUPROC.SYSMAP DD DSN=XX_QUALDS.SYSMAP,
// DISP=(MOD,DELETE,CATLG),
// SPACE=(16384,(20,20),,,ROUND),
// UNIT=SYSDA

//DSNUPROC.SYSIN DD *

LOAD DATA LOG NO INDDN SYSREC00 REPLACE INTO TABLE
  XX_OWNER.NVDM_PLAN

................

LOAD DATA LOG NO INDDN SYSREC02 REPLACE INTO TABLE
  XX_OWNER.NVDM_PHASE_NODE

................

LOAD DATA LOG NO INDDN SYSREC02 REPLACE INTO TABLE
  XX_OWNER.NVDM_PHASE_NODE

................

LOAD DATA LOG NO INDDN SYSREC03 REPLACE INTO TABLE
  XX_OWNER.NVDM_FUNCTION

................

LOAD DATA LOG NO INDDN SYSREC04 REPLACE INTO TABLE
  XX_OWNER.NVDM_RESOURCE5

................

LOAD DATA LOG NO INDDN SYSREC05 REPLACE INTO TABLE
  XX_OWNER.NVDM_RESOURCE6

................

LOAD DATA LOG NO INDDN SYSREC06 REPLACE INTO TABLE
  XX_OWNER.NVDM_RESOURCE7

................

LOAD DATA LOG NO INDDN SYSREC07 REPLACE INTO TABLE
  XX_OWNER.NVDM_RESOURCE8

................

LOAD DATA LOG NO INDDN SYSREC08 REPLACE INTO TABLE
  XX_OWNER.NVDM_RESOURCE9

................

Figure 101 (Part 4 of 5). The ERASPLAN job
Erasing Plans Using the DB2 Environment

LOAD DATA LOG NO INDDN SYSREC09 REPLACE INTO TABLE
XX_OWNER.NVDMRESOURCEB

.............

LOAD DATA LOG NO INDDN SYSREC10 REPLACE INTO TABLE
XX_OWNER.NVDMRESOURCEC

.............

LOAD DATA LOG NO INDDN SYSREC11 REPLACE INTO TABLE
XX_OWNER.NVDMRESOURCED

.............

/ * 
// ENDTEST ENDF 
// * 
// TESTRC IF (RC<=4) THEN 
// * 
/ * ************************************************************
REPAIR TABLESPACES 
// * *************************************************************
// *
// UTIL EXEC DSNUPROC,SYSTEM=DSN1,UID='TEMP',UTPROC='' 
// STEPLIB DD DSN=XX_DB2LOAD,DISP=SHR 
// *
// DSNUPROC.SYSIN DD *
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFPLA NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFPHA NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFPHN NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFFUN NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRE5 NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRE6 NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRE7 NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRE8 NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRE9 NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFREB NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFRED NOCOPYPEND
REPAIR OBJECT LOG NO SET TABLESPACE XX_DBNAME.TSTCFREC NOCOPYPEND

/ * 
// ENDTEST ENDF 

Figure 101 (Part 5 of 5). The ERASPLAN job
Chapter 18. Maintaining NetView DM for MVS Files

This chapter shows you how to maintain the following NetView DM for MVS files:

- Resource repository
- Plan library
- Distributed resource directory (DRD)
- Transmission control file (TCF)
- Unsolicited message file
- NetView DM for MVS work file
- Exchange information file (EIF)
- TCP work file
- Request queue file
- GIX data sets
- BQBLIBI file

You maintain NetView DM for MVS files to prevent them from running out of file space. When a file is getting full, NetView DM for MVS generates a warning message. When a file is full, NetView DM for MVS generates an error message and halts.

This chapter shows you how to use the following Batch Utilities to maintain these files:

- PLCOPY
- DSXBUBK
- DSXBURS

Procedure for Maintaining NetView DM for MVS Files

Use the steps outlined below as a general guideline for maintaining NetView DM for MVS files. Each file type is explained in more detail in the rest of this chapter.

Step 1. Use the QUERY SPACE control statement to find out how much space is left in either the resource repository or the plan library.

Step 2. Find out how much space the VSAM file requires. The values (indicated in cylinders and tracks) are intended for the 3380 device type.

Step 3. Use the JCL files shown for each file in this chapter to allocate new VSAM files. These files are similar to the files which are produced when you install NetView DM for MVS, as described in the Installation and Customization book.

Note

The examples given in this chapter are designed to run without the NetView DM for MVS service provider option. The examples that are affected are preceded by a note, describing what you need to change to run with the service provider.

Make sure that you use the file names that are shown in the sample figures. You can modify the figures slightly to define the size of the file you require. You cannot modify the value of the CISZ parameter.

Step 4. If you are working on the plan library or the resource repository, copy the contents of the old file into the new file which you created in the previous step. You will need Maintain repository and plan library authorization to do this.
Maintaining the Resource Repository (System Administrator)

To copy the plan library, use the PLCOPY command. To back up and restore the repository, use DSXBUBK and DSXBURS.

All of the NetView DM for MVS files are VSAM files. They are preloaded with a null record that consists of a NetView DM for MVS record ID, with the remaining fields filled with zeros. NetView DM for MVS system generation produces the JCL to write this null record automatically. This record is not retrieved during NetView DM for MVS processing. It is used so that NetView DM for MVS does not need to insert records in LOAD mode.

Maintaining the Resource Repository

The resource repository stores resources that are waiting to be sent to nodes, and resources that have been retrieved from nodes. The repository consists of two separate data entities, the holding file and the NetView DM for MVS library. These consist of four data sets: the holding file directory and data space, and the NetView DM for MVS library directory and data space.

Table 47 on page 430 and Table 48 on page 430 show the VSAM characteristics of the NetView DM for MVS library and of the holding file:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>RRDS</td>
<td>NVDMS.NDM162AA.NDMLIBT</td>
<td>F 4080</td>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td>Directory</td>
<td>KSDS</td>
<td>NVDMS.NDM162AA.NDMHFD</td>
<td>F 1012</td>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

Table 47. VSAM Characteristics of the Library

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>RRDS</td>
<td>NVDMS.NDM162AA.NDMHFD</td>
<td>F 4080</td>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td>Directory</td>
<td>KSDS</td>
<td>NVDMS.NDM162AA.NDMHFD</td>
<td>F 1012</td>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

Table 48. VSAM Characteristics of the Holding File

Space Requirements

The data portions of the library and of the holding file contain the resources. The amount of space required for the data portions is directly proportional to the number and size of the resources stored in the repository.

The directories contain information that identifies resources and controls their processing. Each directory entry for each resource requires one 1012 byte record.

When you allocate a holding file as close as possible to the VSAM limit of 4GB, using only primary space allocation on a multivolume, the initialization of the data portion of the holding file produces the message:

IEC070I 034-204

This message indicates that the holding file has been initialized successfully and is therefore available. For example, a VSAM 4GB limit translates to 6990 cylinders. You can generate a holding file by allocating 2330 cylinders as primary space over three volumes.
Maintaining the Resource Repository (System Administrator)

Using Extended Addressability

It is possible to extend the maximum size of the SMS Resource Repository above the previous limits of 4 GB. By taking advantage of the XRBA architecture the maximum size has been raised to 400 GB, the limitations of the initialization procedure making it impossible to realize the theoretical maximum of $2^{64}$ bytes.

You can use Extended Addressability with either service provider in a non-Sysplex environment or SMSVSAM in a Sysplex environment, but in either case you must be using DFSMS Version 1.5 or later.

To enable Extended Addressability you will need to customize SMS, following these steps:

Step 1. If your NetView DM for MVS Resource Repository is NOT managed by SMS:
   a. Define an SMS Storage Class with a specific suffix (e.g. NVDMS)
   b. Define an SMS Storage Group (e.g. NVDMSTOR) and assign it to the designated volumes.

Step 2. Define an SMS Data Class (e.g. XRBA1) with the options:

<table>
<thead>
<tr>
<th>Data Set Name Type</th>
<th>EXTENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Extended</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Extended Addressability</td>
<td>YES</td>
</tr>
<tr>
<td>Record Access Bias</td>
<td>USER</td>
</tr>
<tr>
<td>Reuse</td>
<td>NO</td>
</tr>
<tr>
<td>Initial Load</td>
<td>RECOVERY</td>
</tr>
</tbody>
</table>

Step 3. Activate the SMS Defined Objects

Instructions for increasing the size of the resource repository library or holding file to take advantage of Extended Addressability are given in “Reorganizing the Library or Holding File Using Extended Addressability” on page 437.

What to Do If the File Is Full

If the data or directory space is full, you receive an error message like this:

NDM2242T DIRECTORY OR DATA SPACE FULL FOR HOLDING FILE

There are two ways that you can recover from this situation. You can:

- Delete some resources that you no longer require. Before you delete the resources, you can print them or unload them into user data sets. (See Chapter 15, “Managing Resources in the Repository” on page 261.)

- Allocate a file with more space, and restore the contents of the repository using DSXBUBK and DSXBURS.

If the resource repository becomes full while transmission activities are taking place (for example, a resource is being written because of a RETRIEVE function or an NSR), the TCP shuts down and you receive these error messages:

NDM1309W PERFORMANCE DEGRADATION ON FILE file type.
NDM0824E THERE IS NO MORE SPACE IN FILE file name.

You must make more space available in the resource repository before restarting the TCP.
After NetView DM for MVS has been running for some time, the space used in the library or holding file becomes fragmented. The data space is split across several small extents, which causes performance degradation. When performance is degraded, you receive the NDM1309W warning message shown above.

There now follows a description of how to unload your data, initialize a new library or data holding file and reload your data, firstly for when you are not using Extended Addressability ("Reorganizing the Library or Holding File Without Using Extended Addressability") and then for when you are using Extended Addressability ("Reorganizing the Library or Holding File Using Extended Addressability" on page 437).

In addition, there is also a description of how to use the program DSXBIHLF to achieve a fast reinitialization of a fragmented holding file (see “Fast Reinitialization Using DSXBIHLF” on page 440).

**Reorganizing the Library or Holding File Without Using Extended Addressability**

To reorganize the data in the library or holding file, follow these instructions:

**Step 1.** Use DSXBUBK to unload the data into a sequential file on the host (see “DSXBUBK and DSXBURS” on page 442).

**Step 2.** Use DSXBIVIN to define and initialize a new library or holding file without using Extended Addressability

Figure 102 shows the statements required to define and initialize a new library. Figure 103 on page 435 shows the statements required to define and initialize a new holding file.

**Step 3.** Use DSXBURS to load the data back into the library or holding file (see “DSXBUBK and DSXBURS” on page 442).

**Library JOB Sample Without Extended Addressability**

**Note**

This sample is designed to run without the service provider option and in a non-Sysplex (SMVSSAM) environment. If you want to run with the service provider or Sysplex (SMVSSAM) change the highlighted parameters to the values in square brackets [] on the right of the example.

```
//DEFLIBD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=338/VOL=SER=NDMXA6,DISP=SHR
//SYSIN DD /
DELETE (NVDM.V162.NDMLIB) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDM.V162.NDMLIB) -
    SHR(4 3) -
    NOWRITECHECK) - [SHR(2 3) -]
```

Figure 102 (Part 1 of 3). Sample JCL to Define and Initialize the NetView DM for MVS Library
Figure 102 (Part 2 of 3). Sample JCL to Define and Initialize the NetView DM for MVS Library
Maintaining the Resource Repository (System Administrator)

Figure 102 (Part 3 of 3). Sample JCL to Define and Initialize the NetView DM for MVS Library

Notes:

- In this example only the library NDMHLIBT is changed with respect to the base resource repository; the NDMHLIB is unchanged.

- The parameter 'PARM='1000000'' means that this initialization will allow the creation of up to one million records. Thus, the space available will be 4 GB (one million records at four KB each) using extents of 200 cylinders each. The initialization routine will allocate the first extent where it inserts the control record. The remaining space will be allocated by VSAM as needed using normal VSAM procedures. 4 GB is the maximum size possible without using Extended Addressability.
Holding File JOB Sample Without Extended Addressability

**Note**

This sample is designed to run **without** the service provider option and in a non-Sysplex (SMSVSAM) environment. If you want to run **with** the service provider or Sysplex (SMSVSAM) change the highlighted parameters to the values in square brackets [] on the right of the example.

```lisp
//DEFHFD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=3380, VOL=SER=NDMXA6, DISP=SHR
//SYSIN DD *
DELETE (NVDM.V162.NDMHFDI) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDM.V162.NDMHFDI) -
  | SHR(4 3) -
  | NOWRITECHECK) -
  | INDEX (NAME(NVDM.V162.NDMHFDI_INDEX) -
  | UNIQUE -
  | FILE(VOLDD1) -
  | VOL(NDMXA6) -
  | [SHR(2 3) -]
  | [TRK(2 2) -]
  | [TRK(5 5) -]
  | TRK(2) -
  | CISZ(1024) -
  | DATA (NAME(NVDM.V162.NDMHFDI_DATA) -
  | UNIQUE -
  | FILE(VOLDD1) -
  | VOL(NDMXA6) -
  | KEYS(112,0) -
  | CISZ(1024) -
  | [TRK(5 5) -]
  | RECSZ(1012,1012) -
  | CISZ(1024) -
  | CATALOG(CATALOG.MVSICF1.VNDMXA2) ;
/*
/*
//HFDINIT EXEC PGM=DSXBIVIN
//OUTD DD DSN=&&TEMP, DISP=(NEW,PASS),
//UNIT=SYSDA, SPACE=(TRK,1)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
HF
1012
/*
//HFDLOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
REPRO -
INFILE(INPUT1) -
OUTFILE(OUTPUT1)
//OUTPUT1 DD DSN=NVD.M.V162.NDMHFDI, DISP=(OLD)
//INPUT1 DD DSN=&&TEMP, DISP=(OLD,DELETE)
/*
```

Figure 103 (Part 1 of 2). Sample JCL to Define and Initialize the Holding File
Maintaining the Resource Repository (System Administrator)

//DEFHF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=3380,DISP=SHR,
// VOL=SER=NDMXA6
//SYSSIN DD *
DELETE (NVDM.V162.NDMHFDA) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDM.V162.NDMHFDA) -
| | [SHR(4 3) -
| | NOWRITECHECK -
| | FILE(VOLDD1) -
| | VOL(NDMXA6 -
| | -
| | -
| | -
| | -
| | -
| |
| |
| |
| |)
| | UNIQUE -
| | [CYL(3 3) -
| | RECSZ(4080,4080) -
| | CISZ(4096) -
| | NUMBERED) -
| | DATA (NAME(NVDM.V162.NDMHFDA.DATA)) -

Figure 103 (Part 2 of 2). Sample JCL to Define and Initialize the Holding File

Notes:

- In this example only the library NDMHLIBA is changed with respect to the base resource repository; the NDMHLIBI is unchanged.

- The parameter ‘PARM=’1000000’ means that this initialization will allow the creation of up to one million records. Thus, the space available will be 4 GB (one million records at four KB each) using extents of 200 cylinders each. The initialization routine will allocate the first extent where it inserts the control record. The remaining space will be allocated by VSAM as needed using normal VSAM procedures. 4 GB is the maximum size possible without using Extended Addressability.
Reorganizing the Library or Holding File Using Extended Addressability

To reorganize the data in the library or holding file, follow these instructions:

**Step 1.** Use DSXBUBK to unload the data into a sequential file on the host (see “DSXBUBK and DSXBURS” on page 442).

**Step 2.** Use DSXBIVIN to define and initialize a new library or holding file using Extended Addressability. Figure 104 shows the statements required to define and initialize a new library. Figure 105 on page 439 shows the statements required to define and initialize a new holding file.

**Step 3.** Use DSXBURS to load the data back into the library or holding file (see “DSXBUBK and DSXBURS” on page 442).

**Library JOB Sample With Extended Addressability:** This sample job can be used either with the service provider in a non-Sysplex environment, or with SMSVSAM in a Susplex environment.

```
//DEFLIBD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DELETE (NVDMS.NDM162AA.NDMLIB) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDMS.NDM162AA.NDMLIB) -
    SHR(2 3) -
    NOWRITECHECK) -
INDEX (NAME(NVDMS.NDM162AA.NDMLIB.INDEX) -
    UNIQUE -
    CYL(1 1) -
    CISZ(4096) -
DATA (NAME(NVDMS.NDM162AA.NDMLIB.DATA) -
    UNIQUE -
    KEYS(112,0) -
    CYL(5 5) -
    RECSZ(1012,1012) -
    CISZ(4096)) -
/*
/*
//LIBDINIT EXEC PGM=DSXBIVIN
//OUTD DD DSN=&&TEMP,DISP=(NEW,PASS),
//    UNIT=SYSDA,SPACE=(TRK,1)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
LB
1012
/*
```

*Figure 104 (Part 1 of 2). Sample JCL to Define and Initialize the NetView DM for MVS Library Using Extended Addressability*
Maintaining the Resource Repository (System Administrator)

```
//LIBDLOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
  REPRO -
    INFIL(INPUT1) -
    OUTFIL(OUTPUT1)
//OUTPUT1 DD DSN=NVDMS.NDM162AA.NDMLIB,DISP=OLD
//INPUT1 DD DSN=&&TEMP,DISP=(OLD,DELETE)
///c5197
//DEFLIB EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD /c5197
DELETE (NVDMS.NDM162AA.NDMLIBT) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDMS.NDM162AA.NDMLIBT) -
  SHR(2 3) -
  DATACLASS(XRBA1) -
  NOWRITECHECK -
  UNIQUE -
  CYL(200 200) -
  RECSZ(2040,4080) -
  CISZ(4096) -
  NUMBERED) -
DATA (NAME(NVDMS.NDM162AA.NDMLIBT.DATA)) -
//DSXBILIX EXEC PGM=DSXB1HLX,
  PARM='2000000'
//DSXDDND DD DSN=NVDMS.NDM162AA.NDMLIB,DISP=SHR
//DSXDDN DD DSN=NVDMS.NDM162AA.NDMLIBT,DISP=SHR
//SYSPRINT DD SYSOUT=A
///c5197
```
Figure 105 (Part 1 of 2). Sample JCL to Define and Initialize the Holding File with Extended Addressability
Maintaining the Resource Repository (System Administrator)

```
//DSXBIHFI EXEC PGM=DSXBIHLI,
PARM='2000000'
//DSXDDND DD DSN=NVDM5.NDM162AA.NDMHFDI,DISP=SHR
//DSXDDN DD DSN=NVDM5.NDM162AA.NDMHFDA,DISP=SHR
//SYSPRINT DD SYSOUT=A
///c5197
```

Figure 105 (Part 2 of 2). Sample JCL to Define and Initialize the Holding File with Extended Addressability

Notes:

- In this example only the library NDMHFDA is changed with respect to the base resource repository; the NDMHFDI is unchanged.
- The parameter 'RECSZ(2040,4080)' is required to force the creation of a VRDDS file, as required for the correct usage of VSAM XRBA with LSR.
- The parameter 'PARM='2000000'' means that this initialization will allow the creation of up to two million records. Thus, the space available will be 8 GB (two million records at four KB each) using extents of 200 cylinders each. The initialization routine will allocate the first extent of 200 cylinders where it inserts the control record. The remaining space will be allocated by VSAM as needed using normal VSAM procedures. If you require more than 8 GB increase the PARM value and the extents size, accordingly, up to a maximum of 400 GB.
- The DATACLASS parameter in the DEFINE CLUSTER macro takes as its attribute the name of the extended SMS Data Class that you defined, in this case XRBA1. (see “Using Extended Addressability” on page 431 for details).

Fast Reinitialization Using DSXBIHLF

If the data space of the existing holding file or library file has become fragmented, you can reinitialize the file. Using the program called DSXBIHLF, you can re-initialize the data part of the existing holding file or library file in a faster way.

To do this, follow the instructions described below, making sure that the file to be re-initialized has already been initialized at least once, as shown in Figure 103 on page 435. Once the file has been initialized at least once, you can use the JCL statements shown below instead of the ones shown in Figure 103 on page 435.

The JCL shown in Figure 106 on page 441 executes the following steps:

1. Deletes the old directory and defines the new directory space (step DEFHFD)
2. Initializes the new directory (step HFDINIT)
3. Executes the step HFDLOAD to complete the initialization of the directory
4. Executes the program called DSXBIHLF which carries out the fast reinitialization.

The differences between the JCL to fast re-initialize the holding file and the JCL shown in Figure 103 on page 435 are as follows:

- The step DEFHF is not executed
- The name of the program invoked by the last EXEC statement for re-use (DSXBIHLF) is different from the one invoked when defining and initializing a new holding file (DSXBIHLI).

Here is the sample JCL to fast re-initialize the data part of an existing holding file. This sample also applies to the library file, if the appropriate changes are made to the JCL.
Note

This sample runs **without** the service provider option and in a non-Sysplex (SMVSAM) environment.

```plaintext
//DEFHFD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*  
//VOLDD1 DD UNIT=SYSDA,VOL=SER=NDM501,DISP=SHR
//SYSIN DD *
DELETE (NDM13.NDMHFDI) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDM.V162.NDMHFDI) -
  SHR(4, 3) -
  WRITECHECK) -
  INDEX (NAME(NVDM.V162.NDMHFDI.INDEX) -
    UNIQUE -
    FILE(VOLDD1) -
    VOL(NDM501) -
    TRK(30) -
    CISZ(1024) -
  )
  DATA (NAME(NVDM.V162.NDMHFDI.DATA) -
    UNIQUE -
    FILE(VOLDD1) -
    VOL(NDM501) -
    KEYS(112,0) -
    TRK(90) -
    RECSZ(1012,1012) -
    CISZ(1024) -
  )
  CATALOG(NDM501.NDMVSM.ucat) ;
/*
  
  */
//HFDINIT EXEC PGM=DSXBIVIN
//OUTD DD DSN=&TEMP,DISP=(NEW,PASS),
// UNIT=SYSDA,SPACE=(TRK,1)
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
HF
1012
/*
```

Figure 106 (Part 1 of 2). Sample JCL to Fast Re-Initialize the Data Part of an Existing Holding File
Maintaining the Resource Repository (System Administrator)

There are two utility programs that you can use to maintain the resource repository:

- **DSXBUBK** produces a sequential file that contains the contents of the holding file or the library and reclaims the free space. The output file can be on a disk or a tape.

- **DSXBURS** restores the sequential file created by the DSXBUBK program in a new holding file or library. You must already have defined and initialized a new repository, using the JCL/EXEC statements shown in Figure 103 on page 435.

**Attention**

DSXBUBK and DSXBURS require **exclusive control** of the repository. If any other NetView DM for MVS activity takes place between the execution of DSXBUBK and DSXBURS, the repository may not be consistent with other NetView DM for MVS files when it is restored (for example, the DRD and the TCF).

To use the DSXBUBK and DSXBURS programs with the service provider, leave the service provider active and specify DISP=SHR in the JCL utility programs. The service provider then checks if other users are accessing the VSAM files that are used in the JCL utility programs; if so, the request is rejected. While the DSXBUBK and DSXBURS programs are running, the service provider prevents other users from accessing the VSAM files that are used in the JCL utility programs.

**Note:** To maintain the holding file or library, you must specify in the PARM statement:

- USERID=userid, a 1-8 character name of a NetView DM for MVS user who is authorized to use this utility.

- One of the following for the REPOSITORY parameter:
  - HF, for the holding file
  - LB, for the library
  - ALL, for both the holding file and library

Figure 107 on page 443 shows the files you need to run DSXBUBK and DSXBURS. The files for DSXBUBK and for DSXBURS are exactly the same, except that you specify DSXTOUT for DSXBUBK, and DSXTIN for DSXBURS.
//MYJOB JOB (1717,1717),'PGM222'
// MSGLEVEL=(1,1),
// MSGCLASS=A,
// REGION=2048K,
// CLASS=M
//JOBLIB DD DSN=NVDM.V162LOADLIB,DISP=SHR
//JOBCAT DD DSN=NVDM.V162UCAT,DISP=SHR
//STEP1 EXEC PGM=DSXBUBK,
// PARM='USERID=xxxxxxxx,REPOSITORY=ALL'
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DSXDRD DD DSN=NVDM.V162.NDMDRD,DISP=SHR
//DSXHFDI DD DSN=NVDM.V162.NDMHFDI,DISP=(OLD)
//DSXHFDA DD DSN=NVDM.V162.NDMHFDA,DISP=(OLD)
//DSXLIB DD DSN=NVDM.V162.NDMLIB,DISP=(OLD)
//DSXLIBT DD DSN=NVDM.V162.NDMLIBT,DISP=(OLD)
//DSXTOUT DD DSN=TDT/zerodot46BACKHFD,UNIT=338,DISP=(,KEEP),
// DCB=(DSORG=PS,RECFM=VB,LRECL=4096,BLKSIZE=4100),
// VOL=SER=SDBD/zerodot1,SPACE=(CYL,(5,5),RLSE)
/

Figure 107. Sample JCL to Invoke DSXBUBK. This example shows how to back up both the NetView DM for MVS holding file, and the NetView DM for MVS library.

The JCL DD statements used in the above example are explained in Table 49.

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBLIB</td>
<td>System library of load modules for the job</td>
</tr>
<tr>
<td>JOBCAT</td>
<td>VSAM catalog for the job</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Message file</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Dump print file</td>
</tr>
<tr>
<td>DSXDRD</td>
<td>Distributed resource directory</td>
</tr>
<tr>
<td>DSXHFDI</td>
<td>Holding file directory</td>
</tr>
<tr>
<td>DSXHFDA</td>
<td>Holding file data repository</td>
</tr>
<tr>
<td>DSXLIB</td>
<td>NetView DM for MVS library directory</td>
</tr>
<tr>
<td>DSXLIBT</td>
<td>NetView DM for MVS library data repository</td>
</tr>
<tr>
<td>DSXTOUT</td>
<td>DSXBUBK output file</td>
</tr>
<tr>
<td>DSXTIN</td>
<td>DSXBURS input file</td>
</tr>
</tbody>
</table>

Table 49. JCL DD Statements Used by DSXBUBK and DSXBURS
Maintaining the Plan Library

The plan library stores transmission plans created by users. It consists of two data sets:

- The plan library directory, which contains the plan name, user id, creation date and time, last modification date and time, number of phases, and a time stamp for each plan.
- The plan library data portion, which stores the actual contents of the plans.

These data sets have the following VSAM characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>KSDS</td>
<td>NVDM.V162.NDMGIX</td>
<td>V 100 to 400</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>Data</td>
<td>RRDS</td>
<td>NVDM.V162.NDMG IX D</td>
<td>F 4080</td>
<td>10</td>
<td>150</td>
</tr>
</tbody>
</table>

Table 50. VSAM Characteristics of the Plan Library

If you try to insert a new plan into the plan library and the data space allocated for the plan library is full, you receive an error message like this:

```
NDM8044E PLAN PAYROLL NOT SAVED BECAUSE PLAN LIBRARY IS FULL
```

To recover from this situation, you can:

1. Erase one or more unused plans from the library, either using a single command or the SUBMIT utility. If the space that you free in this way is greater than the space needed to insert the new plan, there may be a problem, due to the way in which VSAM manages a KSDS cluster. You can only insert the new item into the library if the VSAM key of the item(s) you just erased is greater than or equal to (in terms of collating sequence) the key of the item you are trying to insert.

2. If the free space you created by erasing plans is not enough, and you still receive the error message, you can allocate a new plan library the same size as the old one. If the plan library becomes full frequently, allocate more space for the new plan library.

   You allocate a new library using the IDCAMS DEFINE CLUSTER command. IDCAMS is a VSAM utility.

   You use PLCOPY to copy all your plans into the new library. This compacts the records stored in the plan library by eliminating the places taken up by records that were erased during the entire existence of the plan library.

### Space Requirements

The amount of space required by the library depends upon the contents of your transmission plans. Each element in a plan needs the following space:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>For each plan</td>
<td>151 bytes</td>
</tr>
<tr>
<td>For each phase in the plan</td>
<td>201 bytes</td>
</tr>
<tr>
<td>For each destination (phase-by-node)</td>
<td>29 bytes</td>
</tr>
<tr>
<td>For each function</td>
<td>137 bytes</td>
</tr>
<tr>
<td>For each begin/end sequence</td>
<td>137 bytes</td>
</tr>
<tr>
<td>For each resource</td>
<td>416 bytes</td>
</tr>
</tbody>
</table>

Table 51. Space Required by the Plan Library Data Portion

**Note:** The maximum size of a plan in the plan library is 127MB.
Figure 108 on page 445 shows the statements required to define and initialize a new plan library.

![Sample JCL to Define and Initialize the Plan Library]

*Figure 108 (Part 1 of 2). Sample JCL to Define and Initialize the Plan Library*
Figure 108 (Part 2 of 2). Sample JCL to Define and Initialize the Plan Library

```jcl
DEFINE CLUSTER (NAME(NVDM.V162.NDMGIXD) -
                 SHR(4 3) -
                 NOWRITECHECK -
                 FILE(VOLDD1) -
                 VOL(A33355 -
                 -
                 -
                 -
                 -
                 -
                 ) -
                 UNIQUE -
                 CYL(1) -
                 RECSZ(4080,4080) -
                 CISZ(4096) -
                 NUMBERED) -
                 DATA (NAME(NVDM.V162.NDMGIXD.DATA)) ;
//DSXBIGIP EXEC PGM=DSXBIGIP,PARM='NDM15'
//DSXGIX DD DSN=NVDM.V162.NDMGIX,DISP=SHR
//DSXGIXD DD DSN=NVDM.V162.NDMGIXD,DISP=SHR
//SYSPRINT DD SYSOUT=A
/*
```
PLCOPY

PLCOPY is the utility that you use to maintain the plan library. Use PLCOPY to copy all of the plans, or a selected plan, from the current plan library into the new library that you have already defined and initialized, using the statements shown in Figure 108 on page 445. If a plan already exists in the output library, the plan is replaced.

Notes:

1. You cannot copy plans if there is an inconsistent initialization between the plan libraries. Starting from NetView DM for MVS Release 4, the plan library is initialized with a different format from that used previously. The PLCOPY utility cannot copy plans stored in a plan library initialized by NetView DM for MVS Release 4 or higher to a plan library initialized with previous NetView DM for MVS versions.

2. If you use the PLCOPY utility to copy plans into a NetView DM for MVS Release 4 plan library, plans that are greater than 256KB may not be copied successfully.

Here are the JCL DD statements that PLCOPY uses:

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBLIB</td>
<td>Load module system library for the job</td>
</tr>
<tr>
<td>JOBCAT</td>
<td>VSAM catalog for the job</td>
</tr>
<tr>
<td>SYSPRINT</td>
<td>Message file</td>
</tr>
<tr>
<td>SNAP</td>
<td>Snap print file</td>
</tr>
<tr>
<td>SYSUDUMP</td>
<td>Dump print file</td>
</tr>
<tr>
<td>DSXDRD</td>
<td>Distributed resource directory</td>
</tr>
<tr>
<td>INPGIX</td>
<td>Input plan library directory</td>
</tr>
<tr>
<td>INPGIXD</td>
<td>Input plan library data</td>
</tr>
<tr>
<td>OUTGIX</td>
<td>Output plan library directory</td>
</tr>
<tr>
<td>OUTGIXD</td>
<td>Output plan library data</td>
</tr>
<tr>
<td>SYSIN</td>
<td>Card image input file that contains the DUPLICATE PLANS and END statements</td>
</tr>
<tr>
<td>BATCHPW</td>
<td>Contains user password if not specified in PARM stream</td>
</tr>
</tbody>
</table>

Table 52. JCL DD Statements Used by PLCOPY

Using PLCOPY, you must include the DUPLICATE PLANS statement, as well as the usual END statement, at the end of the job.
DUPLICAtE PLANS

Use the DUPLICAtE PLANS statement of PLCOPY to copy plans from the current plan library into a new plan library.

Syntax

<table>
<thead>
<tr>
<th>Statements</th>
<th>Operands</th>
<th>Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUPLICAtE PLANS</td>
<td>[PLAN= name]</td>
<td>All node types</td>
</tr>
</tbody>
</table>

NAME Specify the name of the plan to be copied. If you do not enter a name, then all the plans in the plan library will be copied. You can specify 1-8 alphanumeric characters. The first character must be alphabetic.

Figure 109 shows the statements you can use to invoke the PLCOPY batch utility.

```
//MYJOB JOB (1717,1717), 'PGM222',
// MSGLEVEL=(1,1),
// MSGCLASS=A,
// REGION=2048K,
// CLASS=M
//JOBLIB DD DSN=NVDM.V162.LOADLIB,DISP=SHR
//JOBCAT DD DSN=NVDM.V162.UCAT,DISP=SHR
//STEP1 EXEC PGM=DSXPREP,
// PARM='FUNCTION=PLCOPY,USERID=xx,PROFILE=zz'
//SYSPRINT DD SYSOUT=A
//SNAP DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DSXDRD DD DSN=NVDM.V162.NDMDRD,DISP=SHR
//INPGIXD DD DSN=NVDM.V162.NDMGIXD,DISP=(OLD)
//INPGIX DD DSN=NVDM.V162.NDMGIX,DISP=(OLD)
//OUTGIXD DD DSN=NVDM.V162.NEWGIXD,DISP=(OLD)
//OUTGIX DD DSN=NVDM.V162.NEWGIX,DISP=(OLD)
//BATCHPW DD DSN=NVDM.V162.PSW,DISP=(OLD)
//SYSIN DD *
 DUPLICAtE PLANS
 END
 /*
```

Figure 109. Sample JCL to Invoke PLCOPY
Maintaining the Distributed Resource Directory

The distributed resource directory (DRD) contains information about NetView DM for MVS node definitions, resource definitions and assignments, resource tracking information, node and resource group definitions, and user profiles. It consists of a single data set with the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>KSDS</td>
<td>NVM.V162.NDMDRD</td>
<td>V 500 to 4080</td>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

*Table 53. VSAM Characteristics of the DRD*

If you try to insert an item into the DRD and the data space allocated for the DRD is full, you receive an error message. An item refers to one of the following:

- A new user
- A new node
- A new resource definition
- A new resource history record
- A new group of resources or nodes.

For example, suppose you are defining a new user and the DRD is full. You receive a message like this:

```
NDM9089E NEW USER NOT INSERTED IN THE DRD BECAUSE IT IS FULL
```

To recover from this situation, you can:

1. Erase one or more users, nodes, resource definitions, resource history records, resource groups, or node groups that are no longer used from the DRD. You can either use a single command or the SUBMIT utility. If the space that you free in this way is greater than the space needed to store the new item, there may be a problem, due to the way in which VSAM manages a KSDS cluster. You can only insert the new item into the DRD if the VSAM key of the item you just erased is greater than or equal to (in terms of collating sequence) the key of the item you are trying to insert.

2. If the free space you created by erasing items is not enough, and you still receive the error message, you can allocate a new DRD the same size as the old one. If the DRD becomes full frequently, allocate more space for the new DRD.

You allocate a new DRD using the IDCAMS DEFINE CLUSTER command (IDCAMS is a VSAM utility). Run IDCAMS with a REPRO command to copy the records from the old DRD into the new DRD. This compacts the records stored in the DRD by eliminating the places taken up by records that were erased during the entire existence of the DRD.
Space Requirements

This depends on the number of nodes, resources, resource assignments, group definitions, user profiles, and resource history records that are contained in the DRD. The space needed for each of these is:

<table>
<thead>
<tr>
<th>Description</th>
<th>Space Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each node</td>
<td>1236 bytes</td>
</tr>
<tr>
<td>For each resource definition</td>
<td>264 bytes</td>
</tr>
<tr>
<td>For each resource assigned to a node</td>
<td>264 bytes</td>
</tr>
<tr>
<td>For each resource history record</td>
<td>264 bytes</td>
</tr>
<tr>
<td>For each group composed of n nodes</td>
<td>152 + (12 * n) bytes</td>
</tr>
<tr>
<td>For each group composed of n resources</td>
<td>152 + (96 * n) bytes</td>
</tr>
<tr>
<td>For each user</td>
<td>904 bytes</td>
</tr>
</tbody>
</table>

Table 54. Space Required by the DRD

Figure 110 shows the statements required to define and initialize a new DRD.

Note

- This sample is designed to run without the service provider option and in a non-Sysplex (SMSVSAM) environment. If you want to run with the service provider or Sysplex (SMSVSAM) change the highlighted parameter to SHR(2 3) -.
- If you use the IDCAMS REPRO command, you only need to define the cluster. Do not include the DRDINIT, DRDLOAD, and DSXBIDRD statements shown in Figure 110.

```
//DEFDRD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=SYSDA,VOL=SER=NDM3/zerodot1,DISP=SHR
//SYSIN DD *
DEFINE CLUSTER (NAME(NVDM.V162.NEWDRD) -
                  SHR(4 3) -
                  NOWRITECHECK) -
                  INDEX (NAME(NVDM.V162.NEWDRD.INDEX) -
                          FILE(VOLDDI) -
                          VOL(NDM301) -
                          CYL(1) -
                          CISZ(4096) -
                          DATA (NAME(NVDM.V162.NEWDRD.DATA) -
                                  FILE(VOLDDI) -
                                  VOL(NDM301) -
                                  KEYS(104,0) -
                                  CYL(3) -
                                  RECSZ(500,4080) -
                                  CISZ(4096) -
                                  CATALOG(NDM301.NDMVSU.UCAT) ;
```

Figure 110 (Part 1 of 2). Sample JCL to Define and Initialize the DRD
Figure 110 (Part 2 of 2). Sample JCL to Define and Initialize the DRD
## Maintaining the TCF

The transmission control file (TCF) contains submitted transmission plans together with status information. It consists of a single data set, with the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>KSDS</td>
<td>NVDM.V162.NDMTCF</td>
<td>500 to 16232</td>
<td>3</td>
<td>45</td>
</tr>
</tbody>
</table>

*Table 55. VSAM Characteristics of the TCF*

The TCF is composed of a data space and an index. If you try to insert an item into the TCF and the data space allocated for the TCF is full, you receive this error message:

`NDM8444E PLAN NOT SUBMITTED BECAUSE THE TCF IS FULL`

If the TCF fills up while the TCP is receiving a report, the TCP shuts down and you receive this error message:

`NDM80825T THERE IS NO MORE SPACE AVAILABLE IN THE TCF TO HONOR THE REQUEST`

This means that either the data space or the index is full. You must allocate more space to the TCF before attempting to restart the TCP.

An item can be one of the following:

- A plan, either submitted by GIX or the SUBMIT batch utility
- An API request, originated by an application program
- An NSR received from a node
- A TCP checkpoint record generated by the TCP.

If the data space of the TCF is full, erase one or more plans, NSRs, or API requests, using a single command or the TCFMAINT utility. The VSAM key of the items you erase must be greater (in collating sequence) than that of the item you want to insert.

If the free space you created by erasing items is not enough and you still receive the error message, you can allocate a new TCF. The new TCF can be of the same size as the old one, but if the TCF fills up often, you should make the new TCF larger.

You allocate a new TCF using the IDCAMS `DEFINE CLUSTER` command (IDCAMS is a VSAM utility). Run IDCAMS with a `REPRO` command to copy the records from the old TCF to the new TCF. This compacts the TCF by eliminating the space taken up by records that were erased since the TCF was created.

The TCF *index* can also become full. Each item that is inserted into the TCF has a VSAM key. Each new VSAM key creates a new entry in the TCF index. Even if you delete items from the TCF using TCFMAINT, the key remains in the index, so the index gradually fills up. Before the TCF index reaches its limit, you receive the following warning message:

`NDM1111W TRANSMISSION CONTROL FILE INDEX COMPONENT IS AT RBA SPLIT LIMIT`

At this point the index is at 80% of capacity. Processing continues normally until the index is full, but you will then receive a VSAM error, so you should compact the index when you receive this initial warning message.

To compact the index, allocate a new TCF. If you want to copy the contents of the old TCF into the new TCF, use the IDCAMS `REPR0` command.
If the TCF index fills up regularly, consider revising your use of VSAM keys. A new VSAM key is inserted into the index for each new item that you insert into the TCF. If you reuse the keys, for example, by reusing the names of plans, you can minimize the size of the index.

When you use the REPRO command to allocate a new TCF, do not specify the TCF initialization step (TCFINIT) in the JCL. The TCFINIT step inserts a dummy record into the TCF, but the old TCF that you are reproducing already contains the dummy record. Figure 112 on page 455 shows sample JCL statements to redefine the TCF using the REPRO command.

### Space Requirements

The space required in the TCF depends on the size of the plan, particularly on the types of resources referred to in the plan. Table 57 shows the amount of space required for each resource type at each node type, and also for each different LU 6.2 logical unit addressed by the plan.

<table>
<thead>
<tr>
<th>Table 56. Space Required by the TCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>For each plan</td>
</tr>
<tr>
<td>For each phase in the plan</td>
</tr>
<tr>
<td>For each destination (phase-by-node)</td>
</tr>
<tr>
<td>For each function by destination</td>
</tr>
<tr>
<td>For each resource in each function</td>
</tr>
<tr>
<td>For each node solicited request</td>
</tr>
<tr>
<td>For each node API request</td>
</tr>
<tr>
<td>For each different LU 6.2 logical unit addressed by the plan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>VSE</th>
<th>DPPX</th>
<th>RPS</th>
<th>SSP</th>
<th>PDOS</th>
<th>POSC</th>
<th>CMEP</th>
<th>CMFP</th>
<th>NDMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>AS/400 data object containment structure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
</tr>
<tr>
<td>AS/400 object</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
</tr>
<tr>
<td>Category</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Category update</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>CLIST</td>
<td>156</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Configuration file</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>260</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>DSCB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Dump</td>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
</tr>
<tr>
<td>Errorlog</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Error log</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
</tr>
</tbody>
</table>
Maintaining the TCF (System Administrator)

Table 57. Space Occupied by Each Resource Record in the TCF

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>VSE</th>
<th>DPPX</th>
<th>RPS</th>
<th>SSP</th>
<th>PDOS</th>
<th>POSC</th>
<th>CMEP</th>
<th>CMFP</th>
<th>NDMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>File update</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Flat data</td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Job</td>
<td>260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Library</td>
<td>260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Member</td>
<td>260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Message</td>
<td>260</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Microcode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Panel</td>
<td>260</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>260</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td>176</td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Print</td>
<td></td>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>PTF</td>
<td></td>
<td>176</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
<tr>
<td>Relational data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Software</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3400</td>
<td>3400</td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2088</td>
</tr>
</tbody>
</table>

Table 58. Space Occupied by Each Resource Record in the TCF

If the transmission plans submitted for execution, or their recursions, do require more space, you can use VSAM utilities to allocate a new TCF with more space and use the VSAM utility to restore the old one.

The TCF also stores two other types of working object:

- Transmission management records, one for each function related to a microcode or microcode customization data resource, or resource group
- Node solicited requests.

Table 58 shows the space required by these objects.

<table>
<thead>
<tr>
<th>Object Type</th>
<th>CMEP</th>
<th>CMFP</th>
<th>NDMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission Management Records</td>
<td>2164</td>
<td>2164</td>
<td></td>
</tr>
<tr>
<td>Node solicited requests</td>
<td>2164</td>
<td>2164</td>
<td></td>
</tr>
</tbody>
</table>

Table 58. Space Occupied by Working Objects in the TCF

Figure 111 on page 455 shows the statements required to define and initialize a new TCF. For details of how to reallocate the contents of an existing TCF using the REPR0 command, refer to the sample JCL shown in Figure 112 on page 455.

Note

This sample is designed to run without the service provider option and in a non-Sysplex (SMSVSAM) environment. If you want to run with the service provider or Sysplex (SMSVSAM) change the highlighted parameter to SHR(2 3) -.
//DEFTCF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=SYSDA,VOL=SER=NDM3/zerodot1,DISP=SHR
//SYSIN DD *
DEFINE CLUSTER (NAME(NVDM.V162.NEWTCF) -
               SHR(4 3) -
               INDEX (NAME(NVDM.V162.NEWTCF.INDEX) -
                       FILE(VOLDD1) -
                       CYL(1) -
                       CIZ(4096) -
               DATA (NAME(NVDM.V162.NEWTCF.DATA) -
                       FILE(VOLDD1) -
                       CYL(3) -
                       RECSZ(500,16232) -
                       CIZ(4096)) -
               CATALOG(NDM3/zerodot1.NDMVSM.UCAT) ;
//TCFINIT EXEC PGM=DSXBIVIN
//OUTD DD DSN=&&TEMP,DISP=(NEW,PASS),
//       UNIT=SYSDA,SPACE=(TRK,1)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
00
0104
//TCFLOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
   REPRO -
       INFILE(INPUT1) -
       OUTFILE(OUTPUT1)
//OUTPUT1 DD DSN=NVDM.V162.NEWTCF,DISP=(OLD)
//INPUT1 DD DSN=&&TEMP,DISP=(OLD,DELETE)
/*

Figure 111. Sample JCL to Define and Initialize a New TCF

Figure 112 provides an example of how to use the REPRO command to reallocate the TCF space, reproducing the contents of the old TCF in a new TCF. Note that this JCL does not contain the TCF initialization step (TCFINIT). The REPRO command shown at the end of this sample defines the two files: INPUT1 is the source file, which is the fragmented TCF. INPUT2 is the target file, which is the new TCF.

//DEFTCF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=338000,VOL=SER=NDM3/zerodot1,DISP=SHR
//SYSIN DD *
DELETE (TARGET.NEWTCF) CLUSTER PURGE;
DEFINE CLUSTER (NAME(TARGET.NEWTCF) -
                SHR(4 3) -
                SPANNED -
                NOWRITECHECK) -

Figure 112 (Part 1 of 2). Sample JCL to Allocate a New TCF
Figure 112 (Part 2 of 2). Sample JCL to Allocate a New TCF

```plaintext
INDEX (NAME(TARGET.NEWTCF.INDEX) -
   UNIQUE -
   FILE(VOLD01) -
   VOL(CPSGES) -
   CYL(1) -
   CISZ(4096) -
   DATA (NAME(TARGET.NEWTCF.DATA) -
   UNIQUE -
   FILE(VOLD01) -
   VOL(CPSGES) -
   KEYS(36,0) -
   CYL(6) -
   RECSZ(500,16232) -
   CISZ(4096)) -
   CATALOG(NDM301.NDMVSM.UCAT) ;

//*
/*TCFREPRO
//TCFLOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
   REPRO -
   INFILE(INPUT1) -
   OUTFILE(OUTPUT1)
//INPUT1 DD DSN=NVDM.V162.NDMTCF,DISP=SHR
//OUTPUT1 DD DSN=TARGET.NEWTCF,DISP=OLD
```
Maintaining the Unsolicited Message File

The unsolicited message file stores messages issued by the TCP and viewed by IOF. It has the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messages</td>
<td>RRRS</td>
<td>NVDM.V162.NDMUN01</td>
<td>F 508</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 59. VSAM Characteristics of the Unsolicited Message File

Unsolicited messages are stored in wraparound mode. The file must have sufficient space allocated to allow messages to be accumulated until the IOF operator can display them. If the file is full, messages may be overwritten. If this happens, the IOF operator is informed by the following message, displayed on the message line:

MESSAGE LOST

Space Requirements

Each message occupies 508 bytes. For systems running with 50 concurrent SNA sessions, space sufficient for 1000 messages is recommended.

Figure 113 shows the JCL to define a new, larger unsolicited message file.

```
//DEFUN01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=SYSDA,DISP=SHR,
//        VOL=SER=NDM3/zerodot1
//SYSIN DD *
DEFINE CLUSTER (NAME(NVDM.V162.NEWUN/zerodot1) -
                NOWRITECHECK -
                FILE(VOLDD1) -
                VOL(NDM301) -
                CYL(1) -
                RECSZ(508,508) -
                CISZ(2048) -
                NUMBERED) -
                CATALOG(NDM301.NDMVSM.UCAT) ;
//*
```

Figure 113. Sample JCL to Define the Unsolicited Message File
Maintaining the NetView DM for MVS Work File

The NetView DM for MVS work file (DSXWF01) contains working records for IOF functions. This file is only needed if the IOF is active. The file has the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>RDDS</td>
<td>NVD.M.V162.NDWF01</td>
<td>F 1024</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

*Table 60. VSAM Characteristics of the NetView DM for MVS Work File*

If the file is full, the IOF operator is informed by the error message:

```
NDM0553E WORK FILE IS FULL
```

If this condition occurs frequently, you can make the NetView DM for MVS work file larger.

**Space Requirements**

To calculate the amount of space needed by the work file, consider the “worst case” condition that could occur, if the IOF operator chooses to display all phases in the TCF. In this case, you should allocate 1K for each phase defined in the TCF, plus 10% for other temporary records needed by IOF components.

Figure 114 shows the JCL to define a new, larger NetView DM for MVS work file:

```
//DEFWF01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=SYSDA,DISP=SHR,
// VOR=SER=NDM301
//SYSIN DD *
DEFINE CLUSTER (NAME(NVDM.V162.NDWF01) -
 NOWRITECHECK -
 FILE(VOLDD1) -
 VOL(NDM301) -
 CYL(1) -
 RECSZ(1024,1024) -
 CISZ(2048) -
 NUMBERED) -
 CATALOG(NDM301.NDVM0.SM UCAT) ;
//*
```

*Figure 114. Sample JCL to Define the NetView DM for MVS Work File*
Maintaining the Exchange Information File (EIF)

The NetView DM for MVS exchange Information file (EIF) is the file where information exchanged between the TCP and GIX is stored. It has the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>ESDS</td>
<td>NVDM.V162.NDMEIF</td>
<td>V80 - 8166</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

**Space Requirements**

To calculate the amount of space needed by the EIF, you must consider the composition of the transmission plans. Each phase that has a node stored in the TCF and whose status is different from COMPLETED, needs 60 bytes.

For example, the record length of a data set is variable from 80 to 8166; for each record of the data set you can store a maximum of 136 phases for a node. Because each track of a 3380 device contains a maximum of 40960 bytes, you can store 5 records for each track. The number of tracks or cylinders depends on the number of nodes in the network and the number of phases for a node that are different from COMPLETED.

Figure 115 shows the JCL to define an exchange information file.

```plaintext
//DEFEIF EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=3380, VOL=SER=NDMAPI, DISP=SHR
//SYSIN DD /*
DELETE (NVDM.V162.NDMEIF) CLUSTER PURGE;
DEFINE CLUSTER (NAME(NVDM.V162.NDMEIF) -
    SHR(4 3) -
    NOWRITCHECK -
    REUSE -
    FILE(VOLDD1) -
    VOL(NDMAPI) -
    CYL(1) -
    RECSZ(80,8166) -
    CISZ(8192) -
    NONINDEXED) -
DATA (NAME(NVDM.V162.NDMEIF.DATA)) -
CATALOG(CATALOG.MVSICF1.VNDMES1) ;

Figure 115. Sample JCL to Define the NetView DM for MVS EIF
```
Maintaining the TCP Work File

The TCP work file (DSXCWK) is used by the transmission control program (TCP) to maintain its working queues. The file has the following characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>KSDS</td>
<td>NVDM.V162.NDMCWK</td>
<td>V 100 to 142</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 61. VSAM Characteristics of the TCP Work File

If the DSXCWK file is full, the user is informed by the error message:

```
NDM0405T A PLAN SCHEDULE REQUEST FOR TRANSMISSION CANNOT BE PERFORMED BECAUSE THE CWK FILE IS FULL.
```

Space Requirements

Each phase by node record defined in a plan submitted to the TCF and not completed (in other words, in a READY, WAITING, HELD, or PENDING state), needs 142 bytes. Once you have calculated the amount, add an extra 30% to allow for:

- IOF commands (for example HOLD/RELEASE PHASE) for DPPX or VSE nodes
- Time events, such as the recursions of recursive plans
- Information related to transmission plans that are completed but not yet purged by the TCFMAINT batch utility.

Figure 116 shows the JCL to define and initialize a new and larger DSXCWK file.

```
//DEFCWK EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//VOLDD1 DD UNIT=SYSDA,VOL=SER=NDM3/zerodot1,DISP=SHR
//SYSIN DD /c5197
DEFINE CLUSTER (NAME(NVDM.V162.NEWCWK) -
    NOWRITECHECK) -
    INDEX (NAME(NVDM.V162.NEWCWK.INDEX) -
        FILE(VOLDD1) -
        VOL(NDM301) -
        CYL(1 1) -
        CISZ(4096) -
    ) -
    DATA (NAME(NVDM.V162.NEWCWK.DATA) -
        FILE(VOLDD1) -
        VOL(NDM301) -
        KEYS(64,0) -
        CYL(1 1) -
        RECSZ(100,180) -
        CISZ(4096) -
    ) -
    CATALOG(NDM301.NDMSM.UCAT) ;
/*
```

Figure 116 (Part 1 of 2). Sample JCL to Define and Initialize the TCP Work File
```c
//
//CWKINIT EXEC PGM=DSXBIVIN
//OUTD DD DSN=&&TEMP,DISP=(NEW,PASS),
//     UNIT=SYSDA,SPACE=(TRK,1)
//SYSIN DD *
W0
0104
//CWKLOAD EXEC PGM=IDCAMS
//SYSIN DD *
    REPRO -
        INFILE(INPUT1) -
        OUTFILE(OUTPUT1)
//OUTPUT1 DD DSN=NVDM.V162.NEWCKW,DISP=(OLD)
//INPUT1 DD DSN=&&TEMP,DISP=(OLD,DELETE)
//
```

*Figure 116 (Part 2 of 2). Sample JCL to Define and Initialize the TCP Work File*
Maintaining the Request Queue File

The request queue file (RQF) provides communication between the NetView DM for MVS components (TCP, GIX and the Batch Utilities). The file has the following VSAM characteristics:

<table>
<thead>
<tr>
<th>Contents</th>
<th>Type</th>
<th>Default DSN</th>
<th>Record length</th>
<th>Cyls</th>
<th>Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>RRDS</td>
<td>NVDM.V162.NDMRQFDA (MVS)</td>
<td>F 500</td>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td>Directory</td>
<td>RRDS</td>
<td>NVDM.V162.NDMRQF</td>
<td>F 2000</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 62. VSAM Characteristics of the Request Queue File

If the request queue file is full, you are informed by an error message like this:

- NDM9192E  OPERATION CANNOT BE PERFORMED BECAUSE REQUEST QUEUE FILE IS FULL
- NDM3485E  PLAN SUBMISSION REJECTED

Space Requirements

Each entry in the request queue file directory uses a file record that is 2000 bytes long. GIX or Batch Utilities add the following entries to the file:

- One entry for each submitted plan
- One entry for each plan deleted from the TCF
- One entry for each phase deleted from the TCF
- One entry for each reset plan
- One entry for each HYSYST destination in the plan.

The maximum number of entries in the request queue file is 7700. The requests are queued either by GIX or by Batch Utilities, and they are dequeued by the TCP. If the TCP is active, there is a dynamic condition in which requests are constantly queued and dequeued. If the TCP is not active, all requests are held in the queue. This could lead to a “file full” condition. If the file becomes full, you can do one of the following:

- Activate the TCP, in order to dequeue requests
- Define a new, larger RQF if the current one has been defined with space allowing less than 7700 entries
- Redefine the directory file.

Figure 117 shows the statements required to define and initialize a new request queue file.

```plaintext
//DEFRQF EXEC PGM=IDCAMS
//SYSPRNT DD SYSOUT=B
//VOLDD1 DD UNIT=SYSDA,DISP=SHR,
//     VOL=SER=NDM001
//SYSIN DD *
```

Figure 117 (Part 1 of 2). Sample JCL to Define and Initialize the RQF
DEFINE CLUSTER (NAME(NVDM.V162.NEWRQF) -
    SHR(4 3) -
    NOWRITEMCHECK -
    FILE(VOLDD1) -
    VOL(NDM001 -
    -
    -
    -
    -
    -
    -
    ) -
    UNIQUE -
    TRK(30) -
    RECSZ(2000,2000) -
    CISZ(2048) -
    NUMBERED) -
    DATA (NAME(NVDM.V162.NEWRQF.DATA)) -
    CATALOG(CATALOG.MVSICF1.VNDM0001) ;
//DEFRQFD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=B
//VOLDD1 DD UNIT=SYSDA,DISP=SHR,
//    VOL=SER=NDM/zerodot/zerodot1
//SYSIN DD *
DEFINE CLUSTER (NAME(NVDM.V162.NEWRQFDA) -
    SHR(4 3) -
    NOWRITEMCHECK -
    FILE(VOLDD1) -
    VOL(NDM001 -
    -
    -
    -
    -
    -
    ) -
    UNIQUE -
    TRK(600) -
    RECSZ(500,500) -
    CISZ(512) -
    NUMBERED) -
    DATA (NAME(NVDM.V162.NEWRQFDA.DATA)) -
    CATALOG(CATALOG.MVSICF1.VNDM0001) ;
/*
//NDMBIRQI EXEC PGM=DSXBIRQI
//NDMRQF DD DSN=NVDM.V162.NEWRQF,DISP=SHR
//NDMRQFDA DD DSN=NVDM.V162.NEWRQFDA,DISP=SHR
//SYSPRINT DD SYSOUT=B
.restart
*/

Figure 117 (Part 2 of 2). Sample JCL to Define and Initialize the RQF
Maintaining the GIX Data Sets

Table 63 on page 465 shows how the GIX data sets are structured. You can maintain the data sets by treating them as physical sequential data sets.

The data sets are the following:

- Skeletons library
- CLISTs library
- Snap data sets
- Commands and validation errors data set
- Log data sets
- Reports data set

Skeletons Library

The NetView DM for MVS skeletons library contains the skeletons used when submitting a NetView DM for MVS batch utility by GIX. The name of this library is specified in the SKELLIB parameter of the NDMGEN macro (described in the Installation and Customization book). The default is NVDM.V162.NDMSKLS.

CLISTs Library

The NetView DM for MVS CLISTs library contains the CLISTs used to invoke GIX from TSO. The name of this library is specified in the CLISTLB parameter of the NDMGEN macro (described in the Installation and Customization book). The default is NVDM.V162.NDMCLST.

GIX Edit Data Sets

This data set is used to edit CLIST and JOB resources using the ISPF/PDF EDIT service. It is automatically created at GIX start-up time. The default name of this data set is userid.NDMGIX.EDIT01, but optionally you can specify two additional qualifiers to be added before the userid value, coding the parameter FQL of the NDMGEN customization macro.

Snap Data Sets

The snap data sets contain the storage dump produced whenever GIX terminates abnormally. The name of these data sets is specified in the DSNAME parameter of the NDMGIX macro (described in the Installation and Customization book). The number of data sets (from 1 to 10) are specified in the DUMPNUM parameter of the NDMGIX macro.

If you need to define new GIX dump data sets and you do not want to rerun the installation step for the macro NDMGIX, you have to allocate them using the program called IEBGENER.

Commands and Validation Errors Data Set

GIX uses this data set to store the command table and validation error messages issued when the validation step of a transmission plan is not successfully completed. When a transmission plan is deleted, NetView DM for MVS automatically maintains this data set by deleting the related member from the data set. However, you can use the Batch Utilities if you need to carry out further maintenance.

The data set is allocated at installation time with the NVDM.V162.NDMTABLE default DSN, and the characteristics shown in Table 63 on page 465.
The amount of space required for the data set is directly proportional to the transmission plan that has validation errors. However, you can change the default values at installation time. The amount of space required is defined in the SPACE parameter of the NDMDATA generation macro.

Log Data Sets

The allocation of the log data sets is required only if you do not accept the installation default for the output of the GIX command log. The default is:

ALLOCATE FILE (DSXLOG) SYSOUT(A)

If you want the logging information on a data set, you must define the data set using ISPF or an MVS utility and replace the statement in the CLIST used to invoke GIX from TSO with:

ALLOCATE FILE (DSXLOG) DATASET ('data set name') MOD

Because the same data set cannot be shared among GIX users, you must define a data set (containing a different DSXLOG data set name) for each GIX user.

Reports Data Set

The data set used to store and print information on resources and plans obtained through GIX is automatically created at GIX startup time. The name of this data set is userid.NDMGIX.PRINT/zerodot1. The data set name must be specified in the GIX CLIST. The GIX program allocates the data set, which has the characteristics shown in Table 63.

The volume serial number and unit, if specified, are the ones coded in the NDMGIX macro (VOL= and UNIT= parameters). Otherwise, available work space is allocated.

<table>
<thead>
<tr>
<th>File</th>
<th>Organization</th>
<th>Record Format</th>
<th>Record Length</th>
<th>Block Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skeletons library</td>
<td>PO</td>
<td>FB</td>
<td>80</td>
<td>3120</td>
</tr>
<tr>
<td>CLISTs library</td>
<td>PO</td>
<td>FB</td>
<td>80</td>
<td>3120</td>
</tr>
<tr>
<td>Snap data sets</td>
<td>PS</td>
<td>VBA</td>
<td>125</td>
<td>882</td>
</tr>
<tr>
<td>Commands and validation errors data set</td>
<td>PO</td>
<td>FB</td>
<td>80</td>
<td>3120</td>
</tr>
<tr>
<td>Log data sets</td>
<td>PS</td>
<td>VBA</td>
<td>125</td>
<td>882</td>
</tr>
<tr>
<td>Reports data set</td>
<td>Sequential</td>
<td>Fixed</td>
<td>133</td>
<td>1330</td>
</tr>
</tbody>
</table>

Table 63. Characteristics of GIX Data Sets
Maintaining the DSXJOBS File

The DSXJOBS file is used to store JCL batch jobs that can be referenced and submitted to the operating system at the end of a transmission plan phase, if the conditions specified in the phase attributes are met.

The specific NDMEXITJ member is also submitted to the operating system at the end of the resource preparation using the NDMPGM program.

This file must have the following characteristics:

**Organization**
PO

**Record Format**
FB

**Record Length**
80

**Block Size**
A multiple of 80.

To calculate the amount of space needed by DSXJOBS, you must consider the size of JCL batch jobs to be stored in the file. The following is a sample job for use in maintaining DSXJOBS:

```jcl
//TORJOB JOB(,GABRIELE,A,),GABRIELE,
// CLASS=I,MSGCLASS=7,MSGLEVEL=1,
// USER=GABRIELE, PASSWORD=STEFANIN
//JOBCAT DD DSN=CATALOG.MVSICF1.VNDM002,DISP=SHR
//*/
//*/
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=* 
//SYSIN DD *

DELETE NVDM.V162.DSXJOBS
//*/
//*/
//STEP2 EXEC PGM=IEFBR14
//NDMJOBS DD DSN=NVDM.V162.DSXJOBS,DISP=(NEW,CATLG),
// UNIT=3380,VOL=SER=NDM002,SPACE=(TRK,(1,1,6)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

*Figure 118. Sample JCL to Maintain the DSXJOBS File*
Part 3. Using the Interactive Operator Facility (IOF)

This part describes how to control the transmission control program (TCP) using the interactive operator facility (IOF). You can also control the TCP using MVS MODIFY commands direct from the system console. For a complete list of the MVS commands you can use, refer to Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549.

Note: Not all of the attached nodes support the parameters that NetView DM for MVS can generate. For specific limitations, refer to the appropriate node documentation.

Chapter 19. Starting the Transmission Control Program (TCP) 471

What Is the TCP? 471
Multiple TCPs 471
Operator Control 471

Preparing Job Control Statements 472

Specifying TCP Startup Parameters 475
Specifying the TCP Profile (PROFILE=xx) 476
Specifying Operator Control (OPCTL=) 476
SPMF Instance Number (INST=) 477
Controlling Phase Startup (AUTOSTR=) 477
Controlling TCP Termination (AUTOEND=) 478
Specifying Warm/Cold Startup (START=) 478
Suppressing Unsolicited Messages (MSGINFO=) 479
Specifying the Maximum Number of SNA Sessions (MAXTASK=) 480
Specifying the Current Number of SNA Sessions (CURTASK=) 480
Specifying Which Transmission Profiles to Use (RESTTP=) 481
Specifying the Effect of Network Changes (NETCHNG=) 481
Setting Up the D&CC API Environment (NDCCAPI=) 482
Example of Startup Parameters 483

Submitting the JCL 483
Handling Inconsistencies in the Network During TCP Cold Start 483

Chapter 20. Logging On to IOF and Starting Transmission 485

Logging on to IOF 485
Before You Log On 485
Logging on from VTAM or TCAM 486
Logging on from NCCF/TAF 486
What Could Go Wrong 486

IOF Logon Panel 486
What Could Go Wrong 487
The Transmission Control Master Menu 488
Selecting from the Master Menu 488
What Could Go Wrong 489

IOF Panel Layout 489
Using IOF with a Katakana Terminal 492
Getting Help 492

IOF Operator Functions and Panel Flow 493
Option 1 – Start Transmission 494
Using an MVS MODIFY Command to Start Transmission 494
What Could Go Wrong 494
Logging off from the IOF 494
### Chapter 21. Controlling Transmission Schedules

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Schedule Display and Control</td>
</tr>
<tr>
<td>3</td>
<td>Hold Transmission for a Node/Group</td>
</tr>
<tr>
<td>4</td>
<td>Release Transmission for a Node/Group</td>
</tr>
</tbody>
</table>

#### Selecting the Type of Transmission Request

- Specific Phases, NSRs, or D&CC API Requests
- Transmission Requests Within a Date or Time Range
- Transmission Requests With a Specific Status

#### Examples

- What Could Go Wrong

#### Displaying Selected Transmission Requests

- Information Shown for Each Transmission Request

#### Changing the Transmission Schedule

- Using Line Commands
  - C – Change the Start Time and Date
  - D – Delete a Transmission Request
  - H – Hold Transmission of a Phase or D&CC API Request
  - R – Release a Held Phase or D&CC API Request
  - S – Restart a Phase
  - L – List Status of All Functions of a Transmission Request
  - I – Display a List of Conditioning Phases for Node
  - E – Display a List of Conditioned Phases for Nodes

### Chapter 22. Controlling Transmission with the Nodes

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Hold Transmission for a Node/Group</td>
</tr>
<tr>
<td>4</td>
<td>Release Transmission for a Node/Group</td>
</tr>
</tbody>
</table>

#### Holding a Single Node

- Immediately

#### Holding a Group of Nodes

- Immediately

#### What Could Go Wrong

- What To Do when Transmission is Held by the System

### Chapter 23. Changing Transmission Profiles

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Display/Change CURTASK or Transmission Profiles</td>
</tr>
</tbody>
</table>

#### Changing the CURTASK Values

#### Changing Transmission Profiles

### Chapter 24. Stopping Transmission and the Transmission Environment

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Quiesce Transmission</td>
</tr>
<tr>
<td>6</td>
<td>Immediate Quiesce Transmission</td>
</tr>
<tr>
<td>7</td>
<td>End Transmission</td>
</tr>
<tr>
<td>8</td>
<td>Immediate End Transmission</td>
</tr>
<tr>
<td>9</td>
<td>Disconnect IOF</td>
</tr>
<tr>
<td>10</td>
<td>Log Off from IOF</td>
</tr>
</tbody>
</table>

### Chapter 25. Handling Unsolicited Messages

<table>
<thead>
<tr>
<th>Types of Messages</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handling Unsolicited Messages</td>
<td>533</td>
</tr>
</tbody>
</table>
Chapter 19. Starting the Transmission Control Program (TCP)

This chapter explains how to initialize the transmission control program (TCP), before you can begin to use the interactive operator facility (IOF).

What Is the TCP?

The transmission control program (TCP) controls the execution of several different types of transmission request:

- Requests defined in transmission plans. A transmission plan is a set of requests that a GIX or batch utility user defines and submits to the transmission control file (TCF) for processing by the TCP.
- Single transmission requests that the host receives from nodes in an unsolicited way. These are called node solicited requests (NSRs).
- Requests issued by user programs through the Distribution and Change Control Application Programming Interface (D&CC API).

The term “transmission requests” is used throughout this part to refer to all three of the above units of work.

The TCP is a batch job which runs under control of the operating system. Before you can log on to IOF, you must activate the TCP. You activate the TCP in the same way as any other batch job, by preparing job control statements and submitting them to the operating system for processing, whenever you want to activate the transmission environment.

**Note:** Before submitting the batch job to activate the TCP, you must have defined at least one node to the distributed resource directory (DRD).

Multiple TCPs

A large network may be partitioned functionally, geographically, by system node type, or by other criteria, creating individual sub-networks. Each sub-network has its own TCP, and each TCP is controlled by an operator. So, you can run multiple TCPs on the same MVS host system at the same time.

For more information on setting up and installing multiple TCPs, refer to the Installation and Customization book.

Operator Control

You can run the TCP, or multiple TCPs, in one of two modes:

**Attended mode**  An operator controls the TCP using either the IOF interface to the TCP, or MVS MODIFY commands direct from the system console. The operator can issue commands from the TCP to control transmission activity and respond to messages generated by the TCP.

**Unattended mode**  The TCP runs without direct operator control.
Preparing Job Control Statements

You specify the mode you want to run in using job control language (JCL) when you define startup parameters.

The following pages show you examples of the job control statements needed to run a single TCP or multiple TCPS, and then describe the parameters that you can define at startup time.

Preparing Job Control Statements

Figure 119 shows an example of JCL that you can use to run a single TCP in the SPMF environment:

```
1 //STEP1 EXEC PGM=DSXTMM00,
2 //PARM='PROFILE=02,START=WARM,OPCTL=IOF,INST=n'
3 //SYSPRINT DD SYSOUT=A
4 //SNAP DD SYSOUT=A
5 //SYSUDUMP DD SYSOUT=A
6 //DSXLIB DD DSN=NDM16.NDMLIB,DISP=SHR
7 //DSXLIBT DD DSN=NDM16.NDMLIB,DISP=SHR
8 //NDMEIF DD DSN=NDM16.NDMEIF,DISP=SHR
9 //DSXHFDA DD DSN=NDM16.NDMHFDA,DISP=SHR
10 //NDMRQFDA DD DSN=NDM16.NDMRQFDA,DISP=SHR
11 //NDMRQF DD DSN=NDM16.NDMRQF,DISP=SHR
12 //DSXWF DD DSN=NDM16.NDMWF,DISP=SHR
13 //DSXUN DD DSN=NDM16.NDMUN,DISP=SHR
14 //DSXCWK DD DSN=NDM16.NDMCWK,DISP=SHR
15 //DSXJOBS DD DSN=NDM16.NDMJOBS,DISP=SHR
16 //DSXXSUB DD SYSOUT=(A,INTRDR)

* The following statements are needed for TCAM only *

17 //DSX DD QNAME=qname
18 //DSXI0F DD QNAME=qname
```

Figure 119. JCL to Run a Single TCP in SPMF Environment

Figure 120 on page 473 shows an example of JCL that you can use to run a single TCP in a non-SPMF environment:
Preparing Job Control Statements

//STEP1 EXEC PGM=DSXTMM00,
//PARM='PROFILE=02,START=WARM,OPCTL=IOF,MAXTASK=(200,100),CURTASK=(100,60)'
//SYSPRINT DD SYSPRT=A
//SNAP DD SYSPRT=A
//SYSDUMP DD SYSPRT=A
//DSXRDR DD DSN=NDM16.NDMDRD,DISP=SHR
//DSXLIB DD DSN=NDM16.NDMLIB,DISP=SHR
//DSXLIBT DD DSN=NDM16.NDMLIBT,DISP=SHR
//NDMEIF DD DSN=NDM16.NDMEIF,DISP=SHR
//DSXHFDI DD DSN=NDM16.NDMHFDI,DISP=SHR
//DSXHFDAA DD DSN=NDM16.NDMHFDAA,DISP=SHR
//NDMRDFDA DD DSN=NDM16.NDMRDFDA,DISP=SHR
//NDMRQF DD DSN=NDM16.NDMRQF,DISP=SHR
//DSXJOBS DD DSN=NDM16.TEST.JCL,DISP=SHR
//DSXUN/zerodot1 DD DISP=OLD,DSN=NDM16.NDMUN/zerodot1B
//DSXWF/zerodot1 DD DISP=OLD,DSN=NDM16.NDMWF/zerodot1B
//DSXDDSUB DD SYSOUT=(A,INTRDR)

* The following statements are needed for TCAM only *

//DSX DD QNAME=qname
//DSXIOF DD QNAME=qname

Figure 120. JCL to Run a Single TCP

Figure 121 shows an example of JCL that you can use to run a second TCP (TCP-B) in a non-SPMF environment:

//*************************************************************
/** START OF NDM TCP-B (VTAM MUST BE ACTIVE) */
//*************************************************************

//STEP1 EXEC PGM=DSXTMM00,
//PARM='OPCTL=IOF,AUTOSTR=N,AUTOEND=N,PROFILE=02'
//STEPLIB DD DSN=NDM16.NDMLOAD,DISP=SHR
//DSXPRINT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SNAP DD SYSOUT=A
//DSXRDR DD DSN=NDM16.NDMDRD,DISP=SHR
//DSXLIB DD DSN=NDM16.NDMLIB,DISP=SHR
//DSXLIBT DD DSN=NDM16.NDMLIBT,DISP=SHR
//NDMEIF DD DSN=NDM16.NDMEIF,DISP=SHR
//DSXHFDI DD DSN=NDM16.NDMHFDI,DISP=SHR
//DSXHFDAA DD DSN=NDM16.NDMHFDAA,DISP=SHR
//NDMRDFDA DD DSN=NDM16.NDMRDFDA,DISP=SHR
//NDMRQF DD DSN=NDM16.NDMRQF,DISP=SHR
//DSXJOBS DD DSN=NDM16.TEST.JCL,DISP=SHR
//DSXUN/zerodot1 DD DISP=OLD,DSN=NDM16.NDMUN/zerodot1B
//DSXWF/zerodot1 DD DISP=OLD,DSN=NDM16.NDMWF/zerodot1B
//DSXDDSUB DD SYSOUT=(A,INTRDR)

Figure 121. JCL to Run a Second TCP (Multiple TCPs)

Note: The numbers 1 through 5 in the figures are used for illustrative purposes only. They are not part of the JCL files.
Preparing Job Control Statements

The JCLs shown in Figure 119 on page 472, Figure 120 and Figure 121 contain the following sets of statements:

1. An EXEC statement, which invokes the TCP with the DSXTMM00 program. This program processes the passed parameters (TCP Profile and TCP Invocation).

2. The PARM= statement, which you use to:
   - Specify the transmission environment for this run of the TCP, using the PROFILE=xx parameter.
   - Specify the instance number to refer to the correct set of DB2 tables, using INST=n parameter. This applies only in the SPMF environment.
   - Change the startup parameters defined at installation time for the TCP profile specified.

The parameters you define here override those specified at installation time in the NDMTCP generation macro, as described in the *Installation and Customization* book. “Specifying TCP Startup Parameters” on page 475 describes the startup parameters you can define.

3. JCL data definition (DD) statements that define the following NetView DM files:
   - **DSXDRD**: Distributed resource directory (DRD)
   - **DSXLIB**: NetView DM for MVS library directory
   - **DSXLIBT**: NetView DM for MVS library data
   - **DSXHFDI**: Holding file directory
   - **DSXHFDA**: Holding file data
   - **DSXTCF**: Transmission control file (TCF)
   - **NDMRQF**: Request queue file directory
   - **NDMRQFDA**: Request queue file data
   - **NDMEIF**: NetView DM for MVS exchange information file

   Note: In the SPMF environment, the DRD and TCF VSAM clusters are no longer referenced by NetView DM for MVS programs. For this reason, the following JCL DD statements must be removed as shown in Figure 119 on page 472:
   ```
   //DSXDRD
   //DSXTCF
   ```

   Note: The distributed resource directory (DRD) must contain at least one defined node before you can start the TCP, otherwise you will receive an error message.

4. JCL DD statements which refer to files used by the TCP only:
   - **DSXWF01**: Work file for IOF displays
   - **DSXUN01**: Unsolicited message file for IOF
   - **DSXCWK**: TCP work file.
   - **DSXJOBS**: Partitioned data sets containing batch procedures to be submitted at the end of a phase. If you use the DSXJOBS DD statement, you must code the DSXDDSUB DD statement as follows:
   ```
   //DSXDDSUB DD SYSOUT=(A,INTRDR)
   ```

474 NetView DM for MVS User's Guide
If you do not, the batch jobs stored in the DSXJOBS file are not submitted, even if the conditions specified in the PROC operand of the DEFINE PHASE statement are met.

**DSXDDSUB**  MVS internal reader.

**Note:** If you are not using IOF (in other words, you choose to run in unattended mode), you do not require the DSXWF01 and DSXUN01 files.

If you are running multiple TCPs (without the SPMF installed), the TCPs can share the following files:
- Holding file (DSXHFDI and DSXHFDA)
- NetView DM for MVS library (DSXLIB and DSXLIBT)
- Distributed resource directory (DSXDRD).

Multiple TCPs (without SPMF installed) cannot share the following NetView DM for MVS files:
- Transmission control file (DSXTCF)
- Request queue file directory (NDMRQF)
- Request queue file data (NDMRQFDA)
- NetView DM for MVS exchange information file (NDMEIF)
- Unsolicited message file for IOF (DSXUN01)
- TCP work file (DSXCWK).

The last two statements in Figure 120 on page 473 identify the NetView DM for MVS TCP (DSX) and the NetView DM for MVS control operator (DSXIOF), if you are running in attended mode. Each statement should specify QNAME=qname, where qname is the name of a TPROCESS statement specifying RAPI=YES in the TCAM (telecommunications access method) Message Control Program (MCP). These names must match the application names in the APPLID and IAPPLID operands of the TCP profiles, defined in the NDMTCP generation macro. Refer to the Installation and Customization book for information.

---

**Specifying TCP Startup Parameters**

You use the PARM= statement to:
- Specify the transmission environment you want to use for this run of the TCP, using the PROFILE=xx parameter.
- Change the startup parameters defined at installation time for the TCP profile. You can decide:
  - Whether you want to run NetView DM for MVS transmission operations with operator control (OPCTL=IOF or OPCTL=NETV), or without operator control (OPCTL=NONE).
  - The instance number to reference the correct set of DB2 tables. This applies only to the SPMF environment.
  - Whether you want the transmission activities to start with intervention from the IOF operator (AUTOSTR=NO), or without intervention (AUTOSTR=YES).
  - Whether you want the transmission to end with intervention from the IOF operator (AUTOEND=NO), or without intervention from the IOF operator (AUTOEND=YES).
Specifying TCP Startup Parameters

- Whether you want the TCP initialization to be a cold start (START=COLD), or a warm start (START=WARM). The default is COLD.
- Whether you want to suppress unsolicited information and warning messages (MSGINFO=).
- The maximum number of SNA sessions that can run concurrently (MAXTASK=).
- The current number of SNA sessions that can run concurrently (CURTASK=).
- Which set of transmission profiles you want to use when warm starting the TCP (RESSTP=).
- Whether you want to TCP to reflect network configuration changes immediately in transmission activities (NETCHNG=IMMEDIATE), or only when the TCP is warm started with NETCHNG=IMMEDIATE or cold started (NETCHNG=DELAYED).
- Whether you want to use the D&CC API to receive transmission requests coming from user programs (NDCCAPI=).

Once you have specified these parameters at TCP startup time, the only parameter that you can change during the current invocation of the TCP is the CURTASK parameter. The other parameters remain effective for this invocation of the TCP.

You can change the CURTASK parameter while the TCP is running, to define the number of concurrent SNA sessions that are currently running. The maximum value you can assign to the CURTASK value is the value defined for the MAXTASK parameter. The minimum value is 1. If you do not specify a value for CURTASK when the TCP is started, the default is the value defined for MAXTASK.

Use either the IOF menu option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES, described in Chapter 23, “Changing Transmission Profiles” on page 521, or a MODIFY (F) command from the system console.

Specifying the TCP Profile (PROFILE=xx)

The TCP profile defines the transmission environment for this run of the TCP. The PROFILE=xx parameter of the PARM='.......' statement defines this environment. xx represents the suffix parameter of the TCP profile. Specify two numeric digits. The default value is 01.

Specifying Operator Control (OPCTL=)

This parameter specifies whether you want an operator to control the TCP, in attended mode, or whether you want the TCP to run without operator control, in unattended mode.

If you want to run in attended mode, specify one of the following:

OPCTL=IOF

Allows you to monitor the TCP from the IOF and issue commands to control the TCP. You can also view and respond to messages issued during transmission.
OPCTL=NETV
This is the same as OPCTL=IOF above, but with the additional capability of being able to send selected messages to the NetView console.

When you specify OPCTL=NETV, the TCP sends messages to NetView DM for MVS using the Write To Operator (WTO) macro. For more information on the WTO macro and its NDMTCP operands, refer to the Installation and Customization book.

Note: If your TCP has been customized with NETVML=YES in the NDMTCP customization macro, authorizing use of the multiline WTO macro, the TCP must run APF-authorized.

The rest of this part shows what you, the operator, can do to monitor and control transmission operations in attended mode, using the IOF.

If you want to use MVS MODIFY commands from the system console to control transmission operations, you can find a complete list of available MVS commands in Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549.

Sometimes you may want to run the TCP in unattended mode. To do this, specify:

OPCTL=NONE

SPMF Instance Number (INST=)
If you are running in the SPMF environment, you specify the instance number that you set at installation time to reference the correct DB2 tables.

Controlling Phase Startup (AUTOSTR=)
If you are running in attended mode, you can specify whether you want to start transmission automatically, or with an IOF command entered by the operator.

The parameters you can define are:

AUTOSTR=YES|Y
The TCP starts transmission activities automatically.

AUTOSTR=NO|N
The TCP does not start transmission operations automatically. You must log on to IOF and use the IOF START TRANSMISSION command to begin transmission. This command is described in Chapter 20, “Logging On to IOF and Starting Transmission” on page 485.

When running in unattended mode, you must specify that transmission activities start automatically.
Controlling TCP Termination (AUTOEND=)

If you are running in attended mode, you can specify whether you want to end transmission:

- Automatically
- With an IOF command entered by the operator

If you are running in unattended mode, you can specify whether transmission should end automatically.

The parameters you can define are:

**AUTOEND=YES|Y**

The TCP will automatically end when there are no more phases whose start time falls within the maximum waiting time. Whoever is installing NetView DM for MVS defines the maximum waiting time in the MAXWAIT parameter at installation time.

If you specify AUTOEND=YES and are working with the following types of node:

- VSE nodes
- DPPX, SSP, or VM nodes transmitting compressed data
- PDOS or SSP end nodes connected to the host via intermediate node support
- POSC nodes executing asynchronous CLISTs
- User-defined type nodes with NetView DM for MVS Transfer (NDMT) functional capabilities
- User-defined type nodes with Change Management Entry Point (CMEP) and Change Management Focal Point (CMFP) functional capabilities

The TCP ends when it receives all of the expected delayed acknowledgements.

**AUTOEND=NO|N**

The TCP does not end transmission operations automatically. If you are running in attended mode, you can either use an IOF command to end transmission, or you can enter an MVS MODIFY (F) command from the system console. Refer to the *MVS System Commands* book for information about system console commands.

Specifying Warm/Cold Startup (START=)

Before the TCP starts to execute transmission plans, it erases the old work file (DSXCWK), reads the TCF, and rebuilds the work file.
You can specify in the JCL whether you want to invoke a warm or cold startup of the TCP:

**START=COLD|C**

This is the default value. The first time that you run the TCP you can specify either **START=COLD** or **START=WARM**, providing that there is at least one node defined in the NetView DM for MVS DRD. This can be either after the installation of NetView DM for MVS, or when you reinitialize the TCF or the work file. After a cold start NetView DM for MVS rebuilds the work file.

You must specify a cold start if **NETCHNG=IMMEDIATE** (see “Specifying the Effect of Network Changes (NETCHNG=)” on page 481) when the DRD is shared between multiple TCPs and node definitions have been changed or added to the DRD.

**START=WARM|W**

Specify a warm start to start the TCP quickly. A warm start adds to or deletes from the work file only those modifications made to the TCF while the TCP was not active.

Do not specify a warm start if a previous TCP run terminated abnormally (for example, with an abend or host system failure). In this case, the TCP does not start and NetView DM for MVS issues a message.

**Note:** In the case of phases with transmission windows defined, the status of the phase is rebuilt from the work file, as it was checkpointed when the TCP shut down.

This means that after a warm start of the TCP, phases with transmission windows will have the same status, either HWIN or READY, as they had when the TCP shut down. The phase remains in this status even if a new window was reached, or even expired, between the time when the TCP shut down and the warm startup. Windows are not updated until you issue a request to start transmission (see “Option 1 – Start Transmission” on page 494).

**Suppressing Unsolicited Messages (MSGINFO=)**

The TCP generates unsolicited messages containing information or warnings about transmission activities.

If you want messages to be displayed at the NetView console, perform the following steps:

- Define **OPCTL=NETV**.
- Insert the message in the NDMNETTB table. Refer to the *Base Application Programming* book for information.

You do not need to specify a value for the **MSGINFO** parameter.

You can specify whether you want to get unsolicited messages at all, and where you want to get them. Specify one of the following:

**MSGINFO=0**

The TCP does not issue unsolicited information or warning messages.

**MSGINFO=1**

The TCP routes information and warning messages to SYSPRINT in both attended and unattended mode.
Specifying TCP Startup Parameters

**MSGINFO=2**

The TCP routes unsolicited information and warning messages to SYSPRINT, and also to the unsolicited message file if you are running in attended mode (OPTCTL=IOF or OPTCTL=NETV).

**Specifying the Maximum Number of SNA Sessions (MAXTASK=)**

This parameter specifies the maximum number of SNA sessions that can be initiated by the TCP at the same time.

**MAXTASK=(number1,number2)**

You must specify two values:

- The total number of sessions for both leased and switched lines (number 1)
- The maximum number of sessions using switched lines only (number 2).

If there are more switched ports available than the value you specify, the TCP will use them and initiate more sessions, if the nodes request more sessions.

The default values are (2,1), and the minimum allowed values are (1,0). The second value must be less than the first value.

The maximum allowed for the first value is the maximum number of subtasks allowed by the operating system, minus those used internally by NetView DM for MVS. Subtasks used by NetView DM for MVS are:

- One for the request queue (RQ) processor (always present)
- One for IOF (only used if OPCTL=IOF or OPCTL=NETV)
- One to manage requests from nodes (only if DDPREQ=NETV in the TCP generation macro)
- One for TCP internal use (a phase initiator/terminator)
- One for the D&CC API request processor (if you are using the D&CC API).

Each subtask requires approximately 12K of storage, in addition to the VSAM buffer requirements.

The number of sessions you specify for the CURTASK parameter affects only the session that can be initiated by the host. This means that the number of NSRs that can be initiated by TCP at the same time depends on the value you specify for MAXTASK only. To initiate an high number of NSRs you must specify a low value for CURTASK and a high value for MAXTASK.

**Specifying the Current Number of SNA Sessions (CURTASK=)**

This parameter specifies the current number of SNA sessions that can be initiated by the TCP at the same time.

**CURTASK=(number1,number2)**

You must specify two values:

- The current number of sessions for both leased and switched lines (number 1)
- The current number of sessions using switched lines only (number 2).

The maximum allowed values are the values specified for MAXTASK. The values specified for MAXTASK are also the default values.
You can change the values you enter for the CURTASK parameter while the TCP is running, using either the IOF or an MVS MODIFY command. Figure 120 on page 473 shows an example of a JCL statement that includes the CURTASK startup parameter. Refer to Chapter 23, “Changing Transmission Profiles” on page 521 for information about how to change this value.

Use the following MVS MODIFY command from the system console to change the value of the CURTASK parameter:

```
MODIFY | F jobname,CURTASK=(number1,number2)
```

jobname is the name of the job in which you want to modify the CURTASK value.

The number of sessions you specify for the CURTASK parameter affects only the session that can be initiated by the host. This means that the number of NSRs that can be initiated by TCP at the same time depends on the value you specify for MAXTASK only. To initiate an high number of NSRs you must specify a low value for CURTASK and a high value for MAXTASK.

### Specifying Which Transmission Profiles to Use (RESTTP=)

This parameter is only relevant when you perform a warm start of the TCP. It is ignored during a cold start. The parameter tells the TCP whether you want to use the transmission profiles that you modified when you last used IOF, or go back to using the transmission profiles as they were before modifying them. Transmission profiles are described in Chapter 23, “Changing Transmission Profiles” on page 521.

You can specify one of the following:

- **RESTTP=YES**
  
  This tells the TCP to use the set of transmission profiles that NetView DM for MVS saved after the last TCP cold start. This means that you want to use the transmission profiles as they were before being modified.

- **RESTTP=NO**
  
  This tells the TCP to use the set of transmission profiles saved during the last TCP orderly closedown. An orderly closedown is a normal shutdown of the TCP (completion code=0), rather than an abend or other abnormal condition. This means that you want to use the changed transmission profiles.

If you do not specify a value here, the default is RESTTP=YES.

### Specifying the Effect of Network Changes (NETCHNG=)

You can use this parameter to override the value defined at installation time by the NETCHNG operand of the NDMTCP generation macro.

You can specify one of the following:

- **NETCHNG=IMMEDIATE**
  
  This tells the TCP to reflect network configuration changes in transmission activities immediately.

  Network configuration changes that can come into effect immediately are:

  - The definition of a new node
Specifying TCP Startup Parameters

- The deletion of an existing node definition
- The update of an existing node definition that affects at least one of the following attributes:
  - Logical unit
  - Logon mode
  - RGN (LU 6.2 only)
  - REN (LU 6.2 only)
  - Connection type (LU 0 only)
  - Directory name (LU 0 only)
  - Node class
- The assignment of a logical unit to a transmission profile
- The assignment of a logical unit to a connection profile.

If you want the TCP to be notified of network configuration changes, you also have to run GIX, SUBMIT batch utility or your application program accessing the D&CC API with a profile that specifies NETCHNG=IMMEDIATE in the NDMGIX, NDMBATCH and NDMAPPL customization macros.

**NETCHNG=DELAYED**
This tells the TCP not to reflect network configuration changes in transmission activities until either the next warm start with NETCHNG=IMMEDIATE, or the next cold start.

**Setting Up the D&CC API Environment (NDCCAPI=)**
You can use this parameter to override the value defined at installation time by the NDCCAPI operand of the NDMTCP generation macro. This parameter defines whether you want to use the Distribution and Change Control Application Programming Interface (D&CC API). You can then process D&CC API requests coming from user programs.

The parameters you can define are:

**NDCCAPI=YES**
NetView DM for MVS sets up the D&CC API request processor environment which allows you to process D&CC API requests coming from user programs. Specifying this as a TCP startup parameter overrides the NDCCAPI operand of the NDMTCP generation macro, which describes in the *Installation and Customization* book.

**NDCCAPI=NO**
Specify this parameter if you do not want to set up the D&CC API request processor environment. This overrides the values defined at installation time.
Example of Startup Parameters

The following example describes how to specify a particular set of startup parameters. Suppose that you want to:

- Warm start the TCP
- Control and start transmission operations using the IOF
- Route unsolicited information and warning messages to both SYSPRINT and the unsolicited message file
- Use the set of transmission profiles that NetView DM for MVS saved after the last cold start.

You would use this JCL statement:

```
//NDMTCP EXEC PGM=DSXTMM00,
   //   PARM=('START=WARM,OPCTL=IOF,AUTOSTART=NO',
   //   'MAXTASK=(50,10),CURTASK=(20,5),MSGINFO=2,RESTTP=YES')
```

When you specify the startup parameters, remember to follow the JCL rules for continuing statements.

The parameters that you do not specify will take the values specified at installation. If no values were specified at installation time, the TCP uses the NetView DM for MVS default values.

Submitting the JCL

Once you have prepared the job control statements, you submit them to the operating system for processing to activate the TCP.

**Note:** When you activate the TCP, VSAM may issue the message IEC070I, RC=104 and RC=203, during the initialization of the unsolicited message file. You can ignore this message.

When you have initialized the TCP, you can log on to IOF and begin transmission. The next chapter describes how to do this.

If you decide that you want to stop the TCP now, turn to Chapter 24, “Stopping Transmission and the Transmission Environment” on page 529.

Handling Inconsistencies in the Network During TCP Cold Start

NetView DM for MVS performs some checks during the cold start of the TCP, to verify the consistency of the definitions of nodes belonging to the network. If NetView DM for MVS detects an inconsistency, the TCP indicates these inconsistencies and excludes the node definition containing the errors from the “in-storage image” of the network. For each node that is in error, the TCP writes a message to SYSPRINT and to the system console.

At the end of the cold start processing, a Write To Operator with Reply (WTOR) is sent to the system console and the system prompts you to enter either one of the following commands:

- A GO command, if the TCP can continue without the nodes that have been discarded
- An END command, if the TCP has to be ended.
Handling Inconsistencies

You must enter the reply to the WTOR in the format /nn,GO, or /nn,END, where nn is the number identifying the WTOR on the system console. If the reply is GO, submitted plans that involve the discarded nodes are not taken into account until the next TCP cold start. You can recover discarded nodes without ending the TCP if the TCP is running with the NETCHNG installation or startup parameter set to IMMEDIATE, and the Batch, GIX, or D&CC API application program profile (depending on which of these you are using) is also running with NETCHNG=IMMEDIATE.

To recover the inconsistent nodes, you must change the node definition. If you are using Batch Utilities, use the CHANGE NODE statement. If you are using GIX, use option 1 on the Configure Network panel. If you are using D&CC API, use the DCCCHGND verb.

If changing the node definition does not correct the original inconsistencies (and you do not introduce any new inconsistencies), NetView DM for MVS accepts the change to the network but issues a message to inform you that the node is still inconsistent.

Two or more nodes cannot have the same RGN and REN values within the network. If two nodes are inconsistent because they have the same RGN and REN values, and you resolve this inconsistency by changing the definition of one of the nodes while the TCP is running, the other node definition is automatically changed.

NetView DM for MVS checks that the following conditions have been met when checking the consistency of the network definition:

- Nodes that share the same LU cannot have different protocol types
- LU 0 nodes that share the same LU can have different node types only if one of the following conditions are met:
  - They are defined with a connection type of Intermediate
  - The node type is PDOS
- Two or more nodes that share the same LU cannot have RGN and REN equal to blank, because NetView DM for MVS replaces the blanks with the NETWORK_ID and the partner LU NAME.
Chapter 20. Logging On to IOF and Starting Transmission

This chapter explains how to log on to the interactive operator facility (IOF), and provides you with the information you need to begin using the IOF. This includes:

- The transmission control master menu
- The layout of IOF panels
- The flow of panels you use to perform specific tasks
- The procedure for logging off from the IOF

Finally, this chapter shows you how to start transmission activities using IOF. You can also start transmission using an MVS MODIFY command direct from the system console.

Logging on to IOF

You can log on to IOF from any terminal supported by NetView DM for MVS. You can log on from the IBM telecommunications access method (TCAM), the IBM virtual telecommunications access method (VTAM), or if you are using the IBM Network Communication Control Facility (NCCF) from the Terminal Access Facility (TAF). Only one operator at a time can log on to IOF, but you can run more than one TCP at the same time. There are three cases in which you must log on to IOF:

- You activated the TCP with the parameters OPCTL=IOF or OPCTL=NETV, to request operator control, and AUTOSTR=NO (MVS) which means that you have to start transmission using IOF.
- You disconnected IOF, but now you want to resume control.
- You logged off from IOF. The TCP does not end when you log off from IOF, so you can log on again either to resume control, or to log off from the TCP.

Before You Log On

Before you can log on to IOF, you must ensure that you have met these prerequisites:

<table>
<thead>
<tr>
<th>Prerequisites</th>
<th>How and Where Specified</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPCTL=IOF or</td>
<td>In the NDMTCP generation macro or in the TCP startup parameters. See “Specifying Operator Control (OPCTL=)” on page 476.</td>
</tr>
<tr>
<td>OPCTL=NETV</td>
<td></td>
</tr>
<tr>
<td>IOF=Authorization to use IOF</td>
<td>In your NetView DM for MVS user profile. This is your authorization to use IOF.</td>
</tr>
<tr>
<td>TCP must have been initialized</td>
<td>With the JCL as shown in Figure 120 on page 473.</td>
</tr>
</tbody>
</table>
Logging on from VTAM or TCAM
The way you enter the logon command depends on the VTAM or TCAM logon requirements implemented at your installation. In some systems, NetView DM for MVS can be set up as an option that you select from an application menu. If you have to enter the command yourself, however, it looks like this:

\[ \text{L[OGON]} \text{ APPLID}(\text{iapplid})(\text{PASSWORD}) \]

where \text{iapplid} is the name with which IOF was defined in the NDMTCP generation macro, and to VTAM or TCAM. Once VTAM or TCAM recognizes the \text{iapplid} and password you enter, the IOF Logon panel appears.

Logging on from NCCF/TAF
When you are in session with the IBM Network Communication Control Facility (NCCF), you can log on to IOF through the Terminal Access Facility (TAF). Use the following command:

\[ \text{BGNSESS FLSCN, APPLID}=\text{iapplid},\text{optional parameters} \]

where \text{iapplid} is the one defined in the NDMTCP generation macro. FLSCN specifies that you want to work in full screen mode. For information about the BGNSESS command, refer to the appropriate NCCF/TAF manual.

Once VTAM recognizes the \text{iapplid} you entered from NCCF/TAF, the IOF Logon panel appears.

What Could Go Wrong
If you enter the logon command before initializing the TCP, you will receive the message:

\[ \text{APPLID PARAMETER INVALID} \]

Wait until the TCP is initialized, and then reenter the logon command.

IOF Logon Panel

![Figure 122. The MVS IOF LOGON Panel](image-url)
To log on to IOF, follow these instructions:

1. Enter the Operator ID defined by your system administrator in your user profile in the **OPERATOR ID** field.

2. Enter your Password, if the system administrator assigned one to you in your user profile, in the **PASSWORD** field. The password does not appear on the screen when you enter it.

   If RACF control is activated for IOF, the same operator id and password must be supplied as that used to logon to TSO; the password stored in the DRD will be ignored (the IOF option must be On in the user authorisations of the DRD).

3. Enter the name of the printer you want to use in the **HARD COPY** field, if it is different to the default name currently displayed. If you do not change the value, IOF assumes that this is the printer you want to use in the current IOF session.

   You can enter from 1–8 alphameric or special characters. The first character must be alphabetic or special.

   If you do not need a printer, this has already been defined for you in the IPLUNAM parameter of the NDMTCP customization macro (as IPLUNAM=NONE), and **NONE** is displayed in this field.

4. When you press Enter, the **Transmission Control Master Menu** is displayed.

**What Could Go Wrong**

You will receive an error message in the following circumstances:

- If the operator ID or password you enter is not valid, NetView DM for MVS rejects your logon. You must log on again from VTAM/TCAM or NCCF/TAF.

- If you enter a printer name that is not valid, you receive a message and you must enter a valid name. If you enter the name of a printer currently not available, you also get a message, but you do not have to enter a new name. The printer will be used for your print requests as soon as it is available.

- If NetView DM for MVS rejects your logon because you are not authorized to use IOF. Contact your system administrator, or whoever is responsible for granting authorizations.
The Transmission Control Master Menu

The illustration below shows the Master Menu which is displayed after you log on to IOF, and any time you press PF4 or PF16 from an IOF panel.

<table>
<thead>
<tr>
<th>NDMOC010</th>
<th>TRANSMISSION CONTROL MASTER MENU</th>
<th>PAGE 1 OF 1</th>
</tr>
</thead>
</table>

Select one of the following:

1. START TRANSMISSION
2. SCHEDULE DISPLAY AND CONTROL
3. HOLD TRANSMISSION FOR A NODE/GROUP
4. RELEASE TRANSMISSION FOR A NODE/GROUP
5. QUIESCE TRANSMISSION
6. IMMEDIATE QUIESCE TRANSMISSION
7. END TRANSMISSION
8. IMMEDIATE END TRANSMISSION
9. DISCONNECT IOF
10. LOGOFF FROM IOF
11. DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES

For selections 3, 4 and 11 enter the required parameters (see Help)

Selection ==>
PF= 1/13=HELP 5/17=MSG-UNS
9/21=PRINT
M= 3 STATUS= RUNNING  TRANSM= 15  PRINTER= AVAIL

Figure 123. The IOF Master Menu

Selecting from the Master Menu

The way in which you select options from the Master Menu varies for each option.
The information below shows you which chapter of this book to turn to for information about each menu option.

Option 1, START TRANSMISSION, is described in “Option 1 – Start Transmission” on page 494.

Option 2, SCHEDULE DISPLAY AND CONTROL, is described in Chapter 21, “Controlling Transmission Schedules” on page 495.

Option 3, HOLD TRANSMISSION FOR A NODE/GROUP, is described in “Option 3 – Hold Transmission for a Node/Group” on page 515 in Chapter 3, “Defining Nodes” on page 33.

Option 4, RELEASE TRANSMISSION FOR A NODE/GROUP, is described in “Option 4 – Release Transmission for a Node/Group” on page 518 in Chapter 3, “Defining Nodes” on page 33.

Options 5 through 10 are described in Chapter 24, “Stopping Transmission and the Transmission Environment” on page 529.
Option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES, is described in Chapter 23, “Changing Transmission Profiles” on page 521.

What Could Go Wrong

If you log on to IOF but then decide that you do not want to start transmission operations after all, select Option 7, END TRANSMISSION, to log off from IOF and end the TCP. Option 10, LOGOFF FROM IOF, would be rejected in this case because the system status is QUIESCED. Refer to “System Line” on page 491 which describes the system status possibilities.

IOF Panel Layout

IOF panels generally have the following layout:

<table>
<thead>
<tr>
<th>PANEL ID</th>
<th>PANEL TITLE</th>
<th>PAGE_OF_</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGE</td>
<td>Here you receive error and information messages</td>
<td></td>
</tr>
<tr>
<td>INSTRUCTIONS</td>
<td>This line tells you how to use the panel</td>
<td></td>
</tr>
</tbody>
</table>

INPUT/OUTPUT AREA
This area may contain:
- A list of options, as on the Master Menu
- Data entry fields
- Output fields providing information about the transmission operations.

INSTRUCTIONS These are two more lines for further instruction. On some panels they are part of the input/output area.

Selection ==> Here you enter the selected option number and/or data.
PFKEYS The PF keys you can use to enter commands are displayed here. See “PF Keys” on page 490.

TCP STATUS INFORMATION LINE See “System Line” on page 491.

Figure 124. IOF Panel Layout

Line 1 of every panel identifies the panel and the associated IOF procedure. The upper right-hand corner shows how many pages of information there are, and which page you are looking at now.
PF Keys

Use the PF keys on your keyboard to enter predefined commands. The PF keys you can use, and their associated commands, are displayed in lines 21 and 22 of each panel. Only the PF keys that you can use on the panel currently displayed are shown. The following explanation of the PF keys shows two numeric values:

- If you have 12 PF keys on your terminal, use the lower number.
- If you have 24 PF keys, use the higher number.

If you do not have PF keys, enter the PF key number in the `PF=__` field located at the beginning of the PF key line.

You can use the following PF keys:

- **PF1/13=HELP** Displays help information about the current panel.
- **PF3/15=END** Ends your current operation and panel, returning you to the previous level panel.
- **PF4/16=RETURN** Returns you to the Transmission Control Master Menu.
- **PF5/17=MSG-UNS** Displays the unsolicited messages waiting to be viewed. The list shown starts with the last message that you viewed.
- **PF6/18=REFRESH** Updates the displayed panel with new information resulting from NetView DM for MVS activities.
- **PF7/19=BACK** Displays the previous page of information. If you are already on the first page, the request is ignored.
- **PF8/20=FORW** Displays the next page of information. If you are already on the last page, the request is ignored.
- **PF9/21=PRINT** Prints a copy of the current panel on the printer associated with IOF. If the printer is not available, the request is not processed and a message is issued.
- **PF10/22=TOP** Displays the top of the file on the Handle Unsolicited Messages panel.
- **PF11/23=LAST VIEWED** Displays the last message that you viewed on the Handle Unsolicited Messages panel.
- **PF12/24=BOTTOM** Displays the last message written to the unsolicited message file on the Handle Unsolicited Messages panel.

There are two commands, LEFT and RIGHT, which you can use on the List the Status of all Functions of a Phase panel, and on the Display and Control Transmission Schedules panel, if the D&CC API feature is installed. Use the following function keys:

- **PF10/22=LEFT** Displays the left side of the panel.
- **PF11/23=RIGHT** Displays the right side of the panel.
System Line
The last line on every panel is the system information line. It looks like this:

\[
\begin{array}{c}
M=3 \\
STATUS=RUNNING \\
TRANSM=5 \\
PRINTER=AVAIL
\end{array}
\]

The fields provide the following information about the system:

\[\text{M=} \quad \text{The number of unsolicited messages not yet viewed.}\]

\[\text{STATUS=} \quad \text{The status of the NetView DM for MVS transmission activity. The status determines whether a command you enter from the Master Menu is valid. As the status of transmission is continually changing, the TCP regularly refreshes the status line, so that the most up to date status is displayed. This may be too frequent, preventing you from actually entering a command. You can change the rate at which the TCP refreshes the status line using the STALINE parameter of the NDMTCP macro (refer to the Installation and Customization manual).}\]

The status, which also changes when you enter a command from the IOF, can be one of the following:

\[\text{RUNNING:} \quad \text{One or more phases or NSRs are transmitting data between NetView DM for MVS and remote nodes.}\]

\[\text{WAITING:} \quad \text{No phases or NSRs are currently active. NetView DM for MVS is waiting because there is no work to be performed at the moment. Transmission begins automatically when a phase is ready, and when the node sends a request.}\]

\[\text{QUIESCED:} \quad \text{Transmission activity has stopped, either because you requested it, or because you logged on but have not yet started transmission. Transmission activity starts when you enter the Start Transmission command.}\]

\[\text{QUIESCING:} \quad \text{You have requested to quiesce transmission activity, using either the Quiesce Transmission or the Immediate Quiesce Transmission command, but not all activities have stopped yet.}\]

\[\text{ENDING:} \quad \text{You have requested to end the TCP, using either the End Transmission or the Immediate End Transmission command, and the TCP is in the process of ending all active transmission requests. When transmission activity has stopped, you are logged off from the IOF (if you are still logged on), and the TCP ends.}\]

\[\text{TRANSM=} \quad \text{The number of transmission operations that the TCP is currently executing.}\]

\[\text{PRINTER=} \quad \text{The status of the printer:}\]

\[\text{AVAIL:} \quad \text{The printer is available.}\]

\[\text{UNAVAIL:} \quad \text{The printer is not available.}\]

\[\text{INTREQ:} \quad \text{The printer is available, but there is a problem that requires the operator to intervene before printing can start.}\]
Using IOF with a Katakana Terminal

You can use IOF with a Katakana terminal, keyboard, and printer. The IOF panels will display all characters in English, using uppercase characters only.

Getting Help

Whenever you are not sure about what to do on an IOF panel, you can press PF1/13 to display some help text that explains the panel. To leave the help panel and return to the panel from which you asked for help, press PF3/15 again.
IOF Operator Functions and Panel Flow

Figure 125 shows the operator functions and the panels you use to perform the functions:

LOGON through VTAM, through VTAM/TCAM, or through NCCF/TAF

<table>
<thead>
<tr>
<th>IOF OPERATOR LOGON see Chapter 20</th>
</tr>
</thead>
</table>

TRANSMISSION CONTROL MASTER MENU

1 START TRANSMISSION Chapter 20
2 SCHEDULE DISPLAY AND CONTROL
3 HOLD TRANSMISSION FOR A NODE/GROUP Chapter 22
4 RELEASE TRANSMISSION FOR A NODE/GROUP
5 QUIESCE TRANSMISSION
6 IMMEDIATE QUIESCE TRANSMISSION
7 END TRANSMISSION Chapter 24
8 IMMEDIATE END TRANSMISSION
9 DISCONNECT IOF
10 LOGOFF FROM IOF
11 DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES Chapter 23

Chapter 21

SCHEDULE DISPLAY AND CONTROL

Select a transmission request or a list of requests

DISPLAY AND CONTROL SELECTED TRANSMISSION REQUESTS

Control phases/NSRs: change schedule, delete, hold, release, restart

List conditioning or conditioned phases

List all functions of a transmission request

LIST OF FUNCTIONS

View all functions of a transmission request

LIST OF SELECTED PHASES

Control phases with the same commands as above

Figure 125. IOF Operator Functions and Panel Flow
Exiting the IOF

Option 1 – Start Transmission

After you have logged on to the IOF, you are ready to start transmission activities. If you specified AUTOSTR=YES, transmission begins automatically. If you defined the TCP startup parameter AUTOSTR=NO, to start transmission with a command from the IOF, you must either select Option 1, START TRANSMISSION, from the Master Menu, or use an MVS MODIFY command to start transmission.

You also use Option 1 after a previous quiesce request, using Option 5 or 6, in order to restart transmission. Refer to Chapter 24, “Stopping Transmission and the Transmission Environment” on page 529 for information about menu options 5 and 6.

You can only select Option 1 if the system status shown on the panel is:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Start Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUIESCED</td>
<td>Restartable transmission requests not previously executed are placed in the READY or WAITING status. READY transmission requests are started. PENDING transmission requests are started if a request comes from a node. If no transmission requests are READY, the system status is changed to WAITING. Transmission starts as soon as a transmission request becomes READY or an NSR comes from a node.</td>
</tr>
</tbody>
</table>

Using an MVS MODIFY Command to Start Transmission

Use the following MVS MODIFY command from the system console to start transmission activities:

MODIFY | F jobname,START

What Could Go Wrong

If the NetView DM for MVS system status shown on the Master Menu is not QUIESCED, a message is displayed on the message line.

Logging off from the IOF

There are two ways in which you can log off from the IOF. Which option you use depends on whether you have already started transmission activities.

If you log on to IOF but then decide that you do not want to start transmission after all, select Option 7, END TRANSMISSION to log off from IOF, as the system status is QUIESCED when you first log on.

After you start transmission, which is described in “Option 1 – Start Transmission,” use Option 10, LOGOFF FROM IOF, to log off from the IOF.
Chapter 21. Controlling Transmission Schedules

Once a user has submitted a transmission plan for processing, or the TCP has received an NSR or D&CC API request, you can look at the status of the transmission, and change the schedule if you need to. This chapter explains how to display and control transmission schedules. You do this in two steps:

**Step 1.** Select the transmission requests you want to work with.

**Step 2.** Enter commands to change the schedule of a phase, or to display the status of all functions of a phase.

You can also display the status of the transmission for a node, or for all functions of a phase, using MVS MODIFY commands from the system console (listed in Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549). The display MVS MODIFY command is available for LU 6.2 and NetView DM Transfer (NDMT) protocols.

Figure 126 shows the panel flow for Option 2, SCHEDULE DISPLAY AND CONTROL.

---

**Figure 126. Option 2 – SCHEDULE DISPLAY AND CONTROL**
Option 2 – Schedule Display and Control

After you have logged on to IOF and started transmission:

- Select Option 2, SCHEDULE DISPLAY AND CONTROL, from the Master Menu.

The Display and Control Transmission Schedules panel is displayed.

![Figure 127. Example of the Display and Control Transmission Schedules Panel](image)

**Note:** The D&CC API is an optional feature of NetView DM for MVS. The API Requests field will display the message 'D&CC API not available'. You should ignore the instructions relating to D&CC API requests.

You can use the Display and Control Transmission Schedules panel to display phases, NSRs, D&CC API requests, or all types of transmission request in the TCF.

**To display all transmission requests in the TCF:**

Press Enter without specifying any selection criteria.

Alternatively, you can tailor the list that will be displayed by using the fields as selection criteria. Each option on this panel defines the selection criteria you want to use to select transmission requests. The selection criteria you can use are divided into four sets of fields:

**Fields 1–3** Use these fields to select the *type* of transmission requests you want to display.
Option 2 – Schedule Display and Control

Fields 4–8  Use these fields to select specific transmission requests of the type you selected.

Fields 9–12  Use these fields to select transmission requests scheduled within a specific date or time range.

Fields 13–18  Use these fields to select transmission requests that have a particular status in the TCF.

Selecting the Type of Transmission Request

Use fields 1–3 to select the type of transmission request you want to see. You can display one type of transmission request, or a combination of types. You can select:

- Phases: Enter an N in the NSRs and API requests fields to suppress display of NSRs and D&CC API requests. Or, you can suppress one and display a combination of phases and NSRs for example.
- NSRs: Enter an N in the Phases and API Requests fields to suppress display of phases and D&CC API requests.
- D&CC API requests: Enter an N in the Phases and NSRs fields to suppress display of phases and NSRs.

Selecting Specific Phases, NSRs, or D&CC API Requests

Use fields 4–8 if you want to refine the list further. The fields you can use depend on the type of transmission request you selected (phases, NSRs or D&CC API requests). The fields you can use with each type of request are shown in the following tables.

For phases, follow these instructions:

<table>
<thead>
<tr>
<th>What You Want to Display</th>
<th>What You Must Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>A specific phase</td>
<td>Enter the plan and phase name in the Plan and phase names field</td>
</tr>
<tr>
<td>All phases for a specific node</td>
<td>Enter a node name in the Node/NSR origin name field</td>
</tr>
<tr>
<td>All phases for a specific node group</td>
<td>Enter a node group name in the Group name field</td>
</tr>
<tr>
<td>All phases for a specific transmission profile</td>
<td>Enter a transmission profile name in the Transm. Profile name field</td>
</tr>
</tbody>
</table>

Table 64. Displaying Selected Phases

For NSRs, follow these instructions:
Option 2 – Schedule Display and Control

What You Want to Display | What You Must Enter
---|---
NSRs from a specific node | Enter an NSR origin name in the **Node/NSR origin name** field
NSRs from a specific node group | Enter a node group name in the **Group name** field
NSRs for a specific transmission profile | Enter a transmission profile name in the **Transm. Profile name** field

Table 65. Displaying Selected NSRs

For D&CC API requests, follow these instructions:

<table>
<thead>
<tr>
<th>What You Want to Display</th>
<th>What You Must Enter</th>
</tr>
</thead>
<tbody>
<tr>
<td>All D&amp;CC API requests for a specific node</td>
<td>Enter a node name in the <strong>Node/NSR origin name</strong> field</td>
</tr>
<tr>
<td>D&amp;CC API requests for a specific node group</td>
<td>Enter a node group name in the <strong>Group name</strong> field</td>
</tr>
<tr>
<td>D&amp;CC API requests from a specific user program</td>
<td>Enter the name of the program in the <strong>API user program name</strong> field</td>
</tr>
<tr>
<td>D&amp;CC API requests for a specific transmission profile</td>
<td>Enter the transmission profile name in the <strong>Transm. Profile name</strong> field</td>
</tr>
</tbody>
</table>

Table 66. Displaying Selected D&CC API Requests

Selecting Transmission Requests Within a Date or Time Range

You can use fields 9–12 to specify a time or date range for the transmission requests. You can use these fields for all types of requests. Each field is optional; you can enter just one field, or several fields in combination.

Scheduled Date From and Scheduled Date To

Follow the instructions below to display all transmission requests, of the type you selected, that are scheduled within a specific date range:

1. Enter the scheduled dates in the **Scheduled date from** and **Scheduled date to** fields, in the form MMDDYY, where:
   - **MM** = month 2 digits, from 01 to 12.
   - **DD** = day 2 digits, from 01 to 31.
   - **YY** = year 2 digits, from 00 to 99.
2. Press Enter to display the selected transmission requests.

Scheduled Time From and Scheduled Time To

Follow the instructions below to display all transmission requests, of the type you selected, that are scheduled within a specific time range:

1. Enter the scheduled times in the **Scheduled time from** and the **Scheduled time to** selection fields. You can enter **NOW** to select the current time, or you can specify a time in the form HHMM, where:
   - **HH** = hour 2 digits, from 00 to 23.
   - **MM** = minutes 2 digits, from 00 to 59.
Option 2 – Schedule Display and Control

If you specify NOW, you do not have to enter a date. NOW means the current system time and the current date.

2. Press Enter to display the selected transmission requests.

Default Values
If you do not enter a date or time, all the transmission requests will be displayed.

If you enter a date but no time, NetView DM for MVS assumes the following defaults:
FROM-TIME defaults to 0000
TO-TIME defaults to 2359

If you enter a time but no date, the date defaults to the current date.

Selecting Transmission Requests With a Specific Status
You can use fields 13–18 to select transmission requests which have a particular status. You can use these fields for all types of requests. Each field shows a specific status for the transmission request. You choose which transmission requests you do not want displayed, by suppressing the display:

1. Enter N in the entry field for any status you do not want to display.

   If you leave a field blank, NetView DM for MVS will display requests with that status.

2. When you have completed your entries, press Enter.

The status can be one of the following:

EXECUTING When the TCP is executing the transmission request.

READY Transmission requests that the TCP cannot execute, even though their scheduled date and time has passed. The TCP cannot execute them because either the maximum number of concurrent SNA sessions, or the involved logical unit, is already active.

WAITING Transmission requests that the TCP will start to process as soon as they reach the scheduled date and time. Waiting phases do not include phases that have passed their scheduled time and that are now ready for initiation.

HELD Transmission requests that either you held, or that the system holds for one of the reasons described below. If a transmission request is held, the status shown for the request on the panel may be one of the following:

   HELED Appears if you held phases or D&CC API requests explicitly with the H (Hold) command on the Display and Control Selected Phases panel. Sometimes, users may have defined the phase as held when creating the transmission plan.

   HDEST Appears if you held the node for the transmission request. Nodes are held when you select Option 3, HOLD TRANSMISSION FOR A NODE on the Master Menu.
Option 2 – Schedule Display and Control

HRETR  The node for the transmission request has the HRETR status between retries when there are line problems or a node shutdown.

HSYST  The status becomes HSYST when the TCP reaches the maximum number of retries and still cannot establish transmission.

HCOND  Appears if the TCP holds a phase because the start of transmission is conditioned by the processing of another phase.

Note:  You can hold a phase or D&CC API request explicitly (HELD), or the system can hold a phase or D&CC API request implicitly (HDEST, HRETR, HSYST, HCOND, and HWIN), at the same time. If so, the status shown is the one with the highest priority:

- HELD overrides HDEST, HRETR, HSYST, and HWIN
- HDEST overrides HCOND.

HWIN  This status applies to phases that have transmission windows defined. The status becomes HWIN if the phase does not reach pending status by the end of the defined window. The phase remains in HWIN status until the start of the next window. The node addressed by the phase is not held.

RESTARTABLE  This can apply to either of the following:

- Phases that have ended with a temporary error, that might disappear if the phase is run again
- Transmission requests of any type that have been interrupted by an Immediate Quiesce command.

PENDING  Transmission requests containing at least one function in pending status. A function is in a pending status when NetView DM for MVS has delivered it to a node and is waiting for a delayed acknowledgment or data from the node.

Note:  A transmission request can have the temporary status COMPL for completed, or DELET for deleted, when an EXECUTING transmission request completes, or when you enter a Delete command on the Display and Control Selected Phase panel. These disappear when you perform a List Refresh, using PF6/18. NetView DM for MVS rejects any other operation on completed or deleted phases, and issues a message on the error message line. The panel does not display phases that the TCP has deleted automatically, due to a phase cut off.

Examples  Here is an example that shows you how to combine fields on the Display and Control Transmission Schedules panel to display specific transmission requests.

To display all of the phases in the TCF that are PENDING for a specific node, and that are scheduled during a specific time range, enter the following data in these fields:
1. Enter an N in the **NSRs** and **API Requests** fields. Leave the **Phases** field blank to display phases only.

2. Enter the name of the node in the **Node/NSR origin name** field.

3. Enter the scheduled date and time in fields 9 through 12.

4. Enter an N in fields 13, 14, 15, 16, and 17. Leave field 18 (**Pending status**) blank, to select phases that are pending.

**What Could Go Wrong**

If you make a mistake when entering data on this panel, a message appears in the message line describing the error. The message line displays what you entered, so you can correct it.

NetView DM for MVS also issues a message if there are no transmission requests that fit your selection criteria.
Displaying Selected Transmission Requests

If NetView DM for MVS finds transmission requests to match your selection criteria, it displays the information on one or more pages, depending on the number of transmission requests found.

Figure 128 is an example of a Transmission Display and Control panel displaying selected phases, NSRs, and D&CC API requests. With this panel, you can review the transmission requests selected and then change their schedules by entering a line command. “Changing the Transmission Schedule” on page 503 describes line commands.

Information Shown for Each Transmission Request

This panel shows one line of information for each selected transmission request.

Status: The status information is shown in order of priority. All transmission requests with the same status are grouped together. “System Line” on page 491 shows you the possible statuses and what they mean. The panel displays the status in an abbreviated form, as follows:

- EXEC: Executing
- READY: Ready
- WAIT: Waiting (phases only)
- HELD: Held by operator (phases and D&CC API requests only)
- HDEST: Held by operator

Note: There is one other field that is not displayed on the panel above. Use the PF11 key to display the TRANSM. PROFILE field by scrolling to the right of the screen.
### Changing the Transmission Schedule

By the time NetView DM for MVS processes the command you specify, the status of the transmission request may have changed. For example, an executing phase may have completed. In this case, IOF may reject the command you enter. To avoid this, press PF6/18 from time to time to refresh the transmission request list with the latest status information.

**NSR Origin-Sequence or Node:** This field displays the name of either of the node that originated an NSR, or the name of the node referred to in a transmission plan, phase, or D&CC API request.

**Plan:** This field displays the name of the transmission plan containing the transmission request.

**Phase:** This field displays the name of the phase that is being processed.

**API User Program:** This field displays the name of a D&CC API user program that originated a D&CC API transmission request.

**Note:** This field only displays data when the D&CC API optional feature is being used.

**Scheduled/Arrival Date and Time:** For phases only, this is the scheduled date and time of the phase. This can change as a result of the commands you enter.

**Start Date and Time:** These fields show when the TCP started transmission of a transmission request.

**Transmission profile:** This field is on the far right-hand side of the panel. You will probably need to scroll to the right using the PF11 key. The field displays the name of the transmission profile that the logical unit addressed by the transmission request belongs to.

## Changing the Transmission Schedule

The top of the Transmission Display and Control panel shows you a list of line commands. These are the commands you can use to change the transmission schedule that is displayed. The line commands you can use are:

- **C(Change)**: Change the start time and date (phases only)
- **D(Delete)**: Delete a phase for node, NSR, or D&CC API request
- **H(Hold)**: Hold transmission of a phase or D&CC API request
- **R(Release)**: Release a held phase or D&CC API request
- **S(reStart)**: Restart a phase
- **L(List)**: List status of all functions/resources of a phase for node, NSR, or D&CC API request
Changing the Transmission Schedule

I(conditioning) Display a list of conditioning phases
E(conditionEd) Display a list of conditioned phases

The following pages describe these line commands, in alphabetical order.

Using Line Commands

The example below shows you how to use line commands to change a transmission schedule:

1. Enter a line command in the Command field, for as many of the displayed transmission requests as you want.
   If the display carries over to other pages, you can page forward with PF8/20, and backward with PF7/19.
2. Press Enter when you want to begin processing commands. The first line with a command is highlighted.
3. You can process this command by pressing Enter again, or cancel the request by erasing the command and pressing Enter.

After the IOF has processed the command, a response appears to the right of the command itself. This can be either REJ or ACC. REJ indicates that the requested action was not accepted because of an error or you canceled the requested action. For example, a phase you wanted to delete has already completed. The panel shows a message explaining why the command has been rejected. ACCEPTED indicates that you confirmed the requested action and that it was accepted.

As you confirm or cancel each request, the next line with a command not yet processed is highlighted. If all commands on a page have been processed, the first command on the next page is automatically highlighted. You can page backward and forward while processing commands, if you want to enter new commands. When you press Enter, the first command to be processed in the list will be highlighted.

What Could Go Wrong

If you enter an invalid command, that is, a character other than one shown on the panel, the invalid command is highlighted. You have to correct it before you can continue processing another command.

You can enter more than one command on the same phase (for example, a Hold command followed by a Delete command). When you do this, the execution of the commands is delayed because the TCP has to wait for a reply from the node for each command.

C – Change the Start Time and Date

This command changes the scheduled start time or date, or both, of phases.

Note: This command applies to phases only.

You can only change the schedule of a phase if its status is:
Changing the Transmission Schedule

Note: If you are changing the start time and date of a phase that specifies a group of nodes that share the same next intermediate node, the following limitations apply:

<table>
<thead>
<tr>
<th>Phase Status</th>
<th>Effect of Change Date and Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELD, HDEST, HRETR,</td>
<td>The change is made. The phase or NSR status is not changed.</td>
</tr>
<tr>
<td>HSYST, HCOND, HWIN</td>
<td></td>
</tr>
<tr>
<td>WAIT, READY</td>
<td>The change is made. If the new date and time are in the past, the phase is READY for processing. If the new date and time are in the future, the phase is WAITING.</td>
</tr>
<tr>
<td>REST</td>
<td>The change is made. The status remains RESTARTABLE.</td>
</tr>
</tbody>
</table>

Note: You cannot change the start time or date, or both, of phases which contain Send or Initiate functions to LU 0 end nodes connected to the host through an intermediate node.

What Could Go Wrong

The command is rejected if:

- The date or time is invalid. A message appears on the screen. You can correct the input and press Enter again, or you can cancel the command.

- The new date and time is beyond the expiration date and time of the phase (for recursive plans only). A message appears on the screen and the date and time field is highlighted. You can correct the input and press Enter again, or you can cancel the command.

- The date or time is beyond the cut-off date and time.

- The phase you want to change addresses a node for which fanout has to take place.

- The status of the phase is EXEC, PEND, COMPL, or DELET.
**Changing the Transmission Schedule**

### D – Delete a Transmission Request

You can delete a phase, NSR, or D&CC API request if the status is:

<table>
<thead>
<tr>
<th>Transmission Request Status</th>
<th>Effect of Delete Transmission Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The transmission request is interrupted at the next checkpoint and the status is changed to DELET.</td>
</tr>
</tbody>
</table>
| PEND                        | For all LU 6.2 nodes, and LU 0 nodes that are *not* using intermediate node support, the transmission request is deleted immediately and the status changed to DELET.  
For VSE nodes, the execution of the command can result in an error message if a previous incompatible command is in progress or the phase status is changed at the node. If the command is accepted, the phase is left pending until the command is executed at the node.  
For LU 0 nodes using intermediate node support, the phase is left in its original status until clean-up operations are executed at the front end node. |
| READY, WAIT, HELD, HDEST, HRETR, HSYST, HCOND, REST, HWIN | The transmission request is deleted.  
If you are deleting a phase or D&CC API request that specifies a group of LU 6.2 nodes that share the same next intermediate node, the following limitations apply: |

<table>
<thead>
<tr>
<th>Phase/D&amp;CC API Request Status</th>
<th>Effect of Delete Phase or D&amp;CC API Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The phase for node or D&amp;CC API request is deleted regardless of whether other phases for nodes or D&amp;CC API requests are deleted, for those nodes that share the same next intermediate node.</td>
</tr>
<tr>
<td>Previously EXEC but transmission suspended (by REST, HDEST, HRETR, HSYST)</td>
<td>When the transmission is restarted, the distribution completes to all the nodes rather than deleting the phases for nodes that share the same next intermediate node. This is because it is difficult to determine whether or not the distribution list has already been sent to the intermediate node. If all the phases for nodes sharing the same next intermediate node have already been deleted, then transmission cannot be restarted, however.</td>
</tr>
<tr>
<td>HELD, READY</td>
<td>The phase for node is deleted. The distribution list will not include this node name.</td>
</tr>
<tr>
<td>PENDING</td>
<td>The phase for node or D&amp;CC API request is deleted, causing reports or files from the nodes to be discarded.</td>
</tr>
</tbody>
</table>

The panel shows the status of deleted transmission requests as DELET, until you refresh the list by pressing PF6/18. They then disappear from the list. You cannot schedule a deleted transmission request again.
What Could Go Wrong
IOF rejects the command if:

- The status of the transmission request is COMPL or DELET. IOF issues an error message.
- You are not authorized to delete transmission requests. The system administrator defines your user authorization in your user profile.

H – Hold Transmission of a Phase or D&CC API Request
This command holds the execution of a phase, or a D&CC API request, until you release it with an R command. If you enter this command against a phase that has windows defined, the phase is held and only released when you release it with the R command. The windows then take effect as usual.

Note: This command does not apply to NSRs.

You can also hold a phase by node using an MVS MODIFY command (see Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549).

You can only hold transmission if the status of the phase or D&CC API request is:

<table>
<thead>
<tr>
<th>Phase/D&amp;CC API Request Status</th>
<th>Effect of Hold Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The phase or D&amp;CC API request is interrupted at the end of the current function and the status is changed to HELD.</td>
</tr>
<tr>
<td>REST, HDEST, HCOND, HRETR, HSYST, HWIN</td>
<td>The status of the phase or D&amp;CC API request is changed to HELD.</td>
</tr>
<tr>
<td>PEND</td>
<td>If the node is a VSE node and it acknowledges the hold request, the status of the phase, or D&amp;CC API request, is changed to HELD. If the VSE node does not accept the hold request, the request is canceled, the phase, or D&amp;CC API request, remains pending, and a message is issued. If the node is not a VSE node, the hold command is accepted and becomes effective when the pending function completes. If the phase, or D&amp;CC API request, is for an LU 6.2 type node and it has been delivered to the node, the hold command is accepted and becomes effective when the pending function completes.</td>
</tr>
</tbody>
</table>
Changing the Transmission Schedule

If you are holding a phase, or a D&CC API request, that specifies a group of LU 6.2 nodes that share the same next intermediate node, the following limitations apply:

<table>
<thead>
<tr>
<th>Phase/D&amp;CC API Request Status</th>
<th>Effect of Hold Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The phase for node, or D&amp;CC API request, is HELD only after the currently EXECUTING function finishes (just like phases for nodes, or D&amp;CC API requests, that do not share the same next intermediate nodes, but with other nodes addressed by the request).</td>
</tr>
<tr>
<td>EXEC and then HRETR, HDEST, HSYST</td>
<td>A phase for node, or D&amp;CC API request is HELD only after transmission is restarted and the currently EXECUTING function completes.</td>
</tr>
<tr>
<td>READY</td>
<td>The phase, or D&amp;CC API request, is HELD.</td>
</tr>
<tr>
<td>PENDING</td>
<td>The phase, or D&amp;CC API request, is HELD only when the currently executing function completes.</td>
</tr>
</tbody>
</table>

What Could Go Wrong

IOF ignores the command if the phase or D&CC API request status is already HELD. If the phase or D&CC API request status is COMPL or DELET, you will receive an error message in the message line.

R – Release a Held Phase or D&CC API Request

This command releases a phase, or D&CC API request, that you have previously held with an H command. Releasing a request does not affect the scheduled date or time of the request.

You can also release a phase by node using an MVS MODIFY command (see Appendix A, “Using MVS MODIFY Commands with NetView DM for MVS” on page 549).
You can only release a request when the status of a phase or D&CC API request is:

<table>
<thead>
<tr>
<th>Phase/D&amp;CC API Request Status</th>
<th>Effect of Release Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELD</td>
<td>The status of the phase, or D&amp;CC API request, is changed to READY or WAIT depending on the scheduled date and time.</td>
</tr>
<tr>
<td>HELD and HDEST</td>
<td>The status of the phase, or D&amp;CC API request, is changed to HDEST.</td>
</tr>
<tr>
<td>HELD and HRETR</td>
<td>The status of the phase, or D&amp;CC API request, is changed to HRETR.</td>
</tr>
<tr>
<td>HELD and HSYST</td>
<td>The status of the phase, or D&amp;CC API request, is changed to HSYST.</td>
</tr>
<tr>
<td>HELD and HCOND</td>
<td>The status of the phase, or D&amp;CC API request, is changed to HCOND.</td>
</tr>
<tr>
<td>HELD and PEND</td>
<td>For all node types except VSE, the command is accepted and the transmission request is held as soon as the currently pending function completes, as long as it is not the last function. If it is the last function, the transmission request is completed. For VSE nodes, the status of the transmission request is changed to PEND after the node has acknowledged the release request. If the node does not accept the request, the release request is canceled, the phase or D&amp;CC API request remains HELD, and a message is issued.</td>
</tr>
</tbody>
</table>

Some LU 0 node types have the capability of remotely releasing a phase that has been held at the NetView DM for MVS host. These nodes are:

- DPPX
- PC/DSNX
- SSP
- VSE

These LU 0 nodes can issue a “release a session” command to NetView DM for MVS. LU 6.2 node types can send NSRs to NetView DM for MVS.

**What Could Go Wrong**

The node ignores the command if the status of the phase is not HELD. An error message is issued to the unsolicited messages file (if available), and to SYSPRINT.

IOF rejects the command if the phase is in HWIN status. You can remove transmission windows only by resetting the plan.

**S – Restart a Phase**

This command tries to execute a phase again. You cannot use this command to restart phases for VSE nodes. The TCP retries phases for VSE nodes automatically, on request from the node program.
Changing the Transmission Schedule

You can restart a phase only if its status is:

<table>
<thead>
<tr>
<th>Phase Status</th>
<th>Effect of Restart Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST</td>
<td>The phase is made READY for processing, unless its schedule has been changed prior to issuing the restart command.</td>
</tr>
</tbody>
</table>

What Could Go Wrong

IOF rejects the command if the phase is not restartable. The panel displays a message.

L – List Status of All Functions of a Transmission Request

Use the L command if you want to see the status information for all the transmission functions of a phase, NSR, or D&CC API request. The List the Status of all Functions of a Phase, NSR, or API Request panel, shown in Figure 129, displays this status information.

<table>
<thead>
<tr>
<th>Function information for phase/NSR-Seq/API user program: SDMF001_node:PSNX001</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION STATUS RC BYTES RESOURCE NAME</td>
</tr>
<tr>
<td>RET DATA RUNN 0 4324 NDM.NDMDAT1</td>
</tr>
</tbody>
</table>

| M=    | 2         | STATUS=RUNNING | TRANSM= 2 | PRINTER=AVAIL |

Figure 129. Example of the List the Status of All Functions of a Phase, NSR, or API Request Panel

For each transmission request the panel shows the following fields:

**FUNCTION:** This is the abbreviated form of the NetView DM for MVS functions and resources. Functions are identified with three characters, resources with two, three, or four, shown in the following table.
### Changing the Transmission Schedule

#### Table 67. Abbreviated Forms of Functions and Resources

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>RESOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLE</td>
<td>ABEN</td>
</tr>
<tr>
<td>DEL*</td>
<td>CATE</td>
</tr>
<tr>
<td>INI</td>
<td>CATU</td>
</tr>
<tr>
<td>RET*</td>
<td>CLIS</td>
</tr>
<tr>
<td>SEN*</td>
<td></td>
</tr>
<tr>
<td>ACC*</td>
<td>DATA</td>
</tr>
<tr>
<td>ACT*</td>
<td>DSCB</td>
</tr>
<tr>
<td>INS*</td>
<td>ERRO</td>
</tr>
<tr>
<td>REM*</td>
<td>FILE</td>
</tr>
<tr>
<td>UNI*</td>
<td>FILE</td>
</tr>
<tr>
<td></td>
<td>FILE</td>
</tr>
<tr>
<td></td>
<td>FP</td>
</tr>
<tr>
<td></td>
<td>FSP</td>
</tr>
<tr>
<td></td>
<td>JOB</td>
</tr>
<tr>
<td></td>
<td>LIB</td>
</tr>
<tr>
<td></td>
<td>MEMB</td>
</tr>
<tr>
<td></td>
<td>MICR</td>
</tr>
<tr>
<td></td>
<td>PANE</td>
</tr>
<tr>
<td></td>
<td>PRIN</td>
</tr>
<tr>
<td></td>
<td>PO</td>
</tr>
<tr>
<td></td>
<td>PROG</td>
</tr>
<tr>
<td></td>
<td>PTF</td>
</tr>
<tr>
<td></td>
<td>DUMP</td>
</tr>
<tr>
<td></td>
<td>AS400CTN</td>
</tr>
<tr>
<td></td>
<td>AS400OBJ</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: The abbreviations with an asterisk denote functions which apply to LU 6.2 resource types only.

**STATUS:** The status of the functions is abbreviated to:

- **COMPL** Completed successfully or unsuccessfully as indicated by the NetView DM for MVS return code
- **PEND** Pending
- **REST** Restartable
- **RUNN** Running
- **WAIT** Waiting. This panel also shows WAIT if the TCP did not process a function because it terminated the sequence.
**RC:** The NetView DM for MVS return code.

The return code values that the IOF specifies for function termination for all node types (except VSE nodes) have the following meanings:

00 Function completed successfully, or function RESTARTABLE (for example, after a line failure or IOF interruption). When a function is RESTARTABLE, resources partially retrieved in the repository are flagged as INCOMPLETE and are not available for use.

04 Function completed, but warning messages were issued (for example, the operation was DELETE for a data set that does not exist). However, the TCP performed the requested operation.

06 Function was not completed due to a temporary error. With this type of error, the function is restartable. When the TCP runs the function again, the error could disappear and the function then would be completed successfully. Here are some examples of temporary errors:

- A data set specified in the plan was not in the holding file.

  **Note:** For user-defined type nodes (CMFP, CMEP and NDMT), the return code for this type of error is 08, and not 06.

- A VSAM error occurred in the holding file.

- The printer was not available at the node for a send print function.

08 Function completed with an error that is not temporary. This type of error is not restartable, so the TCP will not try to run the function again. An example of this type of error is when a logical error exists in a plan. Any resources partially retrieved in the repository are canceled. NetView DM for MVS always issues this return code when it receives a negative delayed acknowledgment from a node working in asynchronous mode.

For VSE nodes, there are only two return codes:

00 Function completed successfully, or the function is restartable.

08 Function was not completed successfully. NetView DM for MVS replaces the variety of different condition codes from the node with this single code.

This means that for VSE nodes, NetView DM for MVS stops executing a sequence whenever the return code is not zero.

**BYTES:** The number of bytes transmitted. When the transmission of a large resource is in progress, you can press PF6/18 to update this field. In a multi-tiered network with intermediate nodes, NetView DM for MVS can distribute resources using the fanout capability. During fanout with LU 0 nodes, this field shows the number of bytes actually transmitted to the first node, and zero for the other nodes in the group.

**RESOURCE NAME:** The name of the resource in the NetView DM for MVS resource repository. Program temporary fixes (PTFs), catalogs and file control (FCTL) names may consist of more than one data object. IOF shows you the name of each data object, rather than the name of the complete resource. For example, if the plan requests a SEND FCTL function, and the FCTL contains five data sets, you will see a separate line with the name and status information for each of the five data sets.
Using the PF Keys
The following PF keys are particularly useful when displaying the status of all functions of a phase:

PF6/18 Refreshes the display with current data, if any is available.
PF7/19 Scrolls a page backward in the list.
PF8/20 Scroll a page forward in the list.
PF10/22 Use this key after PF11/23 to move to the left of the screen and return to the original view.
PF11/23 Allows you to view data to the right of the screen.

I – Display a List of Conditioning Phases for Node
The term phase for node is used because the phase conditioning can be at node level. Refer to the CONDMODE operand of the DEFINE PHASE control statement of the SUBMIT batch utility. This is described in “DEFINE PHASE” on page 321.

You enter the I command for a phase for node if you want a list of all phases for nodes that the TCP must process before it can start the selected phase for node. As a result of the I command, the Display and Control Conditioning Phases panel is displayed.

The top of the panel shows the line commands you can enter. These are the same as those discussed previously in “Changing the Transmission Schedule” on page 503.
E – Display a List of Conditioned Phases for Nodes

You enter the E command for a phase for node if you want a list of all phases for nodes whose start depends on the completion of the selected phase for node. The E command displays the Display and Control Conditioned Phases panel. This panel is very similar to the Display and Control Conditioning Phases panel shown in Figure 130 on page 513. The instruction line is slightly different, and reads:

Enter command (CMD) for phases conditioned by: (Phase name)

The top of the panel shows the line commands you can enter. These are the same as those discussed previously in “Changing the Transmission Schedule” on page 503.
Chapter 22. Controlling Transmission with the Nodes

This chapter explains how you can hold and release transmissions to a node, or group of nodes, using Options 3 and 4 on the Master Menu.

The chapter also tells you how to hold and release a specific phase, and what to do when the system holds transmission.

Refer to the Installation and Customization book for information about the NDMTCP and NDMTCP macros.

<table>
<thead>
<tr>
<th>TRANSMISSION CONTROL MASTER MENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 HOLD TRANSMISSION FOR A NODE/GROUP</td>
</tr>
<tr>
<td>4 RELEASE TRANSMISSION FOR A NODE/GROUP</td>
</tr>
</tbody>
</table>

Option 3 – Hold Transmission for a Node/Group

You can hold all transmissions for a node, or group of nodes, using either Option 3, HOLD TRANSMISSION FOR A NODE/GROUP from the IOF menu, or using MVS MODIFY commands from the system console. NetView DM for MVS will hold transmissions to the node or group you specify at the end of the current transmission.

To start transmission again with a held node, you must release the node using either Option 4, RELEASE TRANSMISSION FOR A NODE/GROUP, from IOF or MVS MODIFY commands from the system console. NetView DM for MVS releases the node or group when the TCP ends and performs a cold start. If the TCP restarts with a warm start, it does not automatically release the nodes. You must release them with Option 4 or an MVS MODIFY command.

You can select Option 3 only if NetView DM for MVS has the following status:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Hold Transmission for a Node/Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNNING, WAITING, QUIESCED</td>
<td>The transmission requests for the specified node or group of nodes are held.</td>
</tr>
</tbody>
</table>

If you are holding transmission for transmission requests that specify a group of LU 6.2 nodes that share the same next intermediate node, the following limitations apply:
Option 3 – Hold Transmission for a Node/Group

To hold transmission for a node or group of nodes

Enter 3,(node/group name),(options) in the Command field.

**Node/Group name:** You must specify a node name, or the name of a group of nodes.

**Options:** There are two options. The first specifies whether you are holding transmission for a phase belonging to a single node, or to a group of nodes. The second specifies whether you want to hold transmission immediately, or after the currently executing phases have finished. You can use the two options together.

- **G** Specify this option if the hold command is for the named group of nodes. For each of the nodes, transmission will be interrupted at the end of the phase that is currently executing.

- **I** Specify this option if you want to hold either a single node, or a group of nodes immediately. For each of the nodes, transmission will be interrupted as soon as the next block of data is read or written from or to the resource repository. For functions whose status is pending, the Immediate (I) Hold command becomes effective when the function is completed.

The phases, NSRs, or D&CC API requests addressing the node(s) affected by this hold command and whose status is different from EXECUTING, change to HDEST status. The node or group is held for one invocation of the TCP. The node or group is automatically released when the TCP ends and restarts with a cold start.

<table>
<thead>
<tr>
<th>Phase or D&amp;CC API Request Status</th>
<th>Effect of Hold Node/Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The phase, or D&amp;CC API request is held with a status of HDEST, but only after the currently executing function completes (just like phases for nodes that do not share next intermediate nodes with other nodes addressed by the transmission request). This happens even if you specify an Immediate Hold, unless you issue the command against each node that shares the same next intermediate node.</td>
</tr>
<tr>
<td>EXEC and then HRETR, HSYST</td>
<td>The phase, or D&amp;CC API request, is HELD with a status of HDEST, but only after transmission is restarted and the currently EXECUTING function completes, or after you issue a Hold Immediate command against each node that shares the same next intermediate node.</td>
</tr>
<tr>
<td>READY</td>
<td>The phase, or D&amp;CC API request is held and the status is changed to HDEST.</td>
</tr>
<tr>
<td>PENDING</td>
<td>The phase, or D&amp;CC API request is held and the status is changed to HDEST only when the currently EXECUTING function completes.</td>
</tr>
</tbody>
</table>
Option 3 – Hold Transmission for a Node/Group

Holding a Single Node
To hold a single node:
- Enter 3, Name of a single node

The phase, NSR, or D&CC API request that is executing for this node is allowed to terminate and then the node is held. No transmissions can start with this node, until you release it.

Holding a Single Node Immediately
To hold a single node immediately:
- Enter 3, Name of single node, I

The phase, NSR, or D&CC API request that is executing for this node is interrupted immediately and the node is held. No transmissions can start with this node, until you release it.

Note: This does not apply in all cases. You cannot interrupt a phase, NSR, or D&CC API request that applies to an INITIATE function for a CLIST resource type.

Holding a Group of Nodes
To hold a group of nodes:
- Enter 3, Name of node group, G

The phases, NSRs, or D&CC API requests that are executing for the nodes of this group are allowed to terminate and then the nodes are held. No transmissions can start with the nodes in this group until you release the group.

Holding a Group of Nodes Immediately
To hold a group of nodes immediately:
- Enter 3, Name of node group, G, I
  or
  3, Name of node group, I, G

The phases, NSRs, or D&CC API requests that are executing for the nodes of this group are interrupted and the nodes are held. No transmissions can start with the nodes in this group until you release the group.

Note: This does not apply in all cases. You cannot interrupt a phase, NSR, or D&CC API request that applies to an INITIATE function for a CLIST resource type.

What Could Go Wrong
NetView DM for MVS issues a message if:
- You did not enter a node or group name, or you entered an invalid node or group name.
- The node or group is not defined in the DRD.
- You entered a blank before or after the comma.
- The NetView DM for MVS system status is ENDING or QUIESCING.
- You enter a Hold command for a VSE node or group.
Option 4 – Release Transmission for a Node/Group

- You hold an LU 0 node belonging to a group, or a group itself, for which fanout is taking place. The transmission of the distribution headers to the front end S/36 may be interrupted. A distribution header is a message that the TCP sends to the front-end node to request it to distribute data to an end node. When a resource is being sent to a group of LU 0 nodes using fanout, the resource is sent to the front-end node headed by a distribution header. This header addresses the first end node for which the transmission activity was scheduled. For the other end nodes, the TCP only sends distribution headers (one for each node).

If you interrupt this transmission of distribution headers, all the phases for nodes for which a distribution header was already sent to the front end S/36 might remain in pending status until the held node or group is released. This is because the front end will not begin sending to the end nodes until it receives the last distribution header.

**Note:** This does not apply to fanout for LU 6.2 node types.

Holding a Specific Phase for a Node

When you are working with Option 2, DISPLAY AND CONTROL TRANSMISSION SCHEDULES, you can hold a specific phase for node, or D&CC API request. "H – Hold Transmission of a Phase or D&CC API Request" on page 507 explains how you can do this.

What To Do when Transmission is Held by the System

NetView DM for MVS may automatically hold transmission if there is a line problem. In this case, NetView DM for MVS holds all transmission with the nodes that belong to the logical unit (LU) on which the problem occurred.

In this case, all transmission requests that address nodes connected through the affected LU are set to HRETR or HSYST status.

Option 4 – Release Transmission for a Node/Group

Option 4, RELEASE TRANSMISSION FOR A NODE/GROUP releases all transmission requests which you, or the system, held for a specified node or group of nodes. You can also use MVS MODIFY commands to release transmission for a node or node group.

Nodes or groups can be held for any of the following reasons:

- You issued a hold command, using Option 3, HOLD TRANSMISSION FOR A NODE/GROUP.
- During a previous transmission, the TCP encountered a LOSTTERM condition (which means “LOST TERMINAL”). This condition occurs when the TCP loses contact with a node.
- The subtask that was carrying out the transmission abended.

Option 4 is valid only if NetView DM for MVS has the following status:
Option 4 – Release Transmission for a Node/Group

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Release Transmission for a Node/Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNNING, WAITING</td>
<td>Any phases for the released node whose scheduled time has passed are initiated.</td>
</tr>
</tbody>
</table>

To release a node or group using IOF

Enter 4 in the command field, followed by the node or group name, followed by the option. For example:

```
4, name of group, G
```

The status of the phases, NSRs, or D&CC API requests addressing the node, or nodes, is changed from HDEST, HRETR, or HSYST, to READY or WAITING, depending on the scheduled date and time.

Name of node/group: You must specify a node name, or the name of a group of nodes.

Options: You must specify the option as G to release transmission requests for a named group of nodes.

To release transmission for a single node, do not specify an option, only the node or group name.

What Could Go Wrong

NetView DM for MVS issues a message if:

- You did not enter a node or group name, or you entered an invalid node or group name.
- You entered Option 4 for a node or group that is not currently in the held state.
- The node or group name is not defined in the resource directory.
- You entered a blank before or after the comma.
- The NetView DM for MVS system status is ENDING or QUIESCING. NetView DM for MVS ignores the release request.

Releasing a Phase or D&CC API Request Held Explicitly

If you held specific phases, or D&CC API requests, using the Hold Transmission of a Phase option, releasing the node does not also release the phases or D&CC API requests. You release phases or D&CC API requests using the R (Release) command, described in “R – Release a Held Phase or D&CC API Request” on page 508.

Releasing Transmission Held by the TCP

If transmission is interrupted because of a line problem, or other communication problem, the TCP holds all of the nodes involved, and any type of transmission request belonging to those nodes. The TCP puts these requests into a HRETR or HSYST status. If you also held these nodes with a Hold command, the TCP puts the transmission requests in an HDEST status.

If the status is HRETR, the TCP tries to reestablish transmission periodically. The retries depend on the RETINT and RETRY parameters specified in the NDMTCP
Option 4 – Release Transmission for a Node/Group

and NDMTP macros at installation time. The values that are specified in NDMTCP are used for nodes connected to the host through logical units that are assigned to default transmission profiles. The values that are specified in NDMTP are used for nodes connected through logical units that are assigned to transmission profiles that the user defines.

The RETINT parameter specifies the time interval between attempts to restart transmission. The RETRY parameter specifies the maximum number of allowable retries.

If there is a line problem at SNA initialization, transmission is not possible. The TCP will not hold the nodes and the transmission requests belonging to the nodes until the TCP has reached the maximum number of retries. In this case, the number of retries depends on other parameters, defined in the NDMTP macro at installation time.

SWDLY and IDLY specify the number of seconds between each attempt to restart transmission. SWRTRY and IRTRY specify the maximum number of allowed retries.

The SWDLY and SWRTRY parameters are used for logical units belonging to the default transmission profiles. The IDLY and IRTRY parameters are used for logical units belonging to user defined transmission profiles.

If all the attempts to restart transmissions are unsuccessful, the transmission requests are HELD. You can release the held node by using Option 4 on the Master Menu. By specifying Option 4 for one node belonging to the failing logical unit, you will release transmission for all nodes belonging to that logical unit.

If line problems occur with a logical unit related to a phase that has transmission windows defined, the phases are held and the status becomes HRETR between the retries.

If the phase is in HRETR status and a window is due to start, the phase is not released. The TCP puts the phase in the READY status when the time interval defined in the RETINT parameter expires.

If the phase is in HSYST status, you can reset the phase status only by resetting the plan or releasing the node. This applies to phases with or without transmission windows defined. Resetting plans is described in “Resetting Plans” on page 169 using GIX, and “RESET PLAN” on page 418 using the TCFMAINT batch utility. Releasing the node is described in “Option 4 – Release Transmission for a Node/Group” on page 518.

For phases without windows, the phase status becomes READY when you enter the command to either reset the plan or release the node.

For phases with windows, the TCP handles the phase as follows when you enter the command to either reset the plan or release the node:

- If the command is issued within the time window, the phase status becomes READY
- If the command is issued outside the time window, the phase status becomes HWIN.
Chapter 23. Changing Transmission Profiles

This chapter shows you how to change the attributes of transmission profiles using the IOF. It also shows you how to change the number of SNA sessions that you can run concurrently, using the CURTASK parameter.

You can use Option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES, for both of these tasks. When you select Option 11 from the Master Menu, you specify a parameter to specify one of these tasks.

You can also use an MVS MODIFY command to change the CURTASK parameter.

The following figure shows the panel flow for Option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES.

```
TRANSMISSION CONTROL MASTER MENU

DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES

DISPLAY AND CONTROL TRANSMISSION PROFILES
Select a transmission profile

CHANGING THE CURTASK VALUES
```

*Figure 131. Option 11 - DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES*
Using Transmission Profiles

Transmission profiles provide an efficient way of using transmission lines. A transmission profile is a way of grouping together nodes which have the same transmission characteristics, referred to in this book as *transmission profile attributes*. Each node in a NetView DM for MVS network must belong to a transmission profile (but not to more than one).

The person who installs NetView DM for MVS defines the attributes of transmission profiles using the NDMTP macro. Now you can change these attributes using IOF option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES. The transmission profile attributes you can change are:

**Type of Transmission Line (LINETYPE)**

This is the most important transmission profile attribute, defining the type of transmission line used between the host and the logical units (LUs) associated with the target nodes. The target nodes are the nodes to which you are transmitting. The lines can either be *leased* or *switched*. This is defined in the LINETYPE parameter of the node. So, you could define a separate transmission profile for nodes connected to NetView DM for MVS in each of the following different ways:

- Through *dial-in* lines (switched)
- Through *dial-out* lines (switched)
- Through the same *multipoint* line
- Through *point-to-point leased* lines.

**Minimum Number of Transmission Tasks (MINGR)**

The transmission profile allows the concurrent execution of at least this many transmission tasks (starting transmission requests, or accepting NSRs).

**Maximum Number of Transmission Tasks (MAXN)**

The maximum number of transmission tasks that can be active at any time for this transmission profile. This value must be greater than or equal to the MINGR value.

**Retry Interval (TPRTINT)**

This is the time interval in minutes after a line failure, before NetView DM for MVS tries to re-establish a connection with the logical unit again.

**Number of Retries (TPRRETRY)**

This specifies the number of times a transmission retry is attempted, in the case of a line failure, before considering the failure as permanent.

**Immediate Retry Interval (IDLY)**

This is the time interval in seconds that a TCP subtask waits before re-attempting to establish an SNA session that failed. The session can fail when the logical unit or a transmission link is temporarily unavailable.

**Number of Immediate Retries (IRTRY)**

This is the number of times that a TCP subtask re-attempts to establish an SNA session that failed due to a logical unit or transmission task being temporarily unavailable.

Besides the transmission profiles you define, NetView DM for MVS always creates two default transmission profiles at installation time. One is called NDMTPL, and is for LUs connected to the host through leased lines. The other is called NDMTPS,
Using Transmission Profiles

and is for LUs connected to the host through switched lines. When a user defines a node, using either GIX or the Batch Utilities, NetView DM for MVS assigns the LU to one of these default transmission profiles.

You can now change the attributes of these default profiles, apart from the type of transmission line. The assignment of logical units to transmission profiles can also be changed from GIX (see “Changing Logical Unit to Transmission Profile Assignments” on page 47), or Batch Utilities (see “CHANGE LUTP” on page 235).

The user can specify the PRIORITY parameter for a plan that contains units of work that address logical units belonging to different transmission profiles. During the scheduling for the unit of work that is ready to be processed, the PRIORITY parameter is always maintained until the MAXN value has been reached, even if MINGR has not yet been reached for the other transmission profiles. Two units of work with the same PRIORITY parameter, which address logical units that belong to different transmission profiles, are scheduled according to the MINGR specified. The maximum (MAXN) number of active sessions are always guaranteed regardless of the priority.

For example if you have the following:

- The LU1, LU2, LU3 logical units that belong to the NDMTP1 transmission profile with MINGR=2 and MAXN=4
- The LU4, LU5 logical units that belong to the NDMTP2 transmission profile with MINGR=3 and MAXN=4

the sessions start as follows:

- PLAN1PHASE1LU1 PRIORITY 1
- PLAN2PHASE1LU4 PRIORITY 1
- PLAN3PHASE1LU2 PRIORITY 2
- PLAN4PHASE1LU3 PRIORITY 2
- PLAN5PHASE1LU5 PRIORITY 4

The session PRIORITY 2 PLAN4PHASE1LU3 starts because the MAXN for the NDMTP1 transmission profile has not been reached and the priority is higher than the following unit of work, even though the MINGR has not yet been reached for the previous one.
Option 11 – Display/Change CURTASK or Transmission Profiles

You can use this menu option to perform two different tasks:

- Change the current number of SNA sessions that can be active at the same time. The maximum value is defined in the MAXTASK parameter of the NDMTCP macro at NetView DM for MVS installation time. Refer to “Changing the CURTASK Values.”

- Change the attributes of a transmission profile, such as the minimum and maximum numbers of transmission tasks that the TCP tries to run concurrently. Refer to “Changing Transmission Profiles” on page 526.

Changing the CURTASK Values

You can change the current values for the number of concurrently active SNA sessions while the TCP is running. The MAXTASK parameter of the NDMTCP generation macro originally defines the greatest possible values, but you can change the actual values while the TCP is running, using CURTASK. The person who installs NetView DM for MVS originally defines these parameters at installation time, or you can define them when you start the TCP, “Specifying the Maximum Number of SNA Sessions (MAXTASK=)” on page 480 and “Specifying the Current Number of SNA Sessions (CURTASK=)” on page 480 show you how to do this.

You can also change the CURTASK values while the TCP is running. The two values are:

- The total number of sessions
- The number of sessions on switched lines.

Changing the current values has an effect on the scheduling activities of the TCP. Starting from the time that the TCP accepts the new values, the TCP tries to maintain the number of active sessions within the new limits. This may not be immediate, for example, the number of active sessions at the time that NetView DM for MVS accepts the request may exceed the new limits.

There are two ways to access the Change CURTASK panel. You can either access it directly from the Master Menu, or from the Display and Control Transmission Profiles panel. The example below shows you how get there directly from the Master Menu. To display this panel from the Display and Control Transmission Profiles panel, you simply press PF2.

To change the CURTASK value, follow the instructions below after you have logged on to IOF and started transmission:

1. Select Option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES, from the Master Menu and enter the parameter M to specify the CURTASK option.

The Change CURTASK panel is displayed.
Enter the new values of CURTASK (if a field is not filled, the related value is not changed).

<table>
<thead>
<tr>
<th>New values</th>
<th>Current values</th>
<th>Maximum allowed</th>
<th>Active sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>All the lines</td>
<td>100</td>
<td>200</td>
<td>73</td>
</tr>
<tr>
<td>Switched lines</td>
<td>80</td>
<td>150</td>
<td>68</td>
</tr>
</tbody>
</table>

If neither input value is specified, a refresh is performed.

PF= __ 1/13=HELP 3/15=END 4/16=RETURN 5/17=MSG-UNS 9/21=PRINT

M= 127 STATUS=RUNNING TRANSM= 73 PRINTER=UNAVAIL

Figure 132. Example of the CHANGE CURTASK Panel

This panel shows, both for the total number of sessions, and the number of sessions on switched lines:

- The current values for the maximum number of sessions
- The maximum allowed values
- The number of currently active sessions.

2. Enter the new value in the New values field, either for all the lines, or switched lines only, and press Enter.

The panel is refreshed and the new values are displayed.

If you enter invalid values, or you enter a value that exceeds the maximum allowed value, the panel displays an error message.

You can refresh the data displayed on the panel by pressing Enter without entering any values in the fields. This will give you the most up to date picture of the number of sessions that are currently active.

**Example**

This scenario assumes the following conditions:

- When you started the TCP, you specified MAXTASK=(30,20) and CURTASK=(20,15).
- 15 sessions are currently active, 12 of these are on switched lines.
- You change the CURTASK values to (20,10).
- A user submits a plan with two phases:
  - PHASE1 addresses a node connected through a leased line
  - PHASE2 addresses a node connected through a switched line.

Suppose that the two phases end at the same time. The first phase is for a node connected through leased lines. The second is for a node connected through...
Option 11 – Display/Change CURTASK or Transmission Profiles

Switched lines. The TCP would then schedule the newly submitted plan in the following way, given the above conditions:

- If PHASE1 addresses the same logical unit as the phase that just ended on a leased line, then the TCP executes PHASE1 on the same session.
- If PHASE1 addresses another logical unit already engaged in a session, then the TCP does not execute it until a phase using this session ends.
- If PHASE1 addresses another logical unit not yet engaged in a session, the TCP deactivates the session used by the phase that ended. The TCP activates a session with the new logical unit and executes PHASE1.
- The TCP deactivates the session used by the phase that ended on a switched line, but does not execute PHASE2. It does not matter whether or not it addresses the same logical unit as the one that just ended, or another one. The TCP will only execute this phase when the number of active sessions on switched lines becomes less than 10.

Changing Transmission Profiles

Follow the instructions below after you have logged on to IOF and started transmission:

1. Select Option 11, DISPLAY/CHANGE CURTASK OR TRANSMISSION PROFILES, from the Master Menu and enter the parameter T to specify the transmission profiles option.

   The Display and Control Transmission Profiles panel is displayed.

   ![NDMOC070 DISPLAY AND CONTROL TRANSMISSION PROFILES PAGE 1 OF 1](image)

   Current Maxtask is: 18 14
   Line commands: C(Change)

   Enter command (CMD) for the selected TP:

<table>
<thead>
<tr>
<th>CMD</th>
<th>TP NAME</th>
<th>TYPE</th>
<th>GRANTED</th>
<th>ALLOWED</th>
<th>ALLOCATED</th>
<th>RETRY</th>
<th>RETINT</th>
<th>IRTRY</th>
<th>IDLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>APPL1</td>
<td>S</td>
<td>5</td>
<td>10</td>
<td>7</td>
<td>3</td>
<td>15</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>APPL3A</td>
<td>S</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>NDMTP007</td>
<td>L</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>NDMTP003</td>
<td>L</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>-</td>
<td>NDMTPL</td>
<td>L</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>-</td>
<td>NDMTPS</td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>30</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

   Press ENTER to highlight the first line with a command.
   Press ENTER again to process highlighted command and highlight next command.

   **PF=** 1/13=HELP 2/14=MAXT 3/15=END 4/16=RETURN 5/17=MSG-UN
   6/18=REFRESH 7/19=BACK 8/20=FORW 9/21=PRINT
   **M=** 13 STATUS= RUNNING TRANS= 18 PRINTER= AVAIL

   ![Figure 133. Example of the DISPLAY AND CONTROL TRANSMISSION PROFILES Panel](image)

The first value shown at the top of this panel is for the total current number of sessions that are allowed to run concurrently. The second value is for the number of sessions on switched lines that are allowed to run concurrently.
Option 11 – Display/Change CURTASK or Transmission Profiles

The panel shows one line of information for each transmission profile, listing the transmission profiles in the order in which the user defined them.

2. Enter the C (Change) line command in the Command field next to the transmission profile that you want to change.

   **Note:** You cannot change the transmission profile name or the number of allocated tasks for a transmission profile. Also, for the default transmission profiles, you cannot change the transmission profile line type.

3. Press Enter when you have finished entering commands, and the line containing the first command is highlighted.

4. Move the cursor to the fields you want to change, and type over the values that are currently displayed.

5. When you have finished, press Enter again to process the command for the highlighted line. NetView DM for MVS changes the value, and highlights the next line with a command.

6. Press PF4 when you have finished to return to the Main Menu, or, press PF2 to display the Change CURTASK panel.

The fields on the Display and Control Transmission Profiles panel are described below.

**TP Name:** This field shows the name of the transmission profile, as defined at installation time, or the default transmission profiles called NDMTPL and NDMTPS.

**TYPE:** Shows the type of lines connecting the host to the logical units (LUs) of this transmission profile. The lines can either be Switched or Leased.

**GRANTED:** This shows the minimum granted number of transmission tasks that the TCP tries to run concurrently for a transmission profile. During normal transmission operations, a transmission profile might not use all of its granted tasks. In this case, other more active transmission profiles can automatically acquire some of these idle tasks. The number of extra tasks must not exceed either the maximum allowed for the specific transmission profile, or the CURTASK value for the total number of concurrent tasks.

**ALLOWED:** This shows the maximum number of transmission tasks that can run concurrently for a transmission profile.

**ALLOCATED:** This shows the number of tasks currently allocated to the transmission profile.

**RETRY:** This shows the number of times that NetView DM for MVS will try to reestablish a connection after a line failure.

**RETINT:** This shows the retry interval after a line failure. This is the amount of time, in minutes, before NetView DM for MVS tries to reestablish a connection with a logical unit after a line failure.

**IRTRY:** This shows the number of immediate retries. This is the number of times that a TCP subtask will try to reestablish an SNA session that failed. SNA sessions fail when the logical unit, or the transmission link, is temporarily unavailable.
**IDLY:** This shows the immediate retry interval. This is the amount of time, in seconds, that a TCP subtask will wait before trying to reestablish an SNA session that failed. SNA sessions fail when the logical unit, or the transmission link, is temporarily unavailable.

A change in the number of GRANTED or ALLOWED tasks does not affect the currently ongoing transmissions for logical units that belong to the transmission profile being changed. The change affects only the scheduling of new transmissions.

Therefore, after the following sequence of events, transmissions that were *in progress* when the Immediate End command was issued will be rescheduled, regardless of the initial values of the number of GRANTED and ALLOWED tasks:

1. The number of GRANTED or ALLOWED tasks is changed, for one or more transmission profiles, with respect to the initial values as defined with the NDMTP customization macro.
2. The TCP is running transmissions according to the changed values.
3. The TCP is stopped, by an Immediate End command.
4. The TCP is restarted with START=WARM and RESTTP=YES (that is, the *initial values*, rather than the changed values, have to be used).

The initial values will become effective for new transmission scheduling.
Chapter 24. Stopping Transmission and the Transmission Environment

This chapter shows you how to end transmission when you are running in attended mode under operator control, using IOF. The chapter then explains how to end transmission if you are running in unattended mode. You can do this either from NCCF/TAF (the IBM Network Communication Control Facility/ Terminal Access Facility), or from the NetView console using MVS MODIFY commands.

If you want to end transmission while there are still phases ready to run, you can:

- Stop transmission activity without stopping the TCP
- Stop the TCP
- Stop the IOF

To stop transmission, the TCP, or the IOF, you can choose from Options 5 through 10 on the Master Menu.

### TRANSMISSION CONTROL MASTER MENU

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>QUIESCE TRANSMISSION</td>
<td>Stopping Transmission</td>
</tr>
<tr>
<td>6</td>
<td>IMMEDIATE QUIESCE TRANSMISSION</td>
<td>Stopping the TCP</td>
</tr>
<tr>
<td>7</td>
<td>END TRANSMISSION</td>
<td>Stopping the TCP</td>
</tr>
<tr>
<td>8</td>
<td>IMMEDIATE END TRANSMISSION</td>
<td>Stopping the IOF</td>
</tr>
<tr>
<td>9</td>
<td>DISCONNECT IOF</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>LOGOFF FROM IOF</td>
<td></td>
</tr>
</tbody>
</table>

### Stopping Transmission

There are two ways in which you can stop transmission:

- You can stop the scheduling of transmission requests after the current resource has been transmitted, using Option 5, QUIESCE TRANSMISSION.
- You can stop the scheduling of transmission requests immediately, using Option 6, IMMEDIATE QUIESCE TRANSMISSION.

### Option 5 – Quiesce Transmission

Use Option 5, QUIESCE TRANSMISSION, to stop the TCP from scheduling any more transmission requests, after the current resource has been transmitted.

You can select Option 5 if the NetView DM for MVS system status is:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Quiesce Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAITING, RUNNING</td>
<td>The status is changed to QUIESCING while the transmission requests end. The status is changed to QUIESCED when all transmission activity has ceased.</td>
</tr>
</tbody>
</table>
Stopping the TCP

To resume transmission operations, use Option 1, START TRANSMISSION. The TCP immediately schedules interrupted transmission requests to run again, unless you changed the scheduled date and time of any requests. If you did, the requests are initiated in the newly scheduled sequence. Chapter 21, “Controlling Transmission Schedules” on page 495 shows how to change the schedule.

What Could Go Wrong
If the status is ENDING, QUIESCING, or QUIESCED, NetView DM for MVS ignores the request.

Option 6 – Immediate Quiesce Transmission
Use Option 6, IMMEDIATE QUIESCE TRANSMISSION to stop transmission activities in a way similar to Option 5, QUIESCE TRANSMISSION. With Option 6, however, the currently active transmission requests are ended immediately at the next checkpoint, rather than at the end of the current resource.

You can select Option 6 if the NetView DM for MVS system status is:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Immediate Quiesce Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAITING, RUNNING</td>
<td>The status is changed to QUIESCING.</td>
</tr>
<tr>
<td>QUIESCING</td>
<td>The status is changed to QUIESCED.</td>
</tr>
</tbody>
</table>

Stopping the TCP
Options 7 and 8 on the Main Menu stop both the TCP and the IOF, either after all active transmission requests have been processed, or immediately.

Option 7 – End Transmission
Option 7, END TRANSMISSION, ends transmission activity as soon as all active transmission requests have been processed, terminates the TCP, and logs you off from IOF. The TCP does not initiate any more transmission requests, and rejects requests from nodes to release transmission at the end of the SNA session. Currently active transmission requests end normally. This logs you off and terminates the IOF and TCP.

You can select Option 7 if the NetView DM for MVS system status is one of the following:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of End Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNNING</td>
<td>Currently active transmission requests will end normally. Meanwhile, the system status is changed to ENDING. If no more transmission requests are active, you are logged off and the TCP and the IOF are ended.</td>
</tr>
<tr>
<td>WAITING, QUIESCED</td>
<td>The status is changed to ENDING. You are logged off and the TCP and the IOF are ended.</td>
</tr>
</tbody>
</table>

To restart transmission activity, resubmit the JCL shown in Figure 120 on page 473.
What Could Go Wrong
If the status is QUIESCING, NetView DM for MVS rejects the request and issues a message. If the status is ENDING, NetView DM for MVS ignores the request.

Option 8 – Immediate End Transmission
Option 8, IMMEDIATE END TRANSMISSION, ends all transmission activity immediately, terminates the TCP, and logs you off from IOF. Option 8 is the same as Option 7, END TRANSMISSION, except that currently active transmission requests are ended at the next checkpoint.

You can select Option 8 if the NetView DM for MVS system status is one of the following:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Effect of Immediate End Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUNNING</td>
<td>Currently active transmission requests are ended at the next checkpoint, you are logged off, and the TCP and the IOF are ended.</td>
</tr>
<tr>
<td>WAITING, QUIESCED</td>
<td>The status is changed to ENDING. You are logged off, and the TCP and the IOF are ended.</td>
</tr>
<tr>
<td>ENDING</td>
<td>The mode of ending is changed to IMMEDIATE END. Active transmission requests are ended after the next checkpoint, you are logged off, and the TCP and the IOF are ended.</td>
</tr>
</tbody>
</table>

When you run the TCP again, the TCP restarts the interrupted transmission requests immediately.

What Could Go Wrong
If the status is ENDING because of a previous Immediate End Transmission request, NetView DM for MVS ignores the new request.
Stopping the IOF

Option 9 disconnects IOF. Option 10 logs you off from IOF. You select Option 9, DISCONNECT IOF, if you intend to return to IOF at a later time. If you do not expect to return to IOF, select Option 10, LOG OFF FROM IOF.

Option 9 – Disconnect IOF

Option 9 disconnects the IOF, or you can press the PF key specified by the D= parameter of the BGNSESS command. While IOF is disconnected, you can work with other applications from the same terminal, without having to switch to unattended operation. NetView DM for MVS queues messages until you log on again. Transmission activity continues as if you were still logged on to IOF. Activity stops when all transmission requests have completed. Other operators can still log on to IOF and monitor transmission, even after transmission activity has stopped. To return to the IOF, use the RTRNSESS command. To return to the NCCF, use the TAF.

You can select Option 9 when NetView DM for MVS is in any status. At the next IOF logon, the system status will reflect the changes resulting from the transmission activities which continued while you were disconnected.

Option 10 – Log Off from IOF

To log off from the IOF, you can use Option 10 on the Master Menu, or you can use the ENDSESS FLCSN,APPLID=iapplid command from NCCF. Once you have logged off, the TCP operates in unattended mode. While you are logged off, another operator may log on. This operator will have the same IOF control as you had before.

To disconnect from the IOF, rather than log off completely, refer to “Option 9 – Disconnect IOF.”

The logoff request is only valid if NetView DM for MVS has the following status:

<table>
<thead>
<tr>
<th>SYSTEM STATUS</th>
<th>EFFECT OF LOG OFF FROM IOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDING</td>
<td>The TCP ends when all phases are complete.</td>
</tr>
<tr>
<td>RUNNING</td>
<td>TCP continues to run until all phases have ended. The status is changed to WAITING.</td>
</tr>
<tr>
<td>WAITING</td>
<td>TCP continues to wait. It will resume transmission if there is more work to do. To stop the TCP, issue the MVS Modify (F) command from the system console, or log on to IOF and select Option 7, END TRANSMISSION.</td>
</tr>
</tbody>
</table>

What Could Go Wrong

If the status is ENDING or QUIESCED, NetView DM for MVS ignores the request.
Chapter 25. Handling Unsolicited Messages

This chapter describes unsolicited messages that you receive when you are working with IOF. You can display these messages from any IOF panel by pressing PF5 or PF17. The chapter then explains how to view the messages displayed on your panel.

Types of Messages

When you work with IOF, you receive messages that are related to the dialog you have with IOF. For example, a message may tell you that the IOF has accepted a request you entered, or that you made an error when entering the request. NetView DM for MVS also issues messages that are not related to the dialog. These messages are displayed on the same panel from which you have entered the request, and are called unsolicited messages. They tell you about:

- Problems establishing a connection with a remote node. For example, a message may tell you that the TCP cannot establish connection with a node because the node is inactive at the moment.
- Events concerning the progress of transmissions. For example, a message may tell you that a transmission function for a given node is completed.

Routing of Messages

NetView DM for MVS always displays messages related to the dialog on the panel.

NetView DM for MVS writes unsolicited messages to the data set referred to by SYSPRINT (see Figure 120 on page 473). When the TCP ends, the SYSPRINT file is printed.

If you choose to run the TCP in attended mode (OPCTL=IOF or OPCTL=NETV), unsolicited messages are also put in the unsolicited message file, (referred to by the DD statement DSXUN01; see Figure 120 on page 473), from which you can display them on the terminal. All the unsolicited messages will be queued to the unsolicited message file if you have specified MSGINFO=2 in the NDMTCP customization macro. Information and warning messages will not be queued if you have specified MSGINFO=0.

You may also want to send selected unsolicited messages, including messages related to MVS MODIFY commands issued from the system console, to the NetView console.

To have selected unsolicited messages sent to the NetView console, select the option OPCTL=NETV when starting up the TCP. The selection of messages to be sent to the NetView console is based on the contents of a user-defined table, called NDMNETTB, that lists the IDs of the messages to be sent. When the TCP issues a message, it scans this user-defined table and, if the message ID matches one in the table, sends the message to the NetView console.

For more information on routing messages to the NetView console, refer to the Installation and Customization book.
Viewing Unsolicited Messages

You display the unsolicited messages that are waiting to be viewed by pressing PF5/17 from any IOF panel. Every panel includes information about the unsolicited messages waiting to be viewed. Figure 134 shows how this information appears in the M= field on the IOF panels.

### Figure 134. Displaying the Number of Messages to be Viewed

**M=** The number of unsolicited messages not yet viewed. Press PF5/17 to display the Handle Unsolicited Messages panel. This panel shows you all the messages that you have not viewed previously.

### Figure 135. Example of the HANDLE UNSOLICITED MESSAGES Panel

The fields shown on this panel are as follows.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SEQN</strong></td>
<td>The sequence number you use to respond to a message. For any number above 9999, the panel only shows the last four digits. You use the complete number in order to respond to a message.</td>
</tr>
<tr>
<td><strong>TIME</strong></td>
<td>The time when NetView DM for MVS issued the message.</td>
</tr>
<tr>
<td><strong>MSG NO.</strong></td>
<td>The number of the message. You can use this number to find the message and its explanation in the Messages and Codes book.</td>
</tr>
<tr>
<td><strong>MESSAGE TEXT</strong></td>
<td>The text of the message. If a message is lost, the words MESSAGE LOST appear.</td>
</tr>
</tbody>
</table>

NetView DM for MVS User's Guide
Locating Messages in the List

You can scroll backwards or forwards in the list using PF7/19 and PF8/20. You can:

- Locate a specific message by keying its sequence number in the Command field and pressing Enter.
- Locate the oldest message present in the unsolicited message file by pressing PF10/22.
- Locate the last message already viewed by pressing PF11/23.
- Locate the last written message by pressing PF12/24. In this case, the last written message becomes the last viewed message, as if you had already viewed all of the previous messages.

What Could Go Wrong

- If the message sequence number you enter is not valid, a message is displayed.
- If an unrecoverable error occurs while NetView DM for MVS is accessing the unsolicited message file, you will receive a message telling you that the file is no longer available. You can no longer use the unsolicited message facility for this TCP run. To regain control, you have to end the TCP, assign a different file as the unsolicited message file, and start a new TCP run with the new file. Unsolicited messages continue to be queued to SYSPRINT.
- If the unsolicited message file is full, or nearly full, messages that you have not viewed yet may be overwritten. The storage for the unsolicited message file is allocated when NetView DM for MVS is installed. The messages are stored in wrap-around mode.

If this happens, NetView DM for MVS takes the following actions:

- The M= field, showing the number of messages not yet viewed, is replaced by a highlighted MESSAGE LOST status to call this critical situation to your attention.
- TCP messages are no longer written into the unsolicited message file.
- When you press PF6/18 for messages that are lost, the words MESSAGE LOST are shown on the system line.

To recover from the MESSAGE LOST condition, scroll the display forward to the bottom of the display. This allows NetView DM for MVS to write as many new messages as the number of messages that you have viewed. The MESSAGE LOST entry disappears from the M= field, which will again show the number of messages waiting to be viewed.

Try to make a habit of reviewing messages promptly in order to avoid the MESSAGE LOST condition. If the MESSAGE LOST condition occurs anyway, consider making the unsolicited message file larger. Refer to “Maintaining the Unsolicited Message File” on page 457 in Part 2, “Using the Batch Utilities.”
Chapter 26. Handling Problems

This chapter describes problems that could occur while the transmission control program (TCP) is active. These problems can be related to:

- The environment under which NetView DM for MVS works
- The TCP and its files
- VTAM problems
- Problems with the service provider (SVP)

The Diagnosis book gives more information on diagnosing and solving these problems.

For problems related to specific node types, refer to the appropriate documentation for your node.

The TCP Log File

When the TCP ends, it produces a log of transmission operations in the SYSPRINT file. You can read this log file yourself, or save or print the file for your system programmer's further analysis. You can also write application programs to process the log file and extract the information in the format you require.

The TCP log file logs the following type of information:

- All the error conditions detected during transmission, including error messages related to the SVP.
- The start and end time for each node and each transmission function.
- The final function status.
- The number of bytes transmitted or received.
- All the messages with action code I or W that the TCP issues. This only applies if you specified MSGINFO=1 or 2 when you started the TCP, or when NetView DM for MVS was installed.
- The user identification of each operator who logs on or off IOF.
- All the IOF operator commands that change the status of the TCP, for example, the Start Transmission command.
- All the IOF operator commands that change the status of NetView DM for MVS phases for specific nodes, for example, the Hold Phase or Release Phase commands.

Figure 136 on page 538 shows an example of the contents of a TCP log.
Problems at the Host

Problems that occur at the host are divided into two types: those that cause TCP to shut down and those that do not.

Problems That Cause a TCP Shutdown

The TCP shuts down immediately when one of the following problems occurs:

- A VTAM or TCAM shutdown.
- One of the following NetView DM for MVS files that the TCP accesses fails:
  - Transmission control file (TCF)
  - Distributed resource directory (DRD)
  - Resource repository (library or holding file)
  - Request queue file (NDMRQF)
  - Communication work file (NDMCWK)

VSAM indicates these problems to the TCP. The Diagnosis book contains more information about these problems.

- The service provider has been immediately stopped.

Figure 136. Example of the TCP Log File
Problems at the Host

- A permanent TCP subtask abends (for example the IOF subtask, or the request queue processor). For information about the TCP subtasks refer to the Diagnosis book.
- A recovery subtask abends.

NetView DM for MVS issues a message identifying the problem, in each of the above cases.

The TCP cannot control a shutdown when one of the following cases occurs:
- The system operator enters a Cancel command
- The operating system shuts down
- The TCP main task abends.

What NetView DM for MVS Does
The TCP interrupts any transmission in progress. When the TCP performs an immediate end, it puts all RUNNING phases into a RESTARTABLE or PENDING status. Resources partially retrieved in the repository (flagged as INCOMPLETE) are not available for use. When the TCP cannot control the shutdown, RUNNING phases remain in the current status. The next time that you invoke the TCP, it will put all RESTARTABLE phases into a READY status. RUNNING phases become READY or PENDING, depending on their initial status.

What You Can Do
Once you have solved the problem, invoke the TCP again to restart transmission. Cold start the TCP by specifying START=COLD as a startup parameter. If you want to check the status of phases and functions before restarting transmission, start the TCP with the PARM options OPCTL=IOF and AUTOSTR=NO. You can then check the status by selecting Option 2, TRANSMISSION DISPLAY AND CONTROL, from the IOF Master Menu.

Save or print the TCP log file for your system programmer's further analysis. Refer to Figure 136 on page 538 for an example of a log produced by the TCP.

Problems That Do Not Cause a TCP Shutdown
The TCP does not shut down if a transmission subtask ends abnormally and the main task recovers from the abend successfully.

The TCP does not shut down if an error occurs in the unsolicited message file (DSXUN01) or in the work file for IOF displays (DSXWF01) when OPTCTL=IOF or OPTCTL=NETV.

What NetView DM for MVS Does
- When a subtask ends abnormally, NetView DM for MVS interrupts the transmission in progress for that particular node, and closes the SNA session. The interrupted function completes with a return code 8, and NetView DM for MVS flags the interrupted phase as RESTARTABLE. Resources partially retrieved in the repository (flagged as INCOMPLETE) are not available for use. NetView DM for MVS holds all of the nodes that belong to the involved logical unit. Periodically, the TCP attempts to restart transmission until it reaches the maximum number of retries. For each attempt to restart transmission, the TCP automatically releases the held nodes. The retries depend on two parameters defined during installation time. They are:
Line and Node Problems

- RETINT, which specifies the time interval between each attempt to restart transmission.
- RETRY, which specifies the maximum number of retries.

- If an error occurs in the unsolicited message file, NetView DM for MVS forces the operator control parameter (OPCTL) to NONE. The TCP then continues to run in unattended mode. If an error occurs in the work file for IOF displays, you will no longer be able to use the IOF display function. All other operations remain the same.

**Note:** When the TCP holds transmission with a node, it puts all the phases of the node into an HRETR or HSYST status. This depends on the current value of the retry counter, and the value specified for the RETRY operand in the NDMTCP generation macro. Refer to the *Installation and Customization* book for more information about this generation macro.

**What You Can Do**

- You can release nodes for which transmissions were held, by entering the Release command. This causes an immediate restart of transmissions to nodes that belong to the involved logical unit. After the TCP is shut down, save or print the SYSPRINT file for your system programmer's further analysis.
- If the error occurred in the unsolicited message file, you can continue to work without IOF. If it occurred in the work file for IOF displays, you can continue to work without the IOF Display function.

**Line and Node Problems**

During transmission, the following problems can occur:

- Line problems detected by VTAM/TCAM, for example, a LOSTTERM or BIND failure
- Node shutdown
- Communication logic problems, for example, an invalid function header is transmitted.

**What NetView DM for MVS Does**

In case of a line problem during the initialization of an SNA session, the TCP subtask periodically tries to restart transmission. If all the retries fail, NetView DM for MVS adds an error message to the SYSPRINT file. The retries depend on two parameters specified at installation time:

- SWDLY specifies the number of seconds the TCP must wait, before retrying to initiate another SNA session. This applies only to switched lines. The default value is five seconds.
- SWRTRY specifies the maximum number of times the TCP can retry to initiate an SNA session. This applies only to switched lines. The default value is three times.

The TCP subtask that should continue this SNA session remains “attached” while waiting for the time specified by SWDLY. Only after the maximum number of retries are reached, are the nodes and the corresponding phases “held.”
When NetView DM for MVS is using intermediate node support, all of the detected errors refer to the communication between the TCP and the front end node. The automatic retries specified via the RETRY and RETINT parameters apply to the logical unit defined for the involved nodes.

- RETINT specifies the time interval between each attempt to restart transmission.
- RETRY specifies the maximum number of retries.

The TCP does not have any control over the communication path between the intermediate node and the end nodes, so the end nodes cannot notify NetView DM for MVS of any errors.

During the intervals between the retries, NetView DM for MVS holds all of the nodes that belong to the affected logical unit. NetView DM for MVS puts the phases for all the involved nodes in an HRETR status during the intervals between the retries. If the maximum number of retries is reached and the problem has not been solved, NetView DM for MVS puts all the phases for nodes into the HSYST status, until one of the following events occurs:

- The operator releases the node.
- The operator resubmits the TCP. You must specify a cold start, if you want to reset the phase status and the retry counter for the held nodes.
- One of the involved nodes logs on to NetView DM for MVS to release a held phase or to reestablish an asynchronous communication.

When NetView DM for MVS reestablishes a connection with a node after one or more attempts, it resets the retry counter of the RETRY parameter to zero (0) and the HSYST status.

When NetView DM for MVS is using intermediate node support, all of the detected errors refer to the communication between the TCP and the front end node.

The TCP does not have any control in the communication path between the intermediate node and the end nodes, so the end nodes cannot notify NetView DM for MVS of any errors.

**What You Can Do**

Check the status of phases and functions by displaying the unsolicited message file or using IOF to display the status. Then, do one of the following:

- Hold the affected nodes in order to avoid an automatic release.
- Release the affected nodes in order to restart transmission without waiting for the RETINT interval.

---

**User Problems**

These problems result from inconsistencies in the data entered by a user when defining the:

- Installation and startup parameters
- Network configuration
- Schedules of transmission plans
User Problems

The first and the second types of problems cause a TCP shutdown, while the third does not.

Problems That Cause a TCP Shutdown
Problems that cause a TCP shutdown can result from:

- Inconsistency between the installation parameters and those defined in the VTAM tables (for example, PASSWORD and APPLID).
- Inconsistency in the definition of the network in the distributed resource directory (DRD), for example, a user defined different types of node under the same logical unit.

Note: If you have an MVS console, you can avoid a TCP shutdown in this circumstance by entering a GO command.

What NetView DM for MVS Does
The TCP issues a specific message describing the problem and immediately ends.

What You Can Do
Check the error as logged in the SYSPRINT File. If an error occurred because of an incorrect definition in the installation or startup parameters, correct the parameters and restart the TCP. If there is an inconsistency in the network configuration that the user defined in the DRD, report the problem to the system administrator.

Problems That Do Not Cause a TCP Shutdown
User errors can interrupt the transmission without shutting down the TCP. The TCP detects these errors when it executes scheduled transmission plans. Some affected functions can be restarted. This depends on what kind of error occurred. Examples of problems that interrupt the execution of functions and determine their future status are given below.

Functions That Are Put in a Restartable Status
A function is put in a RESTARTABLE status when one of the following occurs.

Note: This does not apply to user-defined nodes with change management focal point (CMFP) capability, or NetView DM for MVS transfer (NDMT) capability.

- NetView DM for MVS cannot find the resource in the repository at the host, when there is a request to send the resource to a node. The user must insert the resource into the resource repository and start the phase again. The interrupted transmission restarts.
- NetView DM for MVS tries to send an incomplete resource (for example, a partially retrieved resource). The user who defines the transmission plan can avoid this error by making the start of the sending phase dependent on the completion of the retrieving phase. Refer to “Conditioning within a Phase” on page 121 in Part 1, “Using the Generalized Interactive Executive (GIX)” of this book.
What NetView DM for MVS Does
The TCP puts any phase for synchronous nodes which contains interrupted functions into a RESTARTABLE status. Resources partially retrieved in the repository (flagged as INCOMPLETE) are not available for use. You can restart these phases and their functions by following the instructions below.

What You Can Do
You have to remove the error to restart the phases. If you are working with IOF, you can restart the phases by entering the Start Phase command for each RESTARTABLE phase on the Display and Control Transmission Schedules panel. If you are working in unattended mode, any phase in a RESTARTABLE status is resumed when the TCP is restarted.

Functions That Are Not Put in a Restartable Status
When one of the problems listed below occurs, the TCP does not put the involved functions in a RESTARTABLE status:

- The same resource already exists at the node, and the user did not specify Replace
- The same resource already exists at the host, and the user did not specify either Replace or Append
- The resource is not found at the node when a Delete or Retrieve command is issued
- There is a mismatch between the attributes at the node and those at the host
- There is a mismatch between the attributes defined in the TCF and the ones defined in the resource repository
- There is an addressing error, for example, an invalid end node specification or invalid directory name
- An unrecoverable error is detected at one of the intermediate nodes while transmitting data.

What NetView DM for MVS Does
In all of the above cases, the function is put into a COMPLETE status with a return code of 4 or 8. Return code 4 means that NetView DM for MVS issued warning messages. Return code 8 means that NetView DM for MVS encountered an unrecoverable error. If you are working with IOF, display the messages in the unsolicited message file and report the problem to the system programmer.

If you are working in unattended mode, a log of the messages is produced in the SYSPRINT file when the TCP ends. Once you have solved the problem, you can execute the interrupted function by submitting a new plan with the same function.

Problems with the Service Provider
The service provider must always be active in order to provide I/O operations for NetView DM for MVS users working with the service provider option. If the service provider fails or is not active, you receive the same type of errors given by real I/O errors but preceded by a specific service provider error message in the SYSPRINT log.
Problems with the Service Provider

You must also be very careful when starting and ending the SVP to avoid abnormal terminations. The Installation and Customization book describes how to set up, start, and end the service provider.

The following scenarios provide examples of situations where service provider failure can result in problems.

**Scenario 1: Service Provider Fails during TCP Job**

In this scenario, the service provider is active and the user starts a TCP job. After the TCP has finished initialization, the service provider fails or is shut down. The user then attempts to start the IOF to monitor the transmissions on the network.

**What NetView DM for MVS Does**

The TCP ends and the following error message appears on the screen:

```
NDM0540E VSAM ERROR OCCURRED WHEN READING THE SECURITY RECORD
OR SVP NOT ACTIVE
```

**What You Can Do**

To recover from this situation, carry out the following steps:

1. Check the status of the SVP. You can either display the active job, or look in the error log (SYSPRINT). In the error log, check the SVP error message that was issued before the general VSAM error.
2. Restart the SVP. Refer to Installation and Customization for information.
3. Restart the TCP (either a warm or cold start).

If the SVP fails again, contact your system administrator.

**Scenario 2: Service Provider Fails While Using GIX**

In this scenario, the service provider component is being used at your installation and you are now using GIX. The SVP fails. You try to perform an I/O operation using GIX and you receive an error message.

**What NetView DM Does**

The following error message appears on the screen:

```
NDM9251E REQUESTED OPERATION NOT PERFORMED BECAUSE A SYSTEM ERROR OCCURRED
```

**What You Can Do**

To recover from this situation, carry out the following steps:

1. Check the status of the SVP by looking in the active job.
2. To find the cause of the failure, look for the SVP-specific error message in the log. If you are using GIX through TSO, look in the TSO error log to find the SVP error message. If you are using GIX through ISPF, you can also look in the error log but do not select the option to delete the log when you exit from GIX.
3. Log off from GIX.
4. Restart the SVP. Refer to Installation and Customization for information.
5. Restart GIX.
Scenario 3: Service Provider Fails and Is Restarted While Using GIX

This scenario is similar to Scenario 2. You are using GIX and the SVP fails. In this scenario, however, the operator restarts the SVP without informing GIX users that the SVP failed. You try and perform an I/O operation using GIX and GIX abends.

**What NetView DM Does**

GIX abends and the following error message appears on the screen:

```
ISPF PROCESSOR ENDED ABNORMALLY USER ABEND CODE /zerodot/zerodot21 REASON CODE
```

**What You Can Do**

To recover from this situation, log off from GIX and restart GIX (providing that the SVP is still active).

Scenario 4: Service Provider Fails During Transmission

In this scenario, you start the TCP and submit a plan containing 30 transmission functions. During the transmission, the SVP fails.

**What NetView DM Does**

You receive multiple error messages, because each transmission that fails produces several messages. The messages start like this:

```
SERVER FZDSERV IS NOT ACTIVE
0
0
0
+ another nn screens of other error messages.
```

The TCP ends and no further transmissions are performed.

**What You Can Do**

To recover from this situation, perform the following steps:

1. Restart the SVP. Refer to *Installation and Customization* for information.
2. Restart the TCP.

The transmission usually recovers automatically from the point at which it failed.
# Part 4. Appendixes

## Appendix A. Using MVS MODIFY Commands with NetView DM for MVS

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controlling the TCP</td>
<td>550</td>
</tr>
<tr>
<td>Start Transmission</td>
<td>552</td>
</tr>
<tr>
<td>Quiesce Transmission</td>
<td>552</td>
</tr>
<tr>
<td>Immediate Quiesce Transmission</td>
<td>552</td>
</tr>
<tr>
<td>Display Transmission for a Node</td>
<td>552</td>
</tr>
<tr>
<td>Change CURTASK Value</td>
<td>553</td>
</tr>
<tr>
<td>Hold Transmission for a Node</td>
<td>553</td>
</tr>
<tr>
<td>Hold Transmission for a Group of Nodes</td>
<td>554</td>
</tr>
<tr>
<td>Hold Transmission for a Phase by Node</td>
<td>554</td>
</tr>
<tr>
<td>Hold a Logical Unit</td>
<td>555</td>
</tr>
<tr>
<td>Release Transmission for a Node</td>
<td>555</td>
</tr>
<tr>
<td>Release Transmission for a Group of Nodes</td>
<td>555</td>
</tr>
<tr>
<td>Release Transmission for a Phase by Node</td>
<td>556</td>
</tr>
<tr>
<td>Release a Logical Unit</td>
<td>556</td>
</tr>
<tr>
<td>Delete Transmission for a Phase by Node</td>
<td>556</td>
</tr>
<tr>
<td>Delete Transmission for an NSR</td>
<td>557</td>
</tr>
<tr>
<td>End Transmission</td>
<td>557</td>
</tr>
<tr>
<td>Immediate End Transmission</td>
<td>557</td>
</tr>
</tbody>
</table>

### Using the NetView DM for MVS Diagnostic Tools

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ending the Service Provider</td>
<td>558</td>
</tr>
<tr>
<td>End the Service Provider Normally</td>
<td>558</td>
</tr>
<tr>
<td>End the Service Provider Immediately</td>
<td>558</td>
</tr>
</tbody>
</table>

## Appendix B. NetView DM for MVS Node, Function and Resource Types

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetView DM for MVS Node Types</td>
<td>561</td>
</tr>
<tr>
<td>LU 0 Node Types</td>
<td>561</td>
</tr>
<tr>
<td>LU 6.2 Node Types (User-Defined)</td>
<td>562</td>
</tr>
<tr>
<td>NetView DM for MVS Functions</td>
<td>564</td>
</tr>
<tr>
<td>Functions and Nodes</td>
<td>564</td>
</tr>
<tr>
<td>NetView DM for MVS Resource Types</td>
<td>566</td>
</tr>
<tr>
<td>Resources and Nodes</td>
<td>566</td>
</tr>
<tr>
<td>Resources and Functions</td>
<td>567</td>
</tr>
</tbody>
</table>

## Appendix C. Resource Naming Conventions

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Naming Conventions for LU 0 Resource Types</td>
<td>570</td>
</tr>
<tr>
<td>Naming LU 0 Resources at the Node</td>
<td>570</td>
</tr>
<tr>
<td>DPPX Resources</td>
<td>571</td>
</tr>
<tr>
<td>PDOS Resources</td>
<td>571</td>
</tr>
<tr>
<td>POSC Resources</td>
<td>572</td>
</tr>
<tr>
<td>RPS Resources</td>
<td>572</td>
</tr>
<tr>
<td>SSP Resources</td>
<td>572</td>
</tr>
<tr>
<td>VSE Resources</td>
<td>574</td>
</tr>
<tr>
<td>Naming Conventions for LU 6.2 Resource Types</td>
<td>575</td>
</tr>
<tr>
<td>Enterprise-Structured Names</td>
<td>575</td>
</tr>
<tr>
<td>Subtrees</td>
<td>575</td>
</tr>
<tr>
<td>Structured Names</td>
<td>582</td>
</tr>
<tr>
<td>SNA/File Services Class Codes</td>
<td>584</td>
</tr>
<tr>
<td>Character Sets</td>
<td>585</td>
</tr>
</tbody>
</table>
Appendix A. Using MVS MODIFY Commands with NetView DM for MVS

This appendix provides a reference to the MVS MODIFY commands that you can use directly from your system console when working with NetView DM for MVS. The appendix begins by providing a complete reference list of the MVS MODIFY commands you can use. It then provides more detailed information about the commands you can use in each of the following areas:

- Controlling the TCP, as an alternative to IOF
- Using the NetView DM for MVS diagnostic tools, such as the TCP internal trace or the TCP SNAP trace
- Ending the service provider

Summary of MVS MODIFY Commands Used in NetView DM for MVS

The general format of MVS MODIFY commands is as follows:

```
MODIFY | F jobname,COMMAND[,option]
```

`jobname` is the name of the job. `COMMAND` is the name of the command, for example, `HOLD`. `option` is where you can specify certain options for some of the commands.

Table 68 lists the MVS MODIFY commands you can use in NetView DM for MVS, describes what each command does, and tells you where to look for more information in other books belonging to the NetView DM for MVS library.

<table>
<thead>
<tr>
<th>MVS MODIFY Command</th>
<th>Description</th>
<th>For More Information in the NetView DM Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>F jobname,START</td>
<td>Starts transmission</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,QUIESCE</td>
<td>Quiesces transmission</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,QUIESCE,I</td>
<td>Quiesces transmission immediately</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,D,node</td>
<td>Displays information about a node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,D,node,e</td>
<td>Displays information about a node and its active transmission requests</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,CURTASK=(number1,number2)</td>
<td>Changes the CURTASK value</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HOLD,node</td>
<td>Holds transmission for a node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HOLD,node,I</td>
<td>Immediately holds transmission for a node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HOLD,nodegroup,G</td>
<td>Holds transmission for a node group</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HOLD,nodegroup,I,G</td>
<td>Immediately holds transmission for a node group</td>
<td>User's Guide</td>
</tr>
</tbody>
</table>
### Table 68 (Page 2 of 2). MVS MODIFY Commands Used in NetView DM for MVS

<table>
<thead>
<tr>
<th>MVS MODIFY Command</th>
<th>Description</th>
<th>For More Information in the NetView DM Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>F jobname,HPH,plannename,phasename,nodename</td>
<td>Holds transmission for a phase by node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HLU,logical unit</td>
<td>Holds all the nodes belonging to a logical unit</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,HLU,logical unit,I</td>
<td>Immediately holds all the nodes belonging to a logical unit</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,RELEASE,node</td>
<td>Releases transmission for a node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,RELEASE,nodegroup,G</td>
<td>Releases transmission for a node group</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,RPH,plannename,phasename,nodename</td>
<td>Releases transmission for a phase by node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,RLU,logical unit</td>
<td>Releases all the nodes belonging to a logical unit</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,DPH,plannename,phasename,nodename</td>
<td>Deletes transmission for a phase by node</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,DNSR,originname,sequencenumber,origindate</td>
<td>Deletes an NSR</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,E</td>
<td>Ends transmission</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,I</td>
<td>Immediately ends transmission</td>
<td>User's Guide</td>
</tr>
<tr>
<td>F jobname,TRACE[,option]</td>
<td>Starts the TCP internal trace</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>F jobname,NOTRACE[,option]</td>
<td>Stops the TCP internal trace</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>F jobname,Snap</td>
<td>Starts the TCP SNAP trace</td>
<td>Diagnosis</td>
</tr>
<tr>
<td>F jobname,NOSNAP</td>
<td>Stops the TCP SNAP trace</td>
<td>Diagnosis</td>
</tr>
</tbody>
</table>

### Controlling the TCP

You can control the TCP directly from your system console using a set of MVS MODIFY commands. This is an alternative to using IOF.

Table 69 on page 551 shows the different ways in which you can control the TCP. For each task, the table shows the IOF menu option and the equivalent MVS MODIFY command.
### Table 69. Controlling the TCP: Mapping IOF Commands to MVS Commands

<table>
<thead>
<tr>
<th>Task</th>
<th>IOF Menu Option</th>
<th>MVS MODIFY Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Transmission</td>
<td>Option 1</td>
<td>F jobname,START</td>
</tr>
<tr>
<td>Quiesce Transmission</td>
<td>Option 5</td>
<td>F jobname,QUIESCE</td>
</tr>
<tr>
<td>Immediate Quiesce Transmission</td>
<td>Option 6</td>
<td>F jobname,QUIESCE,I</td>
</tr>
<tr>
<td>Display Information about a Node</td>
<td>Option 2</td>
<td>F jobname,D,node</td>
</tr>
<tr>
<td>Display Information about a Node and its Active Transmission Requests</td>
<td>Option 2</td>
<td>F jobname,D,node,e</td>
</tr>
<tr>
<td>Change CURTASK</td>
<td>Option 11</td>
<td>F jobname,CURTASK=(number1,number2)</td>
</tr>
<tr>
<td>Hold Transmission for a Single Node</td>
<td>Option 3</td>
<td>F jobname,HOLD</td>
</tr>
<tr>
<td>Immediately Hold Transmission for a Single Node</td>
<td>Option 3</td>
<td>F jobname,HOLD,I</td>
</tr>
<tr>
<td>Hold Transmission for a Group of Nodes</td>
<td>Option 3</td>
<td>F jobname,HOLD,nodegroup,G</td>
</tr>
<tr>
<td>Immediately Hold Transmission for a Group of Nodes</td>
<td>Option 3</td>
<td>F jobname,HOLD,nodegroup,1,G</td>
</tr>
<tr>
<td>Hold Transmission for a Phase by Node</td>
<td>Option 2</td>
<td>F jobname,HPH,plannname,phasename,nodename</td>
</tr>
<tr>
<td>Hold All the Nodes Belonging to a Logical Unit</td>
<td></td>
<td>F jobname,HLU,logical unit</td>
</tr>
<tr>
<td>Immediately Hold All the Nodes Belonging to a Logical Unit</td>
<td></td>
<td>F jobname,HLU,logical unit,I</td>
</tr>
<tr>
<td>Release Transmission for a Single Node</td>
<td>Option 4</td>
<td>F jobname,RELEASE,node</td>
</tr>
<tr>
<td>Release Transmission for a Group of Nodes</td>
<td>Option 4</td>
<td>F jobname,RELEASE,nodegroup,G</td>
</tr>
<tr>
<td>Release Transmission for a Phase by Node</td>
<td>Option 2</td>
<td>F jobname,RPH,plannname,phasename,nodename</td>
</tr>
<tr>
<td>Release all the Nodes Belonging to a Logical Unit</td>
<td></td>
<td>F jobname,RLU,logical unit</td>
</tr>
<tr>
<td>Delete Transmission for a Phase by Node</td>
<td>Option D</td>
<td>F jobname,DPU,plannname,phasename,nodename</td>
</tr>
<tr>
<td>Delete an NSR</td>
<td>Option D</td>
<td>F jobname,DNSR,originname,sequencenumber,origindate</td>
</tr>
<tr>
<td>End Transmission</td>
<td>Option 7</td>
<td>F jobname,E</td>
</tr>
<tr>
<td>Immediate End Transmission</td>
<td>Option 8</td>
<td>F jobname,I</td>
</tr>
</tbody>
</table>

The rest of this appendix provides a brief outline of each task. For more detailed task information, see Part 3, “Using the Interactive Operator Facility (IOF)” on page 467.
MVS MODIFY Commands to Control the TCP

Start Transmission
Use the following MVS MODIFY command from the NetView console to start transmission, and also to restart transmission after a previous quiesce request:

```
MODIFY | F jobname,START
```

You can only start transmission when the system status is QUIESCED.

Quiesce Transmission
To quiesce transmission means to stop the scheduling of transmission requests after the current resource has been transmitted. You can quiesce transmission using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,QUIESCE
```

For information on quiescing transmission using IOF, refer to “Option 5 – Quiesce Transmission” on page 529.

Immediate Quiesce Transmission
To quiesce transmission immediately means to stop the scheduling of transmission requests immediately. You can quiesce transmission immediately using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,QUIESCE,I
```

Display Transmission for a Node
You can display information about a node using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,D,node
```

jobname is the name of the job; D stands for Display; node is the name of the node.

After you enter the command to display information about the node, the following information is displayed:

- The name of the node
- The name of the logical unit
- The status of a node, which can be one of the following:
  - HOLD, by the user
  - HOLD, by the system
  - RELEASED
  - INCONSISTENT, if the node name you enter in the command does not exist or is not correct
- The remaining retry counter

You can also display details of the active transmission requests for a node, using the following MVS MODIFY command from the NetView console:
MODIFY | F jobname,D,node,e

jobname is the name of the job; D stands for Display; “e” is the option you specify to obtain information about the node and also the active transmission requests.

When you enter an MVS MODIFY command, using the optional parameter “e,” information about the node and the active transmission requests for the node is displayed.

If the unit of work is a phase, the following information is displayed:
- The plan name
- The phase name
- The name of the destination node
- The function and resource type
- The resource name
- The number of resource bytes that have been transmitted
- The percentage of bytes that have been transmitted

If the unit of work is a node-solicited request, the following information is displayed:
- The node
- The sequence number of the NSR
- The origin date and time
- The function and resource type
- The resource name
- The number of resource bytes that have been transmitted
- The percentage of bytes that have been transmitted

Change CURTASK Value

You can change the current values for the number of concurrently active SNA sessions while the TCP is running. The MAXTASK parameter of the NDMTCP generation macro originally defines the greatest possible values, but you can change the actual values while the TCP is running, using the CURTASK parameter.

Use the following MVS MODIFY command from the NetView console to change the value of the CURTASK parameter:

    MODIFY | F jobname,CURTASK=(number1,number2)

jobname is the name of the job in which you want to modify the CURTASK value.

For information on changing the CURTASK parameter using IOF, refer to Chapter 23, “Changing Transmission Profiles” on page 521, or to Chapter 19, “Starting the Transmission Control Program (TCP)” on page 471 for the startup parameter.

Hold Transmission for a Node

You can hold all transmissions for a node using the following MVS MODIFY command from the NetView console:

    MODIFY | F jobname,HOLD,node
jobname is the name of the job; node is the name of the node. The phase, or NSR, that is executing for this node is allowed to terminate and then the node is held. No transmissions can start with this node until you release the node.

To hold all transmissions for a node immediately, use the following command:

```
MODIFY | F jobname,HOLD,node,I
```

NetView DM for MVS immediately interrupts any phase, or NSR, that is executing for this node and holds the node.

For information on holding transmission using IOF, refer to Chapter 22, “Controlling Transmission with the Nodes” on page 515.

**Hold Transmission for a Group of Nodes**

You can hold all transmissions for a group of nodes using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,HOLD,nodegroup,G
```

jobname is the name of the job; nodegroup is the name of the node group.

NetView DM for MVS holds transmission for the group of nodes at the end of the current transmission. The phases, or NSRs, that are executing for the nodes of this group are allowed to terminate and then the nodes are held. No transmissions can start with the nodes in this group until you release the group.

To hold transmission for a group of nodes immediately, use the following command:

```
MODIFY | F jobname,HOLD,nodegroup,G,I
```

The phases, or NSRs, that are executing for the nodes of this group are interrupted and the nodes are held.

**Hold Transmission for a Phase by Node**

You can hold all transmissions for a specific phase for a specific node (called a phase by node) using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,HPH,plannname,phasename,nodename
```

jobname is the name of the job. plannname is the name of the plan to which the phase belongs. phasename is the name of the phase. nodename is the name of the node addressed by the phase.

You will receive a message on your system console informing you whether or not the request was successful.

NetView DM for MVS holds transmission for the phase by node at the end of the current transmission. Transmission is held until you release the phase by node. Phases with transmission windows defined are held until released; once released, the windows take effect as usual.

Refer to “H – Hold Transmission of a Phase or D&CC API Request” on page 507 for details of how the effect of holding a phase depends on the status of the phase.
Hold a Logical Unit

You can hold all the nodes belonging to a logical unit and the transmission related to the nodes using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,HLU,logicalunit
```

*jobname* is the name of the job; *logicalunit* is the name of the logical unit to which the nodes belong. The phase or NSR, that is executing for these nodes is allowed to terminate and then the nodes are held. No transmissions can start with these nodes until you release the nodes.

To hold all the nodes belonging to a logical unit immediately, use the following command:

```
MODIFY | F jobname,HLU,logicalunit,I
```

NetView DM for MVS immediately interrupts any phase or NSR that is executing for these nodes and holds the nodes.

You will receive a message on your system console informing you whether or not the request was successful.

Release Transmission for a Node

A node can be held either by a user entering an IOF command or an MVS MODIFY command, or by the system if the TCP loses contact with the node. In each case, you must release the node before transmission can start again. You can release all transmissions for a node using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,RELEASE,node
```

*jobname* is the name of the job; *node* is the name of the node.

You will receive a message on your system console informing you whether or not the request was successful.

The status of the phases, or NSRs, addressing the node changes from HDEST, HRETR, or HSYST, to READY or WAITING, depending on the scheduled date and time.

For information on releasing transmission using IOF, refer to “Option 4 – Release Transmission for a Node/Group” on page 518.

Release Transmission for a Group of Nodes

You can release all transmissions for a group of nodes using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,RELEASE,nodegroup,G
```

*jobname* is the name of the job; *nodegroup* is the name of a group of nodes. The status of the phases, or NSRs, addressing the nodes in the group changes from
MVS MODIFY Commands to Control the TCP

HDEST, HRETR, or HSYST, to READY or WAITING, depending on the scheduled date and time.

Release Transmission for a Phase by Node
You can release all transmissions that were previously held for a specific phase by node using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,RPH,planname,phasename,nodename
```

jobname is the name of the job. planname is the name of the plan to which the phase belongs. phasename is the name of the phase. nodename is the name of the node addressed by the phase.

Refer to “R – Release a Held Phase or D&CC API Request” on page 508 for details of how the effect of releasing a phase depends on the status of the phase.

Release a Logical Unit
You can release all the nodes belonging to a logical unit using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,RLU,logicalunit
```

jobname is the name of the job; logicalunit is the name of the logical unit.

You release all the nodes belonging to the logical unit that are held either by an MVS MODIFY command or by the system if the TCP loses contact with the node.

You will receive a message on your system console informing you whether or not the request was successful.

The status of the phases or NSRs that address the nodes changes from HDEST, HRETR, or HSYST, to READY or WAITING, depending on the scheduled date and time.

Delete Transmission for a Phase by Node
You can delete all transmissions for a phase by node using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,DPH,planname,phasename,nodename
```

jobname is the name of the job; planname is the name of the plan to which the phase belongs; phasename is the name of the phase; and nodename is the name of the node addressed by the phase.

You will receive a message on your system console informing you whether or not the request was successful.

Refer to “D – Delete a Transmission Request” on page 506 for details of how the effect of deleting a phase depends on the status of the phase.
Delete Transmission for an NSR

You can delete a node-solicited request using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,DNSR,originname,sequencenumber,origindate
```

`jobname` is the name of the job; `originname` is the name of the node that generates a NSR; `sequencenumber` is the number that identifies the NSR; and `origindate` is the date on which the NSR is generated.

You will receive a message on your system console informing you whether or not the request was successful.

Refer to "D – Delete a Transmission Request" on page 506 for details of how the effect of deleting a phase depends on the status of the phase.

End Transmission

To end transmission means to end transmission activity as soon as all active transmission requests have been processed and then stop the TCP.

You can end transmission in one of the following ways:

- Specifying AUTOEND=YES as a TCP startup parameter
- Using IOF Option 7
- Using an MVS MODIFY command

Use the following MVS MODIFY command from the NetView console to end transmission:

```
MODIFY | F jobname,E
```

The TCP stops when it has processed all active transmission requests.

For information on ending transmission using IOF, refer to “Option 7 – End Transmission” on page 530.

Immediate End Transmission

To immediately end transmission means to end all transmission activity immediately and stop the TCP. You can immediately end transmission using the following MVS MODIFY command from the NetView console:

```
MODIFY | F jobname,I
```

Using the NetView DM for MVS Diagnostic Tools

NetView DM for MVS provides a set of trace tools that help you isolate the cause of problems that might occur during TCP operation:

**Storage trace**

The storage trace is always active during a TCP run. The storage trace uses the virtual storage trace table to store trace data. You can view the contents of the trace table in the SNAP data set when an abend occurs or in a dynamic storage dump.
TCP internal trace  This trace is not active at TCP startup. This trace is written into the generalized trace facility (GTF) data set, if the GTF is active.

Use the following MVS MODIFY command from the system console to start a TCP internal trace:

    MODIFY | F jobname,TRACE[,option]

*jobname* is the name of the job to be traced. *option* allows you to specify selection criteria for the trace, so that only certain events are recorded in the GTF data set. These options are described in the *Diagnosis* book.

To stop the current trace, use the following command:

    MODIFY | F jobname,NOTRACE

TCP SNAP trace  As with the TCP internal trace, this trace is not active at TCP startup. This trace is written into a SNAP data set.

Use the following MVS MODIFY command from the system console to start a TCP SNAP trace:

    MODIFY | F jobname,SNAP[,option]

*jobname* is the name of the job to be traced. *option* allows you to specify optional selection criteria for the events to be recorded. The events handled by this trace are different from those handled by the internal trace. These options are described in the *Diagnosis* book.

To stop the current trace, use the following command:

    MODIFY | F jobname,NOSNAP

Ending the Service Provider

You can use MVS MODIFY commands to end the service provider either with a Normal End or an Immediate End command.

End the Service Provider Normally

Always use the MVS MODIFY command to carry out a Normal End. This allows all NetView DM for MVS users to end normally, because the service provider ends only when all of the users are disconnected. The format of the Normal End command is as follows:

    F jobname,E

*jobname* is the name of the service provider job.

End the Service Provider Immediately

If it is absolutely necessary, you can use an Immediate End command to end the service provider, but you must ensure that you end any NetView DM for MVS applications first. The format of the Immediate End command is as follows:

    F jobname,I
The Immediate End command has the effect of forcing the disk containing the VSAM files used by the service provider offline. This means that any I/O operation is interrupted and NetView DM for MVS users receive an I/O error.

**Warning:** *Do not* end the service provider with an Immediate End command without ending all NetView DM for MVS applications first. If you do, the service provider ends without waiting for users to disconnect first.

The service provider must always stay active to allow the I/O operation of NetView DM for MVS users working with the service provider option. If the service provider fails or is not active, you receive the same type of error message given by a “real” I/O error, but preceded by an service provider error message in the SYSPRINT log.
Appendix B. NetView DM for MVS Node, Function and Resource Types

This appendix provides reference information on the node types, functions, and resource types that you can use with NetView DM for MVS. The node type you are working with determines the functions and resources that you can use. This appendix contains a set of tables that illustrate the relationship between node type, function, and resource type.

NetView DM for MVS Node Types

The following tables show the relationship that exists between the node types and the hardware, operating system, and functional capabilities (when appropriate).

LU 0 Node Types

Here are the LU 0 node types you can define:

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Hardware/Operating System/Functional Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPPX</td>
<td>• IBM processors using the DPPX/370 Operating System (for functions not related to the distribution and installation of IBM Licensed Internal Code)</td>
</tr>
</tbody>
</table>
| PDOS      | • IBM PC or IBM PS/2 under the control of IBM DOS connected to the host via S/36 or AS/400 intermediate nodes  
   • IBM PC or IBM PS/2 under the control of IBM DOS or IBM OS/2, as appropriate, directly connected to the host (3270 Emulation). |
| POSC      | • IBM 4680 or 4684 Store System Processors under the control of the 4680 Operating System  
   • IBM 4684 System under the control of IBM DOS  
   • IBM PC or IBM PS/2 under the control of IBM DOS or IBM OS/2, as appropriate, directly connected to the host. |
| RPS       | • IBM RISC System/6000 under control of the AIX Operating System. |
| SSP       | • IBM AS/400 under the control of the OS/400 Operating System or an IBM S/36 under the control of the SSP Operating System directly connected to the host  
   • IBM AS/400 under the control of the OS/400 Operating System or an IBM S/36 under the control of the SSP Operating System connected to the host through an AS/400 or S/36 intermediate nodes. |
| VSE       | • IBM processors using the VSE Operating System. |
LU 6.2 Node Types (User-Defined)

Here are the user-defined LU 6.2 node types supported by NetView DM for MVS:

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Hardware/Operating System/Functional Capability</th>
</tr>
</thead>
</table>
| **user-defined** (PS/2, PS/55, PC) | • IBM PS/2, IBM PS/55, and IBM PC equipped with the NetView DM/2 counterpart program, directly connected to the host  
• IBM PS/2, PS/55, and PC equipped with the NetView DM/2 counterpart program, connected to the host by one of the following acting as a SNA/DS intermediate node:  
  – An IBM PS/2 or IBM PC equipped with NetView DM/2  
  – An AS/400 equipped with the OS/400 Operating System Version 2 Release 3 or later, and SystemView Managed System Services/400  
  – A RISC System/6000 equipped with NetView DM/6000  
• IBM PS/2, PS/55, and PC equipped with the NetView DM/2 counterpart program in a NetBIOS LAN that is controlled by a PS/2, PS/55, or PC equipped with NetView DM/2, either directly connected to the host or to a PS/2, PS/55, or PC equipped with NetView DM/2, acting as an intermediate node  
• IBM PS/2, PS/55, and PC equipped with NetView DM Agent/2 in a TCP/IP network that is controlled by an IBM RISC System/6000 equipped with NetView DM/6000 and connected to the host  
• IBM PS/2, PS/55, and PC equipped with NetView DM Agent/2 in an IPX network that is controlled by a RISC System/6000 equipped with NetView DM/6000 and connected to the host  
• IBM PS/2, PS/55, and PC equipped with NetView DM Agent/DOS in an IPX network that is controlled by a workstation equipped with NetView DM for NetWare and connected to the host  
• IBM PS/2, PS/55, and PC equipped with NetView DM Agent for Windows in a TCP/IP network that is controlled by a RISC System/6000 equipped with NetView DM/6000 and connected to the host  
• IBM PS/2, PS/55, and PC equipped with NetView DM Agent for Windows in an IPX network that is controlled by a workstation equipped with NetView DM for NetWare and connected to the host  
These nodes have CMEP capabilities. |
<p>| <strong>user-defined</strong> (MVS) | • IBM processors under the control of the MVS operating system, running NetView DM for MVS. These nodes have CMFP capabilities, or NDMT capabilities, or both. |
| <strong>user-defined</strong> (3174) | • IBM 3174 Subsystem. This node has CMEP capabilities. |
| <strong>user-defined</strong> (9371) | • IBM 9371 systems running under DPPX/370 Release 3 (used only for centrally controlled distribution and installation of 9371 LIC). See also “DPPX” node type. These nodes have CMEP capabilities. |</p>
<table>
<thead>
<tr>
<th>Node Type</th>
<th>Hardware/Operating System/Functional Capability</th>
</tr>
</thead>
</table>
| user-defined (AS/400) | - IBM AS/400 equipped with the OS/400 Operating System Version 2 Release 3 or later, and SystemView Managed System Services/400, directly connected to the host  
- IBM AS/400 equipped with the OS/400 Operating System Version 2 Release 3 or later, and SystemView Managed System Services/400, connected to the host by an AS/400 equipped with OS/400 Version 2 Release 3 or later, acting as an intermediate node  
- IBM AS/400 equipped with the OS/400 Operating System Version 2 Release 3 or later, and SystemView Managed System Services/400, connected to the host by an IBM PS/2, PS/55, or PC equipped with NetView DM/2  
- IBM AS/400 equipped with the OS/400 Operating System Version 2 Release 3 or later, and SystemView Managed System Services/400, connected to the host by an IBM RISC System/6000 equipped with NetView DM/6000  
These nodes have CMEP capabilities. |
| user-defined (Workstations supporting NetView DM for NetWare) | - Workstations equipped with NetView DM for NetWare, NetWare 3.11 or later, and NetWare for SAA, directly connected to the host  
- Workstations equipped with NetView DM for NetWare, NetWare 3.11 or later, and NetWare for SAA in an IPX network, connected to the host by an IBM PS/2 or PC equipped with NetView DM/2 acting as an intermediate node  
- Workstations equipped with NetView DM for NetWare, NetWare 3.11 or later, and NetWare for SAA, in an IPX network, connected to the host by an IBM AS/400 equipped with OS/400 Version 2 Release 3, and SystemView Managed System Services/400, acting as an intermediate node  
- Workstations equipped with NetView DM for NetWare, NetWare 3.11 or later, and NetWare for SAA in an IPX network, connected to the host by an IBM RISC System/6000 equipped with NetView DM/6000 acting as an intermediate node  
- Workstations equipped with NetView DM for NetWare acting as a CC client in an IPX network that is controlled by a workstation equipped with NetView DM for NetWare acting as a CC server  
These nodes have CMEP capabilities. |
| user-defined (RISC System/6000) | - IBM RISC System/6000 equipped with IBM AIX Version 3.2.2 or later and NetView DM/6000, directly connected to the host  
- IBM RISC System/6000 equipped with IBM AIX Version 3.2.2 or later and NetView DM/6000, connected to the host by an IBM PS/2 or IBM PC equipped with NetView DM/2, acting as an intermediate node  
- IBM RISC System/6000 equipped with IBM AIX Version 3.2.2 or later and NetView DM/6000, connected to the host by an AS/400 equipped with OS/400 Version 2 Release 3, acting as an intermediate node  
- IBM RISC System/6000 equipped with NetView DM Agent/6000 in a TCP/IP network that is controlled by an IBM RISC System/6000 equipped with NetView DM/6000 connected to the host  
These nodes have CMEP capabilities. |
| user-defined (HP-UX Workstations) | - HP-UX workstations equipped with NetView DM Agent for HP-UX in a TCP/IP network that is controlled by an IBM RISC System/6000 equipped with NetView DM/6000 and connected to the host. |
| user-defined (SunOS Workstations) | - SunOS workstations equipped with NetView DM Agent for SunOS in a TCP/IP network that is controlled by an IBM RISC System/6000 equipped with NetView DM/6000 and connected to the host. |
Functions and Nodes

Table 71 (Page 3 of 3). NetView DM for MVS LU 6.2 (user-defined) Node Types

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Hardware/Operating System/Functional Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>user-defined (Sun Solaris Workstations)</td>
<td>• Sun Solaris workstations equipped with NetView DM Agent for Solaris in a TCP/IP network that is controlled by an IBM RISC System/6000 equipped with NetView DM/6000 and connected to the host.</td>
</tr>
</tbody>
</table>

Note: The person who installs NetView DM for MVS specifies a code (user-defined) in the NDMNODE customization macro for each of the LU 6.2 node types, and associates the functions that the node supports with this type code. For example, a particular node type can support change management focal point (CMFP) functions, change management end point (CMEP) functions, NetView DM for MVS transfer (NDMT) functions, or both CMFP and NDMT functions.

NetView DM for MVS Functions

The following table shows the basic transmission functions of NetView DM for MVS. Each function has an associated function type name and a function type code that can be recognized by NetView DM for MVS.

Table 72. Function Type Names and Type Codes

<table>
<thead>
<tr>
<th>Function</th>
<th>Type Name</th>
<th>Type Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accept</td>
<td>ACC</td>
<td>0009</td>
</tr>
<tr>
<td>Activate</td>
<td>ACT</td>
<td>0012</td>
</tr>
<tr>
<td>Clear</td>
<td>CLE</td>
<td>0006</td>
</tr>
<tr>
<td>Delete</td>
<td>DEL</td>
<td>0003</td>
</tr>
<tr>
<td>Initiate</td>
<td>INI</td>
<td>0004</td>
</tr>
<tr>
<td>Install</td>
<td>INS</td>
<td>0007</td>
</tr>
<tr>
<td>Remove</td>
<td>REM</td>
<td>0008</td>
</tr>
<tr>
<td>Retrieve</td>
<td>RET</td>
<td>0002</td>
</tr>
<tr>
<td>Send</td>
<td>SEN</td>
<td>0001</td>
</tr>
<tr>
<td>Send Power Off</td>
<td>SEN</td>
<td>0011</td>
</tr>
<tr>
<td>Uninstall</td>
<td>UNI</td>
<td>0013</td>
</tr>
</tbody>
</table>

Functions and Nodes

The following table shows which functions are available at which nodes. A “Y” indicates that the function is available at the node.

Table 73 (Page 1 of 2). Functions and Nodes

<table>
<thead>
<tr>
<th>Function Name</th>
<th>CMEP</th>
<th>CMFP</th>
<th>RPS</th>
<th>SSP</th>
<th>PDOS</th>
<th>POSC</th>
<th>DPPX</th>
<th>NDMT</th>
<th>VSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTIVATE (NODE)</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Function Name</td>
<td>CMEP</td>
<td>CMFP</td>
<td>RPS SSP PDOS</td>
<td>DPPX</td>
<td>NDMT</td>
<td>VSE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>------</td>
<td>--------------</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITIATE</td>
<td>Y(^3)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTALL</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REMOVE</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RETRIEVE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEND</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UNINSTALL</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^3\) The INITIATE function is not supported by 3174 nodes with CMEP functional capabilities.
Resources and Nodes

NetView DM for MVS Resource Types

The following table shows the resource types that you can use and whether each resource is stored in the holding file or the library. Each resource type has an associated type name and type code. The type code is unique for each resource type, except that microcode and microcode customization share the same code for compatibility with previous releases of NetView DM for MVS.

<table>
<thead>
<tr>
<th>Resource Type</th>
<th>Type Name</th>
<th>Type Code</th>
<th>HF</th>
<th>LIB</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS/400 data object containment structure*</td>
<td>AS400CTN</td>
<td>0158</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AS/400 object*</td>
<td>AS400OBJ</td>
<td>0156</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td>CLIST</td>
<td>0040</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Configuration file*</td>
<td>CONFILE</td>
<td>0220</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Data set</td>
<td>DATASET</td>
<td>0030</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dump (CMEP)*</td>
<td>MSDUMP</td>
<td>0230</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Errorlog (DPPX)</td>
<td>ERRORLOG</td>
<td>0034</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Error log (CMEP)*</td>
<td>MSERRLOG</td>
<td>0250</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Flat data*</td>
<td>FLATD</td>
<td>0100</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Job</td>
<td>JOB</td>
<td>0041</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Library</td>
<td>LIBRARY</td>
<td>0052</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Member</td>
<td>MEMBER</td>
<td>0050</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Message out</td>
<td>MESSAGE</td>
<td>0017</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Microcode and Microcode Customization*</td>
<td>MICROCODE</td>
<td>0060</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Panel</td>
<td>PANEL</td>
<td>0036</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Print</td>
<td>PRINT</td>
<td>0038</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Procedure*</td>
<td>PROC</td>
<td>0080</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>PROGRAM</td>
<td>0042</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PTF</td>
<td>PTF</td>
<td>0044</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Relational data*</td>
<td>RELDB</td>
<td>0120</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Software*</td>
<td>SOFTW</td>
<td>0070</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Storage dump (LU 0)</td>
<td>DUMP</td>
<td>0032</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Trace*</td>
<td>TRACINFO</td>
<td>0240</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Note: Resources marked with an asterisk (*) are LU 6.2 resource types. You can only use them with LU 6.2 node types.

Resources and Nodes

The following table shows which resources are available at which nodes. A “Y” indicates that the resource is available at the node.

Note: Resources that are not LU 6.2 resource types can only be deleted at, retrieved from, or sent to, nodes that have NDMT capability.
The following table shows which LU 6.2 resources can be used with which functions. A “Y” indicates that the function is available for the resource type.

<table>
<thead>
<tr>
<th>Resource Type Name</th>
<th>CMEP</th>
<th>CMFP</th>
<th>RPS</th>
<th>SSP</th>
<th>PDOS</th>
<th>POSC</th>
<th>DPPX</th>
<th>NDMT</th>
<th>VSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS400CTN</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS400OBJ</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIST</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONFILE</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATASET</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMP</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERRORLOG</td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FILE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLATD</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIBRARY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEMBER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESSAGE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MICROCODE</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSDUMP</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSERRLOG</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>MSGOUT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>PANEL</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRINT</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRINTDATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>PROC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROGRAM</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RELDB</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOFTW</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACINFO</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
</tr>
</tbody>
</table>

**Resources and Functions**

The following table shows which LU 6.2 resources can be used with which functions. A “Y” indicates that the function is available for the resource type.
### Table 76. LU 6.2 Resources and Functions

<table>
<thead>
<tr>
<th>Resource Type Name</th>
<th>Accept</th>
<th>Delete</th>
<th>Install</th>
<th>Remove</th>
<th>Retrieve</th>
<th>Send</th>
<th>Initiate</th>
<th>Uninstall</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS400CTN</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS400OBJ</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONFILE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLATD</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>MSDUMP</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>MSERRLOG</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>MICROCODE</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PROC</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>RELDB</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>SOFTW</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>TRACINFO</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Appendix C. Resource Naming Conventions

This appendix describes the NetView DM for MVS naming conventions that you should follow when naming software resources at each node. The appendix contains:

- Naming conventions for LU 0 resource types. Each resource type has a different set of syntax conventions. These are described in alphabetical order.

- Naming conventions for LU 6.2 resource types. This includes information about enterprise-structured names and structured names for microcode.

- The SNA/File Services class codes supported by NetView DM for MVS Release 3 or later, and how these class codes map on to LU 6.2 resource types.

- Character set 1134, used in the routing group name (RGN) and routing element name (REN) parameters.

In the syntax definitions, the letters A through Z, and the special characters @ (at sign), # (pound), $ (dollar), % (percent), _ (underscore), and & (ampersand) are understood to be alphabetic.

Attention: Do not use _ (underscore) as a pad character in the input field of GIX panels. Otherwise, leading and trailing underscore characters are deleted when NetView DM for MVS processes the input field.

Note: The Batch Utilities validate the length of the fully qualified name, but not the length or number of the individual qualifiers. GIX validates the length and number of the individual qualifiers.
Resource Naming Conventions for LU 0 Resource Types

The naming conventions for LU 0 resource types vary according to the specific resource type. However, there are some specific rules for naming LU 0 resource types at the node. This section begins by describing the conventions that apply to all LU 0 resource types at the node. The section describes the conventions specific to each of these resource types:

- DPPX
- PDOS
- POSC
- RPS
- SSP
- VSE

Naming LU 0 Resources at the Node

This section describes the method you can use to name LU 0 resources at the node. You can name resources at the node in both GIX and Batch Utilities:

- In GIX, you use the Name at Node field
- In Batch Utilities, you use the NODENAME operand.

The name at node is the fully qualified name that the resource is to be known by at the node. There are several different points at which you can specify the name of the resource at the node:

1. When the function is specified in a transmission plan
2. If resource attributes are changed when the resource is assigned to a node
3. When you define the resource in the DRD.

If you specify the name at node in more than one place, the order of priority is from (1) to (3) above. If you do not specify a name at node, the resource name is the default value.

For resources that allow qualified names, you can use an asterisk (*) as a qualifier when you specify the name by which a resource is to be known at the node. The qualifier with an asterisk is replaced at the node by the value assigned for that qualifier at the host. The replacement of asterisks with a value is positionally dependent. Replacement starts at the right and moves to the left. Asterisks that do not have a corresponding value to be replaced are dropped, along with their position (see the example below).

When you specify a function for a group of resources, at least one of the qualifiers for the name at node must be an asterisk. You can even enter a single asterisk only in the field or operand.

You can only use an asterisk as qualifier for the name at node for certain types of LU resources. These are:

- **DPPX**: CLIST, data set, dump, panel, program
- **PDOS**: CLIST, data set, panel, program
- **POSC**: CLIST, data set, dump, panel, program
- **RPS**: CLIST, data set, dump, panel, program
- **SSP**: CLIST, data set, dump, panel, program
- **VSE**: Data set, library
Figure 137 on page 571 illustrates some examples of the different points at which you can name a resource at the node, and the resulting name.

### DPPX Resources

**CLIST, Data Set, Dump, Panel, Print, and Program Names:** Enter a fully qualified name for these resource types. You can enter 1–26 alphanumeric characters, including periods, in the form aaaaaaaaa.bbbbbbbb.cccccccc. The periods are delimiters between characters. The qualifiers are:

- **aaaaaaa** This is the 1–8 character alphanumeric name of the volume catalog (optional)
- **bbbbbbb** This is the 1–8 character alphanumeric name of the user catalog (optional)
- **ccccccc** This is the 1–8 character alphanumeric name of the resource (always required).

NetView DM for MVS validates the length of the fully qualified name (GIX and Batch Utilities), and the length and number of the individual qualifiers (GIX only).

**Note:** If you are inserting a CLIST, using GIX or the batch utility control statement INSERT CLIST, the name must be unqualified (1–8 alphanumeric characters), and you can enter only one ID.

**Errorlog and PTF Names:** Enter 1–8 alphanumeric characters. The first character must be alphabetic.

### PDOS Resources

**CLIST, Data Set, Panel, and Program Names:** Specify the fully qualified name. You can enter 1–35 alphanumeric characters, including periods, in the format aaaaaaaaa.bbbbbbbb.cccccccc.dddddddd. The meaning of each qualifier depends on the number of qualifiers.

If you have **four qualifiers**, the meaning is as follows:

- **aaaaaaa** DOS drive letter
- **bbbbbbb** Name of subdirectory
- **ccccccc** DOS file name
- **ddddd** DOS extension.
**SSP Resources**

**Note:** In order to use four qualifiers and use the default drive at the PC/Node Manager (PC/Node Manager) node, you should define the first qualifier as the wildcard option in NetView DM for MVS. The default wildcard option is `§`. For example, if you define a resource as:

```
§aaaaaaaa.bbbbbbb.bcc
```

where `aaaaaaaa` is the name of the subdirectory, and `bbbbbb.bcc` is the name of the file, it will be stored in whichever drive PCNM was started from.

If you have **three qualifiers**:

- `aaaaaaaa` DOS drive letter, or name of subdirectory
- `bbbbbbb` DOS file name
- `ccccccc` DOS extension.

If you have **two qualifiers**:

- `aaaaaaaa` DOS file name
- `bbbbbbb` DOS extension.

If you have **one qualifier**:

- `aaaaaaaa` DOS file name

**POSC Resources**

**CLIST, Data Set, Dump, Panel, and Program Names:** Specify the fully qualified name for these resource types. You can enter 1–26 alphanumeric characters including periods. Each of the three qualifiers can contain 1–8 alphanumeric characters. The following characters are valid anywhere in the name: `A-Z`, `0-9`, `.$`, `,`, `&`, `@`, and `#`. The qualifiers are:

- `aaaaaaaa` Logical name (necessary)
- `bbbbbbb` POSC file name (optional)
- `ccccccc` POSC extension (optional).

**RPS Resources**

**CLIST, Data Set, Dump, Panel, Print, and Program Names:** Specify the fully qualified name for each of these resource types. You can enter 1–35 alphanumeric characters, including periods, in the format `aaaaaaa.bbbbbbb.cccccc.ddddddd`. Each of the four qualifiers must be composed of 1–8 alphanumeric characters. The catalog name used at a node for a Retrieve function has only three qualifiers, in the format `aaaaaaa.bbbbbbb.cccccc`. The qualifiers are:

- `aaaaaaa` Device name (optional). If you do not specify a name, the system resident volume is used.
- `bbbbbbb` Logical volume name (necessary).
- `ccccccc` Data set name (necessary, unless the request is for a logical volume).
- `ddddddd` Member name (use only for members of partitioned data sets).

**SSP Resources**

**CLIST, Data Set, Dump, Panel, and Program Names:** The conventions you must use when naming SSP resources to use in NetView DM for MVS are as follows:

- A name can have up to four qualifiers
- Each qualifier can have up to eight characters
- The following characters are valid anywhere: `A-Z`, `0-9`, `.$`, `@`, and `&`
- There is no restriction on the first character of each qualifier.
These conventions make it possible to refer to all SSP resources as data sets. SSP performs more detailed checks on resource names than those performed by NetView DM for MVS. SSP checks the fully qualified resource name. It uses the number of qualifiers in a name to determine whether it is a file (one or two qualifiers), library (two qualifiers), or member (three or four qualifiers). Qualifiers are separated by a period. Names with two qualifiers can be either in the form abbbbb.ccccc, or abbbbb.dddd, where:

- **abbbbb**: must be A-Z, or #, $, or @. **cccccc** must be A-Z, 0-9, or #, $, @, or &. If the file name contains periods, substitute an ampersand for each period (for example to retrieve #DUMP./zerodot/zerodot, specify RETRIEVE DUMP ID=#DUMP&/zerodot). When specifying a file, the first qualifier cannot be the word ALL. When specifying a library, the first qualifier cannot be ALL, DISK, F1, PRINT, or READER.
  
  You must specify the first qualifier.

- **cccccc**: This second qualifier is optional for a file name. If specified, it must be the six decimal digits that are the SSP creation date of the file, in the system date format of the system that is being referenced.
  
  **dddd**: This second qualifier is necessary if the first qualifier contains a library name. It must be the four characters LIBR. The only valid function for a library is DELETE.

Names with three or four qualifiers are in the form abbbbb.eeeeee.fgggggg.hhhhhh:

- **abbbbb**: The first qualifier specifies the SSP library in which the member can be found. The syntax is the same as for a two qualifier name, described above.
  
  - **LOAD** (load members)
  - **PROC** (procedure members)
  - **SOURCE** (source members)
  - **SUBR** (relocatable subroutine members).

- **fggggggg**: The third qualifier specifies the SSP member name (1-8 characters) where **f** must be A-Z, or #, $, or @, and **gggggg** must be A-Z, 0-9, or #, $, or @, and must not be any of the following:
  
  **ALL** , **DIR** , **NEW** , or **SYSTEM**.

- **hhhhhhhh**: The fourth qualifier is optional. It can be a 1-8 character string which contains user data. It can be used to specify a subtype for NetView DM for MVS reporting, but it is not used by the communications and systems management (C&SM) feature of the SSP. Valid characters are A-Z, 0-9, and #, $, or @. Any valid character can be used as the first character.

**AS/400 Library, Object, and Member Naming Conventions:** The conventions you must use when naming an AS/400 library, object, or member, to GIX or the Batch Utilities, are given below. The “name at node” resource qualifiers map to the object name on the AS/400 system in one of the following ways:

1. **library.object.type**

   These qualifiers are:

   - **library** The 1 to 8 character name of the AS/400 library
   - **object** The 1 to 8 character name of the AS/400 object
   - **type** The object type as used on the AS/400 system.

   **Note:** Refer to the AS/400 Communications and Systems Management User’s Guide for a list of valid types.

2. **library.file.member.MEM**

   These qualifiers are:

   - **library** The 1 to 8 character name of the AS/400 library
   - **file** Source physical file (data base file)
   - **member** The member in the data base file
MEM Indicates that this is a data base file member.

3. library.file.MEM

These qualifiers have the same meanings as described above. However, this form can be used when retrieving, sending, or deleting data differentiated file members used with the System/36 environment. You may also use the four-qualifier format described above for data differentiated file operations, but the member name must always be specified. If you choose to use the three-qualifier format, AS/400 DSNX assigns a member name as follows (yymmdd is the current date on the AS/400 system):

Note:

- If MEM is not specified as the fourth qualifier, the fourth qualifier is ignored, and the object name uses the format library.object.type.
- NetView DM for MVS supports AS/400 resource qualifiers with not more than 8 characters.
- If the destination of the file is a system other than an AS/400 system, the fourth qualifier (MEM) must be used. This prevents OS/400 DSNX from saving the object as a save file.
- To retrieve a file that is already in the save file format, the three part name (library.file.SAVF) should be used.

<table>
<thead>
<tr>
<th>NetView DM for MVS Request</th>
<th>AS/400 DSNX Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>RET lib.file.MEM</td>
<td>Return member lib.file.(LAST) to NetView DM for MVS.</td>
</tr>
<tr>
<td>SEND lib.file.MEM (add option)</td>
<td>Check for the existence of member lib.file.(Myymmdd). If the member exists, do not replace it, return an error to NetView DM for MVS. If the member does not exist, add member lib.file.(Myymmdd).</td>
</tr>
<tr>
<td>SEND lib.file.MEM (replace option)</td>
<td>Check for the existence of member lib.file.(Myymmdd). If the member exists, replace it. If the member does not exist, add member lib.file.(Myymmdd).</td>
</tr>
<tr>
<td>DELETE lib.file.MEM</td>
<td>Delete member lib.file.(Myymmdd).</td>
</tr>
</tbody>
</table>

VSE Resources

Catalog Name: You must specify the fully qualified catalog name. You can enter 1-44 alphanumeric characters, including periods. The periods are delimiters between levels. Each qualification level may have 1-8 characters.

Data Set Name: You must specify the fully qualified data set name. The same rules apply as for the catalog name above.

Job Name: Enter 1-8 alphanumeric characters. The first character must be alphabetic.

Library Name: Enter a library file name or identifier of 1-7 alphanumeric characters. The first character must be alphabetic.

Library/Membership Type: For VSE 1.1 library management (old library), specify the type of library as one alphabetic character:

- C Core image library
- P Procedure library
- R Relocatable library
- S Source statement library.
For VSE 2.1 and VSE 3.1 (new library), specify the type of member. You can enter 1-8 alphanumeric characters.

**Member, Panel, and Program Name:** Enter 1-8 alphanumeric characters.

**Sublibrary Name:** For VSE 1.1 library management, this is one alphanumeric character, only for library type **S** (source). For VSE 2.1 and VSE 3.1, you can enter 1-8 alphanumeric characters.

---

**Naming Conventions for LU 6.2 Resource Types**

In NetView DM for MVS, LU 6.2 resource names consist of up to ten qualifiers, separated by “.” characters (periods). Each qualifier can be up to 16 characters long. The characters must be uppercase alphabetic or numeric characters. The total name length cannot be more than 64 characters, including the separator characters.

There are two types of naming conventions for LU 6.2 resources:

- Enterprise-structured names
- Structured names.

**Enterprise-Structured Names**

All LU 6.2 resource types use enterprise-structured names. Microcode and microcode customization data resources can also use structured names. Enterprise-structured names are free-form, apart from the first token which must always be the registered corporation ID of the data object owner. Microcode, software, flat data and relational data resources can also use subtrees defined by SNA/Management Services as a subset of their tokens.

Table 77 shows the tokens to use for all resources that use enterprise-structured names:

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, not subtree ID, unordered - type unspecified</td>
<td>Corporation ID (enterprise code from the Structured Netid registered with IBM)</td>
</tr>
<tr>
<td>2 - 10</td>
<td>User specifiable</td>
<td>User specifiable</td>
</tr>
</tbody>
</table>

Information about the registered values allowed in the Corporation ID token is available through your IBM branch office.

**Subtrees**

The user-specifiable tokens of the enterprise-structured name contain the component name token(s) and may contain as a subset, one of the architected subtrees as defined in the tables on the following pages. An architected subtree may be used in global names that represent a change (to be consolidated) or a changed (consolidated into) object. Architected subtrees are identified by the presence of an SNA/File Services registered subtree identifier (STI) token having the STI token attribute as defined by SNA/File Services. SNA/Management Services defines the manner in which a change object having one of various architected subtrees in its name applies to a changed object having the group architected subtree in its name.
While microcode and microcode customization change objects are supported by enterprise-structured names, the ability to manage changes using the MCODE and MCUST change management structured names is unaffected.

The management of changes using enterprise-structured names and architected subtrees is based on the notion of the SNA/File Services group object and the following change management concepts apply:

1. A clear distinction is made between a distributed change (to be consolidated) object and the target (consolidated into) object. The target object is also known as the group object.
2. Except in preparation nodes, change objects are consolidated into the group (changed) object.
3. Although the group is a persistent (as opposed to change) object there is no persistence of any installed child objects (elements of the group) of the installed parent group object.
4. There are no change management operations upon consolidated child objects. Such operations only act upon parent group (consolidated into) objects.
5. The token in the name tree at which a group object is created can vary from node to node and time to time without affecting the change management operations that change the groups.
6. All objects in a family share the same level number space. Parent changes with a level that is less than already installed child changes are permitted only if system dependent logic can ensure that no child back-levelling will result.
7. Group objects are never transported.

### Subtree for Group Objects

#### Table 78. Identification Tokens of the Architected Subtree for Group Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C’GRP’</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, unordered LU name</td>
<td>The LU name of the focal point issuing the most recent command affecting this group object</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, not subtree ID, ordered date</td>
<td>The date of the most recent command affecting this group object</td>
</tr>
<tr>
<td>4</td>
<td>Need not match, not generable, not subtree ID, ordered numeric</td>
<td>The command sequence number of the most recent command affecting this group object</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing changed (in other words, consolidated into) objects, this is the SNA/Management Services definition of the SNA/File Services owned GRP subtree for group objects. The GRP subtree identifier token is registered by SNA/File Services.

The existence of a group object and its elemental (in other words, children and grandchildren) objects is implied by the component token(s) in the SNA/File Services hierarchical name. All change management operations for the components represented by the group's component name are targeted to the group object. While SNA/File Services supports group objects that contain other group objects, SNA/Management Services does not support managing changes on nested groups. Group object names
include a GRP token at the end of their common root token string. Elementary objects do not include a GRP token.

Group objects must follow the same rules as do elementary objects with respect to supporting the function of the SNA/File Services global catalog. Specifically, group object names must be unique and object immutability must be enforced. It is the responsibility of the name tree designer and the creator of the objects named with GRP subtrees to ensure that subtree tokens provide for unique object names. For SNA/Management Services, the subtree tokens as defined above satisfy this requirement.

**Subtree for Refresh Change Objects**

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C'REF'</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered numeric</td>
<td>Level: The maintenance level to which this change object refreshes</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned REF subtree to be used in distributed *refresh* change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The REF subtree identifier token is registered by SNA/File Services.

The REF subtree is used in the *refresh* type of distributed change objects. These objects create or replace all or some of the contents of a group object, depending upon the granularity of the component name in the name of the distributed refresh change object. When replacing all the contents, the resultant level of the group object is indicated by the *level* token in the REF subtree in the name of the refresh change object. When elemental (in other words, child or grandchild) refresh change objects are installed, their installation does not affect the level of the parent group as a whole, but instead affects the level of the child or grandchild portion of the group object as indicated by the *level* token.

When the parent group is refreshed, the previous parent refresh and all subsequent related change entries, including those for children and grandchildren, are deleted from the SNA/File Services global catalog.
Subtree for Update Change Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C’UPD’</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered numeric</td>
<td>Oldlvl: The lowest maintenance level from which this change can update</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, not subtree ID, ordered numeric</td>
<td>Newlvl: The maintenance level to which this change object updates</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned UPD subtree to be used in distributed update change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The UPD subtree identifier token is registered by SNA/File Services.

The UPD subtree is used in the update type of distributed change objects. These objects update the level of all or some of the contents of a group object, depending upon the granularity of the component name in the name of the distributed update change object. When updating all the contents, the resultant level of the group object is indicated by the newlvl token of the UPD subtree in the name of the installed update change object. When elemental (in other words, child or grandchild) update change objects are installed, their installation does not effect the level of the parent group as a whole, but instead updates the level of the child or grandchild portion of the group object as indicated by the newlvl token.

The net effect of installing an update change object that raises the level of a group object or portion thereof to newlvl is the same as having installed a refresh change object for the same component with the level token of the REF subtree having the same value as the newlvl token of the UPD subtree.

In order for an update to change the level of a group object or some part of it, the existing level of whatever is being updated must be greater than or equal to the oldlvl token but less than the newlvl token of the UPD subtree specified for the update change object.
Subtrees for Fix Change Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C'FIX'</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered numeric</td>
<td>Fixlvl: The maintenance level to which this fix applies</td>
</tr>
<tr>
<td>3</td>
<td>Must match, not generable, not subtree ID, unordered - type unspecified</td>
<td>Fixid: The fix identifier</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned FIX subtree to be used in distributed fix change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The FIX subtree identifier token is registered by SNA/File Services.

The FIX subtree is used in the fix type of distributed change objects. These objects fix a specific problem effecting all or some of the contents of a group object, depending upon the granularity of the component name in the name of the distributed fix change object. The level of a group object or its affected portion is never modified by the installation of a fix change object, regardless of whether it was installed as a group or elemental (in other words, child or grandchild) change.

Fixes do not delete previous related change entries from the SNA/File Services catalog. In order to fix a problem effecting a group object or some part of it, the fix identifier (for example, the PTF number) must be specified by the fixid token of the FIX subtree and the existing level of whatever is being fixed must be specified in the fixlvl token.
# Subtrees for Library Change Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C'LIB'</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>Library Name</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, not subtree ID, ordered character – format VnRnMx, where n is 0-9, x is 0-9, A-Z, optional</td>
<td>Target Release (optional)</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned LIB subtree to be used in distributed *library* change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The LIB subtree identifier token is registered by SNA/File Services.
Subtree for File Member Change Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C'MEM'</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>Library Name</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>File Name</td>
</tr>
</tbody>
</table>
| 4            | Need not match, not generable, not subtree ID, ordered character | Member Name
See Note 1 |
| 5            | Need not match, not generable, not subtree ID, ordered character – format VnRnMx, where n is 0-9, x is 0-9, A-Z, optional | Target Release (optional)
See Note 2 |

Notes:

1. When this token is left blank, the fetching match flags must be included with the corresponding match flag containing the select high (*HIGHEST) or select low (*LOWEST) value. Select high means the last member in the file. Select low means the first member in the file.

2. This token should only be specified when the data object class is for AS/400 save/restorable packaged members ('2040E200'X). It will be ignored for any other data object class.

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned MEM subtree to be used in distributed file member change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The MEM subtree identifier token is registered by SNA/File Services.
Structured Names

Subtree for AS/400 Save Restorable Object Type Change Objects

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, subtree ID, unordered - type unspecified</td>
<td>C‘OBJ’</td>
</tr>
<tr>
<td>2</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>Library Name</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>Object Name</td>
</tr>
<tr>
<td>4</td>
<td>Need not match, not generable, not subtree ID, ordered character</td>
<td>Object Type</td>
</tr>
<tr>
<td>5</td>
<td>Need not match, not generable, not subtree ID, ordered character – format VnRnMx, where n is 0-9, x is 0-9, A-Z, optional</td>
<td>Target Release (optional)</td>
</tr>
</tbody>
</table>

For enterprise-structured names representing change (in other words, to be consolidated) objects, this is the definition of the SNA/Management Services owned OBJ subtree to be used in distributed AS/400 save restorable object type change objects intended to operate on SNA/Management Services created group software, microcode, and data objects. The OBJ subtree identifier token is registered by SNA/File Services.

Structured Names

Structured names have a format that is predefined by SNA/Management Services, and always contain a data object class descriptor as their first token. As microcode and microcode customization data are the only resource types that continue to use structured names, the first token of a structured name is either MCODE or MCUST.
Table 85 shows the tokens to use for microcode resources that use structured names:

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, type unspecified</td>
<td>C'MCODE'</td>
</tr>
<tr>
<td>2</td>
<td>Must match, not generable, type unspecified</td>
<td>Machine type</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, type unspecified</td>
<td>Model number. If none assigned or if this change is for all model numbers, C'NA'</td>
</tr>
<tr>
<td>4</td>
<td>Need not match, not generable, type unspecified</td>
<td>One of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'PATCH'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'MCF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'SUFFIXEC'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'MAINTEC'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'FUNCTEC'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- C'FEATURE'</td>
</tr>
<tr>
<td>5</td>
<td>Need not match, not generable, type unspecified</td>
<td>Microcode change name (for example, patch or MCF number, EC level)</td>
</tr>
<tr>
<td>6 - 10</td>
<td>Product specifiable</td>
<td>Product specifiable</td>
</tr>
</tbody>
</table>

Table 86 shows the tokens to use for microcode customization resources that use structured names:

<table>
<thead>
<tr>
<th>Token number</th>
<th>Token Attributes (assigned when file is created)</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Must match, not generable, type unspecified</td>
<td>C'MCUST'</td>
</tr>
<tr>
<td>2</td>
<td>Must match, not generable, type unspecified</td>
<td>Machine type</td>
</tr>
<tr>
<td>3</td>
<td>Need not match, not generable, type unspecified</td>
<td>Model number. If none assigned or if this change is for all model numbers, C'NA'</td>
</tr>
<tr>
<td>4</td>
<td>Need not match, not generable, type is NETID</td>
<td>NETID portion of location name of node for which this customizing data was created. If not applicable, C'NA'</td>
</tr>
<tr>
<td>5</td>
<td>Need not match, not generable, type is LUNAME</td>
<td>LUNAME portion of location name of node for which this customizing data was created. If not applicable, C'NA'</td>
</tr>
<tr>
<td>6</td>
<td>Need not match, not generable, type unspecified</td>
<td>Product defined, but must be specified to ensure uniqueness</td>
</tr>
<tr>
<td>7 - 10</td>
<td>Product specifiable</td>
<td>Product specifiable</td>
</tr>
</tbody>
</table>
### Table 87. NetView DM for MVS Resource Types and SNA/File Services Class Codes

<table>
<thead>
<tr>
<th>Resource Type Name</th>
<th>Resource Type Code</th>
<th>Further Description</th>
<th>SNA/FS Class Code 1</th>
<th>SNA/FS Class Code 2</th>
<th>SNA/FS Class Code 3</th>
<th>SNA/FS Class Code 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcode</td>
<td>0060</td>
<td></td>
<td>10</td>
<td>10</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Microcode customization</td>
<td>0060</td>
<td></td>
<td>10</td>
<td>20</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Software</td>
<td>0070 Unspecified</td>
<td></td>
<td>10</td>
<td>30</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Software</td>
<td>0070 AS/400 product</td>
<td></td>
<td>10</td>
<td>30</td>
<td>E2</td>
<td>01</td>
</tr>
<tr>
<td>Software</td>
<td>0070 AS/400 PTF</td>
<td></td>
<td>10</td>
<td>30</td>
<td>E2</td>
<td>02</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080 Unspecified</td>
<td></td>
<td>10</td>
<td>50</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080 Command list</td>
<td></td>
<td>10</td>
<td>50</td>
<td>20</td>
<td>00</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080 SAA REXX procedure</td>
<td></td>
<td>10</td>
<td>50</td>
<td>50</td>
<td>00</td>
</tr>
<tr>
<td>Procedure</td>
<td>0080 AS/400 program</td>
<td></td>
<td>10</td>
<td>50</td>
<td>E2</td>
<td>01</td>
</tr>
<tr>
<td>Flat data</td>
<td>0100</td>
<td></td>
<td>20</td>
<td>10</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Relational data</td>
<td>0120</td>
<td></td>
<td>20</td>
<td>20</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>AS/400 object</td>
<td>0156</td>
<td></td>
<td>20</td>
<td>40</td>
<td>E2</td>
<td>00</td>
</tr>
<tr>
<td>AS/400 data object</td>
<td>0158</td>
<td>containment structure</td>
<td>E0</td>
<td>E2</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Dump</td>
<td>0230</td>
<td></td>
<td>40</td>
<td>10</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Configuration file</td>
<td>0220</td>
<td></td>
<td>40</td>
<td>20</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Trace</td>
<td>0240</td>
<td></td>
<td>40</td>
<td>30</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Error log</td>
<td>0250</td>
<td></td>
<td>40</td>
<td>40</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>
Character Sets

Table 88 contains information about the subset of character set 1134, used in the routing group name (RGN) and routing element name (REN) operands when you define a node using either GIX or Batch Utilities (see “RGN and REN” on page 40 and 219).

<table>
<thead>
<tr>
<th>EBCDIC Code</th>
<th>ASCII Code</th>
<th>Graphic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>41</td>
<td>A</td>
<td>A, Capital</td>
</tr>
<tr>
<td>C2</td>
<td>42</td>
<td>B</td>
<td>B, Capital</td>
</tr>
<tr>
<td>C3</td>
<td>43</td>
<td>C</td>
<td>C, Capital</td>
</tr>
<tr>
<td>C4</td>
<td>44</td>
<td>D</td>
<td>D, Capital</td>
</tr>
<tr>
<td>C5</td>
<td>45</td>
<td>E</td>
<td>E, Capital</td>
</tr>
<tr>
<td>C6</td>
<td>46</td>
<td>F</td>
<td>F, Capital</td>
</tr>
<tr>
<td>C7</td>
<td>47</td>
<td>G</td>
<td>G, Capital</td>
</tr>
<tr>
<td>C8</td>
<td>48</td>
<td>H</td>
<td>H, Capital</td>
</tr>
<tr>
<td>C9</td>
<td>49</td>
<td>I</td>
<td>I, Capital</td>
</tr>
<tr>
<td>D1</td>
<td>4A</td>
<td>J</td>
<td>J, Capital</td>
</tr>
<tr>
<td>D2</td>
<td>4B</td>
<td>K</td>
<td>K, Capital</td>
</tr>
<tr>
<td>D3</td>
<td>4C</td>
<td>L</td>
<td>L, Capital</td>
</tr>
<tr>
<td>D4</td>
<td>4D</td>
<td>M</td>
<td>M, Capital</td>
</tr>
<tr>
<td>D5</td>
<td>4E</td>
<td>N</td>
<td>N, Capital</td>
</tr>
<tr>
<td>D6</td>
<td>4F</td>
<td>O</td>
<td>O, Capital</td>
</tr>
<tr>
<td>D7</td>
<td>50</td>
<td>P</td>
<td>P, Capital</td>
</tr>
<tr>
<td>D8</td>
<td>51</td>
<td>Q</td>
<td>Q, Capital</td>
</tr>
<tr>
<td>D9</td>
<td>52</td>
<td>R</td>
<td>R, Capital</td>
</tr>
<tr>
<td>DF</td>
<td>98</td>
<td>ý</td>
<td>y Diaeresis, Small</td>
</tr>
<tr>
<td>E2</td>
<td>53</td>
<td>S</td>
<td>S, Capital</td>
</tr>
<tr>
<td>E3</td>
<td>54</td>
<td>T</td>
<td>T, Capital</td>
</tr>
<tr>
<td>E4</td>
<td>55</td>
<td>U</td>
<td>U, Capital</td>
</tr>
<tr>
<td>E5</td>
<td>56</td>
<td>V</td>
<td>V, Capital</td>
</tr>
<tr>
<td>E6</td>
<td>57</td>
<td>W</td>
<td>W, Capital</td>
</tr>
<tr>
<td>E7</td>
<td>58</td>
<td>X</td>
<td>X, Capital</td>
</tr>
<tr>
<td>E8</td>
<td>59</td>
<td>Y</td>
<td>Y, Capital</td>
</tr>
<tr>
<td>E9</td>
<td>5A</td>
<td>Z</td>
<td>Z, Capital</td>
</tr>
<tr>
<td>F0</td>
<td>30</td>
<td>0</td>
<td>Zero</td>
</tr>
<tr>
<td>F1</td>
<td>31</td>
<td>1</td>
<td>One</td>
</tr>
<tr>
<td>F2</td>
<td>32</td>
<td>2</td>
<td>Two</td>
</tr>
<tr>
<td>F3</td>
<td>33</td>
<td>3</td>
<td>Three</td>
</tr>
<tr>
<td>F4</td>
<td>34</td>
<td>4</td>
<td>Four</td>
</tr>
<tr>
<td>F5</td>
<td>35</td>
<td>5</td>
<td>Five</td>
</tr>
<tr>
<td>F6</td>
<td>36</td>
<td>6</td>
<td>Six</td>
</tr>
<tr>
<td>F7</td>
<td>37</td>
<td>7</td>
<td>Seven</td>
</tr>
<tr>
<td>F8</td>
<td>38</td>
<td>8</td>
<td>Eight</td>
</tr>
<tr>
<td>F9</td>
<td>39</td>
<td>9</td>
<td>Nine</td>
</tr>
</tbody>
</table>

**Note:** The EBCDIC code values are based on IBM code page 500. The ASCII code values are based on IBM code page 850.
Bibliography

This bibliography contains a list of related documentation for users who want additional information.

TME 10 Software Distribution Version 3.1.5 Publications

The following hardcopy publications are available with TME 10 Software Distribution Version 3.1.5:

- TME 10 Software Distribution Version 3.1.5 for AIX Quick Beginnings, SH19-4333
- TME 10 Software Distribution Version 3.1.5 for OS/2 Quick Beginnings, SH19-4334
- TME 10 Software Distribution Version 3.1.5 for Windows NT Quick Beginnings, SH19-4335
- TME 10 Software Distribution Version 3.1.5 Clients Installation and Configuration SH19-4337
- TME 10 Software Distribution Version 3.1.5 for NetWare Quick Beginnings SH19-4341
- TME 10 Software Distribution Version 3.1.5 for NetWare Command Reference SH19-4342

The following publications are available with TME 10 Software Distribution Version 3.1.5 only in softcopy format:

- TME 10 Software Distribution Version 3.1.5 for AIX Command Reference
- TME 10 Software Distribution Version 3.1.5 for AIX Installation Scenarios
- TME 10 Software Distribution Version 3.1.5 Command Reference
- TME 10 Software Distribution Version 3.1.5 Messages Reference
- TME 10 Software Distribution Version 3.1.5 Pristine and Migration Scenarios

OS/390 Publications

Current documentation about OS/390 can be found on the World Wide Web at http://www.s390.ibm.com/os390/

NetView Distribution Manager/2 Publications

The following publications are available with NetView DM/2 Version 2 Release 1:

- NetView Distribution Manager/2 Concepts and Overview, GH19-4009, is an introduction to NetView DM/2 and how it is used for change control and data distribution.
- NetView Distribution Manager/2 Getting Started, SH19-4067, is a tutorial containing exercises to install and use NetView DM/2.
- NetView Distribution Manager/2 Installation and Customization Guide, SH19-6915, describes how to install and customize NetView DM/2. It outlines the operations that need to be done at host systems running NetView DM for MVS so that they can cooperate with workstations running NetView DM/2.
- NetView Distribution Manager/2 Change Distribution Manager User's Guide, SH19-5048, describes how to use the Change Distribution Manager (CDM).
- NetView Distribution Manager/2 Messages and Error Recovery Guide, SH19-6924, describes how to diagnose problems and recover from errors when installing or using NetView DM/2.

NetView Distribution Manager/6000 Publications

The following publications are available with NetView DM/6000 Release 1.2:

- NetView Distribution Manager/6000 Concepts and Overview, GH19-5001
- NetView Distribution Manager/6000 Getting Started, SH19-4149, is a tutorial containing exercises to install and use NetView DM/6000.
- NetView Distribution Manager/6000 Messages and Error Recovery Guide, SH19-5004, describes normal operational messages and the error
messages that can be produced when using NetView DM/6000.

AS/400 Publications

The following publications may be of use for the AS/400 environment:

- SystemView Managed System Services/400 User's Guide, SC41-0142

MVS Publications

The following master indexes list the publications for MVS:

- MVS/ESA Master Index for MVS/ESA Version 5, GC28-1425

VTAM Publications

The following section lists publications for VTAM Version 4:

- Planning for NetView, NCP, and VTAM, SC31-8063
- Planning for Integrated Networks, SC31-8062
- VTAM V4R3 for MVS/ESA Operation, SC31-6549
- VTAM V4R3 for MVS/ESA Installation and Migration Guide, GC31-6547
- VTAM V4R3 for MVS/ESA Network Implementation Guide, SC31-6548
- VTAM V4R3 for MVS/ESA Operation Quick Reference, SX75-0207
- VTAM V4R3 for MVS/ESA Programming, SC31-6550
- VTAM V4R3 for MVS/ESA Messages and Codes, SC31-6546
- VTAM V4R3 for MVS/ESA Programming for LU 6.2, SC31-6551
- VTAM V4R3 for MVS/ESA Resource Definition Reference, SC31-6552
- VTAM V4R3 for MVS/ESA Resource Definition Samples, SC31-6554
- Estimating Storage for VTAM V4R3, SK2T-6400, a diskette
- VTAM V4R3 for MVS/ESA Glossary, GC31-6556
- VTAM V4R3 for MVS/ESA CMIP Services and Topology Agent Programming Guide, SC31-6544
- VTAM AnyNet Feature: Guide to Sockets over SNA, SC31-6559

NCP Publications

The following section lists publications for NCP Version 7:

- Planning for NetView, NCP, and VTAM, SC31-8063
- Planning for Integrated Networks, SC31-8062
- NCP V7R3, SSP V4R3, and EP R12 Library Directory, SC30-3832
- NCP V7R3, SSP V4R3, and EP R12 Generation and Loading, SC31-6221
- NCP V7R4 X.25 V2R8 Planning and Installation, SC30-3470
- NCP V7R5 Migration Guide, SC30-3833
- NCP, SSP, and EP Resource Definition Reference, SC31-6224
- NCP, SSP, and EP Resource Definition Guide, SC31-6223
- NCP, SSP, and EP Resource Definition Reference, SC31-6224
- NCP V7R5, SSP V4R5, and EP R12 Messages and Codes, SC31-6222

ISPF Publications

The following section lists publications for ISPF Version 4:

- ISPF Getting Started, SC34-4440
- ISPF Dialog Developer's Guide and Reference, SC34-4486
- ISPF Services Guide, SC34-4485
- ISPF User's Guide, SC34-4484
- ISPF Dialog Tag Language Guide and Reference, SC34-4441

Related Publications

The following list shows other related publications that may be useful:

- AIX NetView DM and LPS, GC31-7008
- AIX SNA Server/6000 User's Guide, SC31-7002
- Centralized Files Distribution Using NetView DM for MVS and PC Node Manager, GC24-3459
• DB2 for MVS/ESA Administration Guide, SC26-3265
• DB2 for MVS/ESA Messages and Codes, SC26-3268
• DB2 for MVS/ESA SQL Reference, SC26-3270
• DFSMS/MVS Access Method Services for ICF, SC26-4906
• DFSMS/MVS Utilities, SC26-4926
• DPPX/370 General Information, GC23-0640
• IBM 3174 Subsystem Control Unit Central Site Customization User's Guide, GA23-0342
• IBM 3174 Establishment Controller Central Site Customization Guide - Configuration Support B, GA27-3868
• IBM 4680 Store System Programming Guide, SC30-3517
• IBM 4690 Store System Communication Programming Reference, SC30-3582
• LAN Distributed Platform Licensed Programs Family Programmer's Reference, SC19-5320
• NetView General Information Manual, GC31-7098
• OS/390 MVS JCL User's Guide, GC28-1758
• OS/390 MVS JCL Reference, GC28-1757
• OS/390 SMP/E User's Guide, SC28-1740
• OS/390 SMP/E Reference, SC28-1806
• OS/390 TSO/E Command Reference, SC28-1969
• Personal Computer/Distributed Systems Node Executive Installation and Operation, SH19-6585
• S/36 Communications and Systems Management Guide, SC21-8010
• SNA Technical Overview, GC30-3073
• SNA Formats, GA27-3136
• SNA Management Services Reference, SC30-3346
• SNA Distribution Services Reference, FS1 & FS2, SC30-3098
• SNA Distribution Services Reference, FS1 only, ST00-4665
• SNA File Services Reference, SC31-6807
• VSE/DSNX Planning Installation Operation, SC33-6210
**Glossary**

This glossary defines important terms and abbreviations for NetView DM for MVS and related IBM products. It includes terms and definitions from the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994, which defines many additional data processing terms.

It also includes terms and definitions from:


Definitions are identified by the symbol (A) after the definition.

- The Information Technology Vocabulary, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1). Definitions from published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

**Reference Words Used in the Entries:**

The following reference words may be used in this glossary:

- **Contrast with.** Refers to a term that has an opposed or substantively different meaning.
- **Deprecated term for.** Indicates that the term should not be used. It refers to a preferred term, which is defined.
- **See.** Refers to multiple-word terms that have the same last word.
- **See also.** Refers to related terms that have similar (but not synonymous) meanings.
- **Synonym for.** Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.
- **Synonymous with.** Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

**A**

**ACF/VTAM.** Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for VTAM.

**allocate.** A logical unit (LU) 6.2 application program interface (API) verb used to assign a session to a conversation for the conversation to use. Contrast with deallocate.

**AS/400 data object containment structure.** A NetView DM for MVS CMEP resource type.

**AS/400 object.** A NetView DM for MVS CMEP resource type.

**B**

**batch utilities.** A set of NetView DM for MVS programs that provide a batch interface to NetView DM for MVS facilities. Batch utilities can be invoked by application programs or using JCL, and allow the user to:

- Define nodes and resources to NetView DM for MVS
- Assign resources to nodes
- Manage data objects in the resource repository
- Prepare and submit transmission plans
- Maintain the NetView DM for MVS files

**C**

**category.** The logical organization of SPMF groups. This is the highest level of grouping provided by SPMF and allows groups with common characteristics to be associated with each other.

**CC client.** Change control client

**CC server.** Change control server

**change control.** The control of software and data at a target, by means of commands such as Install, Remove, Accept, Uninstall, and Activate.

**change control (CC) server.** A workstation that controls and tracks the distribution of software and data files to other workstations (its change control clients). In a NetView DM/2 environment, a CC server and all its CC clients must be on the same LAN. In a NetView DM/6000 environment, a CC server and all its CC clients must be in the same TCP/IP network. In a
NetView DM for NetWare environment, a CC server and all its CC clients must be in the same IPX network.

change file. All of the related files needed for a given change to an application or operating system. The change file is a structured file, packaged at a preparation site, which can contain data files, procedures, parameters, and lists.

change management. The area of activity concerned with centrally managed changes to the program code and data made available in a network. Change management is done by applying to resources commands such as Install, Remove, Accept, Uninstall, and Activate.

Change Management Entry Point (CMEP). An SNA node that provides change management support. It can be a type 2, type 2.1, type 4, or type 5 node. It sends SNA formatted change management data about itself and the resources it controls to a focal point for centralized processing. It receives and executes focal point initiated requests to manage and control its resources.

Change Management Focal Point (CMFP). An entry point that provides centralized management and control for other entry points within its sphere of control for change management.

cluster. (1) In VSAM, a named structure consisting of a group of related components, for example, a data component with its index component. (2) A station that consists of a control unit (cluster controller) and the terminal attached to it. In NetView DM for MVS, the term node is used rather than cluster.

CMACTUSE. (1) Type of activation requested by the NetView DM for MVS host to an entry point. It is an attribute of the activate node control statement that specifies which components altered by changes will be used during activation.

command list. (1) A list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in NetView command list language. For nodes supported by NetView DM for MVS, command lists are written in a language that can be understood by the node to which the command is destined. (2) A NetView DM for MVS resource type (CLIST)

communications and systems management (C&SM). A feature of the System/36 SSP that contains the change management support (called DSNX) that allows System/36 to communicate with NetView DM.

compression. The process of condensing the representation of data by replacing a string of repetitive characters with a number that indicates how many characters were eliminated. Contrast with decompression.

conditioning. In NetView DM for MVS, to make the execution of a phase in a plan dependent on the successful or unsuccessful completion of a previous phase in the plan.

configuration file. A NetView DM for MVS resource type.

convert. (1) In NetView DM for MVS, to transfer a data set from the resource repository to a user data set at the host. (2) To change the representation of data from one form to another, without changing the information it conveys. (A) (I)

D

data set. (1) The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. (2) A NetView DM for MVS resource type.

DBCS code. The hexadecimal code, 2 bytes in length, that identifies a double-byte character.

deallocate. An LU 6.2 application program interface (API) verb that terminates a conversation, thereby freeing the session for a future conversation. Contrast with allocate.

decompression. The process of restoring compressed data to its original form. Contrast with compression.

destination. In NetView DM for MVS transmissions, the node to which a resource is being sent.

Distributed Processing Program Executive (DPPX/370). An operating system of the System/370 Information System.

distributed resource directory (DRD). A NetView DM for MVS system file containing the description of the NetView DM for MVS network, its users, the defined resources, and their status tracking records.

Distributed Systems Node Executive. A licensed program at the VSE node (VSE/DSNX) or a program feature at the IBM System/36 SSP node enabling communication with NetView DM for MVS at the host system.

distribution and change control application programming interface. A feature of NetView DM for MVS that allows an application to interact with the transmission control program, providing most of the
**double-byte character set (DBCS)**. A set of characters in which each character is represented by a 2-byte code. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, the typing, display, and printing of DBCS characters requires hardware and programs that support DBCS. Contrast with **single-byte character set**.

**DPPX/370**. Distributed Processing Program Executive/370.

**E**

**entry point**. Any distributed system whose software, microcode, customizing data, user data, updates for each of them, and maintenance information are controlled and tracked by a focal point. While entry points can initiate distributions of objects to their peers and to the focal point, remote change management functions can be initiated only by the focal point. An entry point is often referred to simply as a **node** or **end node**.

**error log**. A NetView DM for MVS CMEP resource type.

**errorlog**. A DPPX resource type. It is a data set used to record information about certain hardware and programming events.

**exit routine**. Any of several types of special-purpose user-written routines.

**F**

**fanout**. (1) A modem feature that permits up to four controllers to be attached to one modem. See also **tailing**. (2) In NetView DM for MVS multitiered networks, the capability of the host system to send a single copy of the resource to the front-end node, which then takes care of distributing copies to the end nodes linked to it, according to the distribution headers sent by NetView DM for MVS.

**flat data**. A NetView DM for MVS resource type.

**focal point**. An entry point that provides centralized management and control for other entry points for one or more network management categories.

**function**. In NetView DM for MVS, the specification of a transmission activity on a resource or group of resources. Functions are grouped into phases. See also **phase** and **plan**.

**G**

**generalized interactive executive (GIX)**. That part of NetView DM for MVS that provides the user of the host system with interactive access to NetView DM for MVS functions.

**Generalized Trace Facility (GTF)**. An optional MVS service program that records significant system events (such as supervisor calls and start I/O operations) for the purpose of problem determination.

**global name**. The name by which a resource is uniquely identified within NetView DM for MVS. It is constructed from text tokens, separated by periods. The number of tokens is in the range 2-10 and each token can be from 1 to 16 characters long. The NetView DM for MVS product family limits the maximum length of the name to 64 characters including periods.

**H**

**hold**. In NetView DM for MVS, a command to suspend the processing of a scheduled phase or to suspend the transmission to one or more nodes. Contrast with **release**.

**holding file**. A NetView DM for MVS system file containing resources that are ready to be sent to the nodes and resources that have been received from the nodes. See also **resource repository**.

**I**

**internetwork packet exchange (IPX)**. A Novell** communication protocol that creates, maintains, and terminates connections between network devices (workstations, file servers, or routers, for example).

**interactive operator facility (IOF)**. That part of NetView DM for MVS that allows the NetView DM for MVS control operator to monitor the progress of transmissions and intervene as necessary.

**Interactive System Productivity Facility (ISPF)**. An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard screen panels and interactive dialogs between the application programmer and the terminal user.

**intermediate node**. In a NetView DM for MVS multitiered network, SSP nodes connecting end nodes to the host system. The intermediate node that is directly connected to NetView DM for MVS performing
fanout distribution of a resource is called the front-end node.

K

kanji. An ideographic character set used in Japanese. See also double-byte character set.


data set. A set of data records.

katakana. A character set of symbols used in one of the two common Japanese phonetic alphabets, which is used primarily to write foreign words phonetically. See also kanji.

key-sequenced data set (KSDS). A VSAM file or data set whose records are loaded in key sequence and controlled by an index.

L

library. A VSE/DSNX resource type.

local administrator. A person who works at a workstation and is responsible for controlling distribution and maintenance of software packages and data files across a domain that consists of (1) a CC server and its CC clients or (2) one CC single node.

local area network (LAN). (1) A network in which a set of devices are connected to one another for communication, and which can be connected to a larger network. See also token ring. (2) A computer network located on a user’s premises within a limited geographical area. Contrast with wide area network.

logical unit (LU). In SNA, a type of network addressable unit that enables end users to gain access to network resources and communicates with each other.

logical unit name. The LU name identifying a node in the network.

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also session parameters.

LU 6.2. Logical unit 6.2.

M

member. A VSE/DSNX resource type.

message. (1) A group of characters and control bit sequences transferred as an entity. (T) (2) In VTAM, the amount of function management data (FMD) transferred to VTAM by the application program with one SEND request. (3) In NetView DM for MVS, a resource type.

node. (1) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. (2) In VTAM, a point in a network defined by a symbolic name. (3) In NetView DM for MVS, the points in the network with which NetView DM for MVS communicates.

node name. (1) In VTAM, the symbolic name assigned to a specific major or minor node during network definition. (2) In NetView DM for MVS, the symbolic name assigned to an end node during the definition of the nodes managed by NetView DM for MVS.

node software profile. A combination of software packages that satisfies the business needs for a set of workstation users within an enterprise. For SPMF, the node software profile is a combination of software functional packages.

node type. In SNA, a designation of a node according to the protocols it supports or the role it plays in a network. Node type was originally denoted numerically (as 1 2.0, 2.1, 4, and 5) but is now characterized more specifically by protocol type (APPN network node, subarea node, and interchange node, for example)
because type 2.1 nodes and type 5 nodes support multiple protocol types and roles.

**NSP.** Node software profile.

**O**

**object.** The unit of distribution in a network controlled by the NetView DM product family to manage software distribution and changes. It can contain any kind of software or user data.

**origin.** In NetView DM for MVS, the end point where the resource came from.

**P**

**package.** A user-defined collection of related change files required to maintain an application or system software product. Typically, the package contains all files necessary to run an application or product; it may contain updates and fixes as well as the base. Also called a software functional package (SFP).

**partial name.** Identifies a group of objects the names of which start with the same characters. A partial name is made up of one or more characters followed by an asterisk (*). The characters identify the common portion of the names of the objects, while the asterisk indicates that from that point on the names may differ.

**phase.** In NetView DM for MVS, a structured description of one or more functions addressing a node or a group of nodes in the network. Phases are grouped into plans.

**plan.** In NetView DM for MVS, a structured description of one or more phases logically grouped together for validation, submission, and monitoring. Each plan contains one or more phases, and each phase contains one or more functions. See also phase and function.

**plan library.** In NetView DM for MVS, the repository where all transmission plans are stored.

**procedure.** A NetView DM for MVS resource type.

**profile.** A combination of software packages that satisfies the business needs for a set of workstation users within an enterprise. For SPMF, the node software profile is a combination of software functional packages. Also called a node software profile (NSP).

**program.** (1) A set of actions or instructions that a machine is capable of interpreting and processing. (2) A NetView DM for MVS resource type.

**program temporary fix (PTF).** A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

**R**

**relational data.** A NetView DM for MVS resource type.

**release.** In NetView DM for MVS, a command to remove a phase from a hold status. Contrast with hold.

**remote administrator.** A person who works at a workstation and is responsible for controlling distribution and maintenance of software packages and data files across a domain that can include multiple CC servers and their clients, and multiple CC single nodes. In NetView DM/2 only, a remote administrator’s workstation can, optionally, be configured so that all change control activities within the domain are reported to a focal point or to a higher-level remote administrator workstation.

**repository.** In NetView DM for MVS, the VSAM files where data resources (holding file) and software resources (NetView DM for MVS library) are stored.

**resource.** (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network. (3) In NetView DM for MVS, all the devices, features, data and software belonging to a node.

**resource class.** In NetView DM for MVS, an optional modifier that can be associated with each resource. Users can be authorized to use only certain classes of resource.

**resource repository.** In NetView DM for MVS, the VSAM files where data resources (holding file) and software resources (NetView DM for MVS library) are stored.

**resynchronization.** In NetView DM for MVS, restarting the transmission of a function at the point where it was interrupted.

**retrieve function.** A NetView DM for MVS function that applies to all node types.

**routing element name (REN).** Part of a distribution services unit (DSU) identifier, according to SNA/DS architecture.

**routing group name (RGN).** Part of a distribution services unit (DSU) identifier, according to SNA/DS architecture.
**scheduling**. In NetView DM for MVS, the process of requesting that the transmission of a phase should be started at a particular time of day or upon the successful or unsuccessful transmission of a previous phase.

**sequence**. In NetView DM for MVS, the grouping of functions within a phase.

**service provider**. A NetView DM for MVS component that acts as a VSAM interface between the NetView DM for MVS components (GIX, batch utilities, and the TCP) and the NetView DM for MVS distributed resource directory (DRD), transmission control file (TCF), holding file, and library files. It runs in a separate address space in order to improve I/O performance.

**session**. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session.

**SFP**. software functional package

**shared**. Pertaining to the availability of a resource to more than one user at the same time.

**single-byte character set (SBCS)**. A character set in which each character is represented by a one-byte code. Contrast with double-byte character set.

**SNA/Distribution Services (SNA/DS)**. A connectionless communications service that distributes objects over a network of LU 6.2 connections.

**SNA/DS**. SNA/Distribution Services

**SNA/File Services (SNA/FS)**. A service that allows files to be fetched, moved, and stored at nodes in an SNA/DS network. SNA/FS provides name structure and version identification mechanisms that uniquely identify files in a network.

**SNA/FS**. SNA/File Services

**SNA/Management Services (SNA/MS)**. The services provided to assist in the management of SNA networks.

**software functional package**. A user-defined collection of related change files required to maintain an application or system software product. Typically, the software functional package contains all files necessary to run an application or product; it may contain updates and fixes as well as the base.

**Software Profile Management Facility (SPMF)**. A feature of NetView DM for MVS that allows software distribution based on workstation profiles.

**submit**. In NetView DM for MVS, to send a plan to the transmission control file (TCF) for subsequent execution by the TCP. When a transmission plan is submitted, it is automatically validated before being written in the TCF.

**Systems Network Architecture (SNA)**. A description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

**T**

**TCP/IP**. Transmission control protocol/internet protocol.

**transmission**. The sending of data from one place for receipt elsewhere. (A) (Transmission implies only the sending of data; the data may or may not be received. The term transmit is used to describe the sending of data in telecommunication operations. The terms move and transfer are used to describe movement of data in data processing operations.)

In NetView DM for MVS, transmission is also used to indicate both sending and receiving.

**transmission control file (TCF)**. A NetView DM for MVS system file that contains the plans describing the work to be done by the transmission control program (TCP). It also contains the results of executed plans.

**transmission control program (TCP)**. That part of NetView DM for MVS that controls the sending and receiving of all data to and from the various nodes in the network.

**Transmission Control Protocol/Internet Protocol**. A set of communications protocols that support peer-to-peer connectivity functions for both local and wide area networks.

**transmission plan**. In NetView DM for MVS, a structured description of one or more phases logically grouped together for validation, submission, and monitoring. Each plan contains one or more phases, and each phase contains one or more functions. See also phase and function.

**transmission profile (TP)**. A set of logical units that have the same type of connection with NetView DM for MVS, and some other common characteristics.
transmission window. In NetView DM for MVS, a user-defined time window within which transmission activities for a phase are to take place.

U

uninstall. To remove completely a component from the hard disk of a target.

V

validate. In NetView DM for MVS, to examine a plan in the plan library and check whether the various scheduling criteria and limitations have been followed.


VTAM. Virtual Telecommunications Access Method.

W

wide area network (WAN). A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities. (T) Contrast with local area network.

window. (1) In SNA, synonym for pacing group. (2) On a visual display terminal, a small amount of information in a framed-in area on a panel that overlays part of the panel. (3) In data communication, the number of data packets a data terminal equipment (DTE) or data circuit-terminating equipment (DCE) can send across a logical channel before waiting for authorization to send another data packet. The window is the main mechanism of pacing, or flow control, of packets. (4) In NetView DM for MVS, synonym for transmission window.
**Index**

### A
- accept if no error 163, 404
- ACCEPT MICROCODE statement (SUBMIT) 346
- ACCEPT RESOURCE statement (SUBMIT) 347
- acceptance 94
- ACTIVATE NODE statement (SUBMIT) 349
- activation 163, 404
- activation use 161, 402
- ACTUSE operand (INSTALL MICROCODE) 360
- ACTUSE operand (INSTALL RESOURCE) 366
- ADD RESOURCE statement 254
- ADD statements (SUBMIT) 248—250
- agent report code 162, 403
- ALIAS operand (PREPARE DATASET/PRINT) 297
- ALLOCATED field 527
- ALLOWED field 527
- ALTERACTIVE operand
  - INSTALL MICROCODE 362
  - INSTALL RESOURCE 369
  - REMOVE MICROCODE 371
  - REMOVE RESOURCE 374
  - UNINSTALL COMPONENT 393
- appending resources 117
- AS/400 naming conventions 573
- ASSIGN LUCP statement (SUBMIT) 234
- assigning resources to nodes using batch 239
- assigning resources to nodes using GIX 85
- assignment 94
- attended mode 471
- attributes, transmission profile 46, 232
- auto accept 163, 404
- auto remove status 163, 404
- AUTOACCEPT operand (INSTALL MICROCODE) 359
- AUTOACCEPT operand (INSTALL RESOURCE) 366
- AUTOEND parameter 478
- automatic acceptance 162, 403
- automatic assignment, GIX field 73
- automatic definition 89
- automatic removal 162, 403
- automatically assign a resource 74
- AUTOREMOVAL operand (INSTALL MICROCODE) 360
- AUTOREMOVAL operand (INSTALL RESOURCE) 366
- AUTOSTR parameter 477
- AUTOSUB operand (DEFINE PLAN) 319

### B
- backup 94
- batch job name (a phase attribute) 130
- batch job submission conditioning value (a phase attribute) 130
- Batch Utilities
  - definition of 197
  - FORMAT 202
  - how to run 202
  - LIBMAINT 202
  - MAINTDAT 203

### Batch Utilities (continued)
- PLCOPY 208
- return codes 201
- SUBMIT 204
- TCFMAINT 206
- batch utility return codes 201
- block size 28
- browsing
  - a message 192
  - a plan or phase 124
  - a plan summary 168
  - a plan using GIX 123
  - logical unit to transmission profile assignments 49
  - node definitions 45
  - node groups 60
  - phase information 152
  - resource definitions 77
  - resource groups 83
  - the repository directory 117
- browsing assignment to connection profiles 52
- BYDESTID operand
- SEND/REPLACE statements 341
- UNINSTALL COMPONENT 393
- bytes transmitted 158, 161, 400, 402

### C
- CATALOG operand (PREPARE DATASET/PRINT) 297
- CATLG operand (RECEIVE/RETRIEVE statements) 335
- CATLG operand (SEND/REPLACE statements) 339
- CATPSW operand (PREPARE DATASET/PRINT) 297
- CATPSW operand (RECEIVE/RETRIEVE statements) 335
- CATPSW operand (SEND/REPLACE statements) 339
- CC operand (PREPARE DATASET/PRINT) 297
- CCSID 103, 263
- change CURTASK value 553
- CHANGE LUTP statement (SUBMIT) 235
- change management entry point (CMEP), definition of 34, 216
- change management focal point (CMFP), definition of 34, 216
- CHANGE NODE statement (SUBMIT) 217
- change the start time and date 504
- CHANGE USER statement (SUBMIT) 212
- changing
  - a node definition 43
  - a node group 59
  - a plan 140
  - a resource definition 76
  - a resource group 82
  - a resource history record 94
  - a transmission plan using batch 317
  - a transmission plan using GIX 126
  - a user profile 29
  - attributes of a resource assignment 89
  - curtask 524
  - GIX authorizations 21
  - logical unit to transmission profile assignments 47
  - node type authorizations 24
changing (continued)
password 21
resource authorizations 25
the network configuration 97
the transmission schedule 503
transmission profiles 521
character sets 585
CL operand (RECEIVE/RETRIEVE statements) 335
class (a resource attribute) 73
class codes (a resource identifier) 112
class codes, SNA/File Services 584
CLASS operand
CHANGE/DEFINE NODE statement 220
DEFINE RESOURCE 256
SEND/REPLACE statements 339
CLCODES operand
CHANGE/DEFINE NODE statement 220
DEFINE RESOURCE 256
SEND/REPLACE statements 339
CLISTs library 464
CMACTUSE operand
ACTIVATE NODE 349
CMD operand 286
CMFP
See change management entry point (CMEP), definition of
See change management focal point (CMFP), definition of
coded character set identifier (CCSID) 263
coding character set identifiers (CCSID) 103
coding the TCP 478
command execution 163, 403
command line 10
commands and validation errors data set 464
commands log, GIX 11
commands, GIX 11
  jump functions 14
  line commands on full-list panels 12
  primary commands 11
completed/deleted phases by node 152, 396
completeness (a resource identifier) 108
compression 115, 116, 290, 297, 335
COMPRS operand
RECEIVE/RETRIEVE statements 335
SEND/REPLACE statements 339
COMPRS operand (PREPARE DATASET/PRINT) 297
COMPRSD operand
LOAD RESOURCE 290
COND operand (DEFINE PHASE) 321
condition phases 514
conditioned phase, definition of 122, 315
conditioning
  between phases 122, 315
  within a phase 121, 315
conditioning between phases 122, 315
conditioning criteria (a phase attribute) 130
conditioning mode (a phase attribute) 130
conditioning phase (a phase attribute) 130
conditioning phase name 152, 396
conditioning phase, definition of 122, 315
conditioning return code 152, 396
conditioning value (a phase attribute) 130
conditioning within a phase 121, 315
CONDMODE operand (DEFINE PHASE) 322
CONDRC operand (DEFINE PHASE) 321
connection (a node attribute) 38, 219
CONNECTION operand
  CHANGE/DEFINE NODE statement 219
connection profiles 50, 232
CONTIG operand (PREPARE DATASET/PRINT) 297
control character (a resource attribute) 114
control statement labels 198
control statements
definition of 197
END 200
general purpose 209
how to define 198
keywords 199
labels 198
operands 199
underscores, note on usage 199
variables 199
verb and modifier 199
Controlling the TCP, MVS MODIFY Commands 550
controlling transmission requests 496
controlling transmission schedules 495
controlling transmission with nodes 515
CONV operand
  PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB 304
  PRINT RESOURCE 281
CONVERT DATASET statement (MAINTDAT) 284
CONVERT DUMP statement (MAINTDAT) 284
CONVERT RESOURCE statement (MAINTDAT) 269
converting resources 183, 267
converting resources from the repository 186
COPIES operand (SEND/REPLACE statements) 339
COPY statement 209
copying an existing phase 135
copying an existing phase, example 136
COREQUISITE operand (INSTALL MICROCODE) 360
COREQUISITE operand (INSTALL RESOURCE) 367
corequisites 162, 403
corpid, definition of 65, 243
creating
  a new phase or function 136
  a node group 58
  a resource history record 92
  a transmission plan using batch 317
  a transmission plan using GIX 126
CURTASK 524
CURTASK parameter 480
cut off (a phase attribute) 129
cut off-date for windows 396, 397
cut off-time for windows 396, 398
cut off-time interval 152, 154, 396, 398
CUTINT operand (DEFINE PHASE) 322
cutoff date for transmission windows 323
cutoff time for transmission windows 323
CUTTIME operand (DEFINE PHASE) 322
CUTWDATE operand
  DEFINE PHASE 323
CUTWTIME operand
  DEFINE PHASE 323
D & CC API requests 471
D & CC API, setting up the 482
DASDCHECK operand
   INSTALL RESOURCE 365
   REMOVE RESOURCE 373
data already compressed (a resource attribute) 115
data set authorizations 250
data set name 188
data set name (a resource attribute) 113
data translation 185
DATATYPE operand (PREPARE DATASET/PRINT) 297
DATATYPE operand (RECEIVE/RETRIEVE statements) 335
DATE operand
   ACTIVATE NODE 349
   DEFINE PHASE 321
   INITIATE PROCEDURE 358
   INSTALL MICROCODE 362
   INSTALL RESOURCE 368
   RESET PLAN 419
   UNINSTALL COMPONENT 392
DB2
   publications xxi
   DB2 tables, referencing 477
DEASGN statements (SUBMIT) 251
DEASSIGN LUCP statement (SUBMIT) 236
DEASSIGN RESOURCE statement 255
DECOMPRS operand (SEND/REPLACE statements) 339
default node class 28
default plan model 123
default resource class 28
DEFINE GROUP statement (SUBMIT) 228
DEFINE NODE statement (SUBMIT) 217
DEFINE PHASE statement (SUBMIT) 321
DEFINE PLAN statement (SUBMIT) 319
DEFINE RESOURCE statement (SUBMIT) 256
DEFINE USER statement 214
DEFINE USER statement (SUBMIT) 212
defining
   a group name 57, 81
   a group of nodes 56
   a group of nodes using batch 228
   a group of resources using batch 228
   a plan name 126
   a resource history record using GIX 92
   a user identification 20
   control statements 198
   nodes using batch 215
   nodes using GIX 33, 35
   resources using batch 239
   resources using GIX 61, 69
termination values 139
   the attributes of a new plan 126
   the attributes of functions in a plan 132
   the node type 69, 79, 85
   the resource type 70, 80
   user profiles 19
   user profiles using Batch Utilities 211
DELAY operand (DEFINE PLAN) 319
delete a transmission request 506
DELETE DEFINITION statement (SUBMIT) 252
DELETE GROUP statement (SUBMIT) 228
DELETE HISTORY statement (SUBMIT) 257
DELETE MICROCODE statement (SUBMIT) 351
DELETE NODE statement (SUBMIT) 223
DELETE PLAN statement (SUBMIT) 324
DELETE RESOURCE statement (SUBMIT) 353
DELETE SPLAN statement (TCFMAINT) 409
DELETE statements (SUBMIT) 328—329
deleted by 150, 152, 395, 396
deleting
   a message 192
   a node definition 44
   a node group 59
   a resource assignment 90
   a resource definition 76
   a resource group 83
   a resource history record 95
   a transmission plan using GIX 141
   a user profile 30
   elements from a plan 139
   nodes using Batch Utilities 223
   resources from the repository 117
deleting plans and phases 163
   nonrecursive plans 164
   plan recursions 164
   recursive plans 164
   specific phase 165
   using GIX 165
   using the Delete at Server option 165, 410
deletion 94
DELMATCHIND operand (INSTALL MICROCODE) 360
DELMATCHIND operand (INSTALL RESOURCE) 367
DELMATCHIND operand (SEND MICROCODE) 383
DELMATCHIND operand (SEND RESOURCE) 387
DELNAME operand (INSTALL MICROCODE) 361
DELNAME operand (INSTALL RESOURCE) 384
DELNAME operand (SEND MICROCODE) 387
DELNAME operand (SEND RESOURCE) 387
destruction 162, 403
DESTRUCTION operand (INSTALL MICROCODE) 361
DESTRUCTION operand (INSTALL RESOURCE) 367
DESTRUCTION operand (SEND MICROCODE) 384
DESTRUCTION operand (SEND RESOURCE) 387
DESTYPE operand 285
detailed data 163, 404
directory name (a node attribute) 38, 219
DIRECTORY operand
   CHANGE/DEFINE NODE statement 38, 219
disconnect IOF (IOF command) 532
DISP operand
   LOAD RESOURCE 291
   PREPARE CLIST/JOB/PROGRAM 294
   PREPARE DATASET/PRINT 297
   PREPARE RESOURCE 278
   RECEIVE/RETRIEVE statements 335
   RETRIEVE RESOURCE 379
   SEND RESOURCE 386
display transmission for a node 552
displaying curtask 524
displaying transmission requests 502
disposition (a resource attribute) 189
DISPOSITION operand (INSTALL MICROCODE) 362
DISPOSITION operand (INSTALL RESOURCE) 369
DPPX resources, naming conventions for 571
DRD updated 154, 159, 398, 400
DSAC operand
  ADD statements 250
  PREPARE DATASET/PRINT 298
  SEND/REPLACE statements 339
DSORG operand (PREPARE DATASET/PRINT) 298
DSORG operand (RECEIVE/RETRIEVE statements) 335
DSXBHILF program 440
DSXBUBK 197, 442
DSXBURS 197, 442
DSXJOBS file 466
DSXPREP, operands 202
  FUNCTION 202
  INST 202
  LINCNT 202
  PASSWORD 202
  PGM 202
  PROFILE 203
  USERID 202
DUMP statement 209
DUPLICA PLANS statement (PLCOPY) 448
dynamic substitution 344

E
E (display conditioned phases) line command 514
editing CLIST and Job resources using ISPF/PDF 118
effect 162, 403
EJECT statement 209
EMBCNTRL operand
  LOAD RESOURCE 292
EMBCNTRL operand (LOAD RESOURCE) 292
embedded forms
  LOAD RESOURCE 292
  PREPARE DATASET 300
  PREPARE PRINT 300
end date/time 154, 398
END statement 200, 210
end transmission (IOF command) 530
ending transmission 478
enterprise-structured names, definition of 65, 243, 575
  entry point, definition of 34, 215
ERASE APIQENT statement (TCFMAINT) 412
ERASE operand (SAVE APIQENT) 420
ERASE operand (SAVE PLAN) 421
ERASE operand (SAVE REQUEST) 422
  erase option 168
  ERASE PLAN statement (TCFMAINT) 413
ERASE REQUEST statement (TCFMAINT) 414
ERASE RESOURCE statement (MAINTDAT) 273
eraser plans and phases 163
ERASPLAN job 423
errors
  batch utility return codes 201
  IOF return codes 512
  NetView DM for MVS host errors 181, 406
  node errors 182, 407
  TCP 153, 397
  transmission function termination return codes 181, 406
errors (continued)
  validation errors 142
  VTAM errors 182, 406
ERRPLAN operand (PRINT PLAN) 416
EXEC statement 474
EXECTIME operand (INITIATE statements) 331
execution 94
exiting from GIX 16
exiting from IOF 494
EXT operand
  CONVERT RESOURCE 270
  DELETE HISTORY 257
  DELETE RESOURCE 354
  ERASE RESOURCE 274
  PREPARE DATASET/PRINT 298
  PREPARE RESORCE 278
  PRINT RESOURCE 281
  RETRIEVE RESOURCE 379
  SEND RESOURCE 386
Extended Addressability 431
EXTENDFILE operand (INSTALL MICROCODE) 363, 369
extent 257
file control name 270
file control name (a resource identifier) 108
filemode number (a resource attribute) 114
FILEPSW operand (PREPARE DATASET/PRINT) 298
FILEPSW operand (RECEIVE/RETRIEVE statements) 335
FILEPSW operand (SEND/REPLACE statements) 340
first start date/time 155, 398
FIRSTQ operand
  UNLOAD RESOURCE 308
FIRSTQ operand (UNLOAD RESOURCE) 308
fixed length records 100, 261
FMN operand (PREPARE CLIST/JOB/PROGRAM) 294
FMN operand (PREPARE DATASET/PRINT) 298
focal point, definition of 34, 215
FORCE operand (ACTIVATE NODE) 349
FORCE operand (DELETE NODE) 223
FORCEIND operand (INSTALL MICROCODE) 363, 369
Format of MVS MODIFY Commands 549
format of variable-length records in user files 101, 262
FORMAT utility 202, 204

NetView DM for MVS User's Guide
function information for LU 6.2 resources (continued)
   to be deleted resource name 162, 403
   transmission return code 161, 402
   when effective 161, 402
function information in the TCF, looking at 156
FUNCTION operand 202
function rc 161, 402
function status 161, 402
function termination rc 161, 402
function type codes 564
function type names 564
functions
definition of 119, 120, 121, 313, 314
   for each node type 564
   list of NetView DM for MVS 564
   resetting 171
G
generable attributes (a resource identifier) 111
general purpose control statements 209
COPY 209
DUMP 209
EJECT 209
END 210
SPACE 210
TITLE 210
general rules for using JCL 200
to run MAINTDAT in VSAM Environment 204
to run SUBMIT 206
to run TCFMAINT 208
what could go wrong 200
GENERATT operand
LOAD RESOURCE 289
Get Plan command 140
GIX
commands 11
commands log 17
exiting from 16
getting help 8
going to ISPF from 15
in Katakana 7
logging on to 5
logon panel 7
main menu 8, 9
moving between panels 13
panel layout 10
printing from 15
GIX edit data sets 464
global name 65, 243
GRANTED field 527
group model 58, 81
group name 58, 81, 228
group name (a resource identifier) 108
group name, node 87
group name, resource 89
group of nodes (a phase attribute) 129
GROUP operand 249
ADD RESOURCE 254
DEASGN statements 251
DEASSIGN RESOURCE 255
PREPARE DATASET/PRINT 297
PRINT HISTORY 260
GROUPID operand
ACCEPT MICROCODE 346
ACCEPT RESOURCE 347
ADD RESOURCE 254
ADD statements 249
CONVERT DATASET/DUMP 284
DEASGN statements 251
DEASSIGN RESOURCE 255
DEFINE PHASE 321
DELETE MICROCODE 351
DELETE RESOURCE 353
DELETE statements 328
INITIATE statements 330
INSTALL MICROCODE 359
INSTALL RESOURCE 364
RECEIVE/RETRIEVE statements 335
REMOVE MICROCODE 371
REMOVE RESOURCE 373
RETRIEVE MICROCODE 376
RETRIEVE RESOURCE 379
SEND MICROCODE 383
SEND RESOURCE 386
SEND/REPLACE statements 340

H (hold transmission of a phase) line command 507
hard copy, IOF 487
help text, displaying 8, 492
highest phase return code 155, 398
highest return code 150, 153, 159, 395, 396, 400
highest return code to invoke proc 396
highest return code to invoke procedure 153
hold option (a phase attribute) 130
hold transmission for a node or group (IOF command) 515
holding a specific phase for a node 518
holding file, fast re-initialization of 440

I (display conditioning phases) line command 513
ID operand
CONVERT DATASET/DUMP 284
INSERT/REPLACE CLIST 286
INSERT/REPLACE JOB 287
PREPARE DATASET/PRINT 297
IDLY field 528
IDLY parameter 522
immediate end transmission (IOF command) 531
immediate quiesce transmission 552
immediate quiesce transmission (IOF command) 530
immediate retry interval 522
INDD operand
LOAD RESOURCE 289
PREPARE CLIST/JOB/PROGRAM 294
PREPARE DATASET/PRINT 296
information about process performed 162
initial values of resource types 62
initiate completion code 163, 402, 404
INITIATE FUNCTION statement (SUBMIT) 330
INITIATE JOB statement (SUBMIT) 330
INITIATE PROCEDURE function, example in a transmission plan 135
INITIATE PROCEDURE statement (SUBMIT) 357
INITIATE statements (SUBMIT) 330
Initiate transmission plan, example of 132
input fields 11
INSERT CLIST statement (MAINTDAT) 286
INSERT JOB statement (MAINTDAT) 287
inserting new phases and functions into a plan 135
inserting resource names into a group 82
inserting resources into the repository 109
INST operand 202
INST parameter 477
INSTALL MICROCODE statement (SUBMIT) 359
INSTALL RESOURCE statement (SUBMIT) 364
installation 94
Instance number 477
intermediate node, definition of 34, 215
IOF
change transmission profiles 521
control transmission schedules 495
exiting from 494
getting help 492
katakana terminal 492
log on from NCCF/TAF 486
log on from VTAM/TCAM 486
logon panel 486
master menu 488
operator functions 493
panel flow 493
panel layout 489
printer 487
start transmission 494
stopping 532
IOF logon panel 486
IOF master menu 488
IRTRY field 527
IRTRY parameter 522
ISPF/PDF
editing CLIST and Job resources 118
going to 15
logging on to GIX from 6

J
JCL cards 29
JCL, using 200
to run MAINTDAT in VSAM Environment 204
to run SUBMIT 206
to run TCFMAINT 208
what could go wrong 200
job ERASPLAN 423
job name 188
job name (a resource attribute) 113
JSC operand 287
jump commands 13, 14
jump functions 14

K
Katakana, using GIX in 7
key number 257, 270
KEYLEN operand (PREPARE DATASET/PRINT) 298
KEYNUMB operand
CONVERT RESOURCE 270
KEYNUMB operand (continued)
  DELETE HISTORY 257
  DELETE RESOURCE 354
  DELETE statements 329
  ERASE RESOURCE 274
  PREPARE RESOURCE 278
  PRINT RESOURCE 281
  RETRIEVE RESOURCE 379
  SEND RESOURCE 386

KEYPOS operand (PREPARE DATASET/PRINT) 298

LBLEN operand
  PREPARE DATASET/PRINT 299

LBLRIND operand (PREPARE DATASET/PRINT) 297

LIBMAINT utility 202, 204

LIBNAME operand
  CONVERT RESOURCE 270
  DEFINE/DELETE GROUP statements 229
  DELETE HISTORY 257
  DELETE RESOURCE 354
  ERASE RESOURCE 274
  PREPARE RESOURCE 278
  PRINT RESOURCE 282
  RECEIVE/RETRIEVE statements 335
  RETRIEVE RESOURCE 379
  SEND RESOURCE 386
  SEND/REPLACE statements 340

library file, fast re-initialization of 440
  library name 229, 257
  library name (a resource identifier) 109
  library type 229, 257
  line and node problems 540
  line commands 503
  line commands on full-list panels 12
  line problem 518
  LINECNT operand 202
  lines per page 28
  linetype (a node attribute) 38, 220
  LINETYPE operand
  CHANGE/DEFINE NODE statement 220

LMTYPE operand
  DEFINE/DELETE GROUP statements 229
  DELETE HISTORY 257
  DELETE RESOURCE 354
  ERASE RESOURCE 274
  PREPARE RESOURCE 278
  PRINT RESOURCE 282
  RECEIVE/RETRIEVE statements 335
  RETRIEVE RESOURCE 379
  SEND RESOURCE 386
  SEND/REPLACE statements 340
  LOAD RESOURCE statement (MAINTDAT) 288

LOAD RESOURCES (Embedded Form) 292
  loading resources into the repository using batch 261
  loading resources into the repository using GIX 100, 107
  locking resources 106
  log data sets 465
  log off from IOF (IOF command) 532
  logging on to GIX 5
    from ISPF/PDF 6
    from TSO 5
    what could go wrong 7
  logging on to IOF 485
    from NCCF/TAF 486
    from VTAM or TCAM 486
    what could go wrong 486
  logical block length (a resource attribute) 114
  logical record length (a resource attribute) 114
  logical unit (a node attribute) 38
  logical units
    browsing assignment to transmission profiles 49
    changing the assignment to transmission profiles 47, 235
    name 235
    printing the assignment to transmission profiles 238
    printing transmission profile assignments 49

LOGMOD operand
  CHANGE/DEFINE NODE statement 219

logon (a phase attribute) 131
  logon ID (a node attribute) 38, 220
  logon mode (a node attribute) 39, 219

Logon to GIX panel 7

LRLEN operand
  PREPARE CLIST/JOB/PROGRAM 294
  PREPARE DATASET/PRINT 299

LU 0 node types 561
  LU 6.2 node types 34, 216, 562
  LU 6.2 resource types 62, 240
  LU0 node types 34, 216
  LU0 resource types 62, 240

LUNAME operand
  CHANGE LUTP statement 235
  CHANGE/DEFINE NODE statement 218

M

MAINTAIN operand (PREPARE DATASET/PRINT) 299
  maintaining NetView DM for MVS files 429

CLISTS library 464
  commands and validation errors data set 464
  distributed resource directory 449
  DSXJOBS file 466
  GIX data sets 464
  GIX edit data sets 464
  log data sets 465
  NetView DM for MVS work file 458
  plan library 444
  procedure 429
  reports data set 465
  request queue file 462
  resource repository 430
  skeletons library 464
  snap data sets 464
  TCF 452
  TCP work file 460
  unsolicited message file 457
MAINTDAT utility
definition of 197, 261
how to run 203
sample JCL to run 204
managing node unsolicited reports 408
managing node-solicited requests 408
managing resources after transmission 183, 267
managing resources in the repository 261
managing submitted plans 147, 395
looking at function information 156, 400
looking at phase by node information 154, 397
looking at phase information 151, 396
looking at plan information 148, 395
looking at TCF information 147
printing the contents of a plan 395
masks, resource name 67, 245
match attributes (a resource identifier) 111
MATCHATT operand
LOAD RESOURCE 289
MATCHIND operand
DELETE MICROCODE 351
DELETE RESOURCE 354
ERASE RESOURCE 275
INITIATE PROCEDURE 358
INSTALL MICROCODE 307, 361
INSTALL RESOURCE 367
PRINT RESOURCE 282
RETRIEVE MICROCODE 376
RETRIEVE RESOURCE 380
SEND MICROCODE 384
SEND RESOURCE 387
matching indicators 161, 402
matching indicators (a resource identifier) 111, 187
max delay 150, 395
max start date/time 153, 155, 397, 398
maximum number of transmission tasks 522
MAXN parameter 522
MAXTASK parameter 480
MEMBER operand
CONVERT RESOURCE 270
DELETE HISTORY 258
DELETE RESOURCE 354
ERASE RESOURCE 274
PREPARE RESOURCE 278
PRINT RESOURCE 282
RETRIEVE RESOURCE 379
SEND RESOURCE 386
member type 229, 257
message
browsing 192
deleting 192
queuing 191
MINGR parameter 522
minimum number of transmission tasks 522
MOD operand (continued)
PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB 304
PRINT RESOURCE 282
RECEIVE/RETRIEVE statements 336
RETRIEVE RESOURCE 379
SEND RESOURCE 387
SEND/REPLACE statements 340
MODE operand (RECEIVE/RETRIEVE statements) 336
modification (a resource identifier) 108
modifier 199
MODIFY Commands 549
moving between GIX panels 13
moving elements in a plan 139
MSG operand (PREPARE MESSAGE) 303
MSGINFO parameter 479
MULTIERA operand
ERASE RESOURCE 275
MULTIERA operand (ERASE RESOURCE) 275
multiple TCPs 471
MVS MODIFY Commands 549
change transmission for a node 553
display transmission for a node 552
immediate quiesce transmission 552
node, display transmission for 552, 553
quiesce transmission 552
start transmission 552
MVS MODIFY Commands, controlling the TCP 550
MVS MODIFY Commands, format 549

N
name at node 93
name at node (a resource attribute) 73
NAME operand
ACCEPT MICROCODE 346
ACCEPT RESOURCE 347
ADD RESOURCE 254
ADD statements 249
CHANGE/DEFINE NODE statement 218
CHANGE/DEFINE USER statement 212
CONVERT RESOURCE 271
DEASGN statements 251
DEASSIGN RESOURCE 255
DEFINE PHASE 321
DEFINE PLAN 319
DEFINE RESOURCE 256
DEFINE/DELETE GROUP statements 228
DELETE DEFINITION 252
DELETE HISTORY 258
DELETE MICROCODE 351
DELETE NODE statement 223
DELETE PLAN 324
DELETE RESOURCE 353
DELETE SPLAN 411
DELETE statements 328
DELETE/PRINT USER statement 214
ERASE PLAN 413
ERASE RESOURCE 274
INITIATE PROCEDURE 357
INITIATE statements 330
INSTALL MICROCODE 359
INSTALL RESOURCE 364
LOAD RESOURCE 288
node type 153, 155, 398
node types
as a batch operand 213, 218, 224, 229, 231, 252
as a GIX field 24, 36, 56, 69, 79, 85
changing authorization to 24
CMEP 34, 216
CMFP 34, 216
definition of 33, 215
list of 561
LU 0 561
LU 6.2 34, 216, 562
LU0 34, 216
NDMT 35, 216
user-defined 34, 216, 562
node unsolicited reports 408
node-solicited requests 408, 471
NODECLASS operand
CHANGE/DEFINE USER statement 212
NODELNAME operand
ADD statements 250
RECEIVE/RETRIEVE statements 336
SEND/REPLACE statements 340
NODENAME operand
ADD statements 250
DELETE statements 328
RECEIVE/RETRIEVE statements 336
SEND/REPLACE statements 340
nodes
browsing 45
changing 43
defining 33, 215
deleting 44
intermediate 34, 215
model 36
name 36, 87
organizing into groups 54, 225
printing 45
NODESLNA operand (RECEIVE/RETRIEVE statements) 336
NODESLNA operand (SEND/REPLACE statements) 340
NODETYPE operand
CHANGE/DEFINE NODE statement 218
CHANGE/DEFINE USER statement 213
DEFINE/DLT GROUP statements 229
DELETE DEFINITION 252
PRINT GROUP statement 231
PRINT HISTORY 260
PRINT NODE statement 224
NOTE operand
CHANGE/DEFINE NODE statement 221
DEFINE PHASE 322
DEFINE PLAN 319
notes (a node attribute) 39, 221
number of completed/deleted phases 150, 395
number of immediate retries 522
number of nodes 153, 397
number of retries 522
operator control 471, 476
OPTCTL parameter 476
organizing nodes into groups 54, 225
organizing resources into groups 79, 225
origin (a resource identifier) 108
ORIGIN operand
CONVERT DATASET/DUMP 284
CONVERT RESOURCE 271
DELETE RESOURCE 354
ERASE RESOURCE 274
PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB 304
PRINT RESOURCE 281
RETRIEVE RESOURCE 379
SEND RESOURCE 387
SEND/REPLACE statements 340
OUTDD operand 285
OUTDD operand (UNLOAD RESOURCE) 307
OUTTYPE operand
UNLOAD RESOURCE 307
OUTTYPE operand (UNLOAD RESOURCE) 307

P
PAD operand (PREPARE DATASET/PRINT) 300
page allocation 28
panel identification 10
panel layout 10
panel title 10
panels, GIX
Assign Resources to Nodes panel 88
Browse Network panel 98
Browse Plan Information panel 125
Browse Plan Library Space panel 141
Browse Selected Information panel (for connection profiles) 53
Browse Selected Information panel (for transmission profiles) 49
Browse Selected Information panel (network configuration) 99
Browse/Print of Plan panel 124
Change CURTASK panel 525
Change LU/TP Assignments panel 48
Change Node Type Authorizations panel 24
Change Password and GIX Authorizations panel 21
Change Resource Authorizations panel 25
Configure Network panel 35
Confirm the Deletion of a User panel 31
Create or Change a Plan panel 127
Create Resource History at the Node panel 93
Create/Change a Group panel 82
Create/Change Node Groups panel 58
Define Resource panel 70
Display and Control Transmission Profiles panel 526
Display and Control Transmission Schedules panel 496
Fast Insertion from DRD panel 137
Fast Insertion from Repository panel 138
Function Information panel (LU0) 158
Function Information panel for microcode distribution 160
Function Resource Selection panel 133, 137, 139
Handle Unsolicited Messages panel 534
Identify the Resource panel 110, 187
Identify the Resource panel for microcode 110
Initiate Parameters panel 134

O
object size (a resource attribute) 115
OPENOUTD operand 285
operands, control statement 199
panels, GIX (continued)
IOF Logon panel 486
IOF Master Menu 488
layout of 10
List of Resources panel 99
List of Selected Resources panel 109, 186
logon to GIX 7
Maintain LU/TP Assignments panel 47
Manage Groups panel 56
Manage Resources panel 107
Manage Submitted Plans 149
moving between 13
NetView Distribution Manager Main Menu 8
Phase by Node Information panel 154
Phase Information panel 152
Phase Parameters panel 129
Plan Information panel 150
Plan Summary panel 169
Prepare Plans panel 123
printing 16
Procedure Qualifiers panel 133
Reset Plan panel 171
Select Resource Type for Grouping panel 80
Select Resource Type panel 99
selecting from 9
Set Profiles panel 20
Specify Attributes for Resource panel 72
Specify Attributes of Plan panel 127
Specify Attributes panel 113, 188
Specify Group and Select Operation panel 81
Specify Node and Select Resource Type panel 86
Specify Node Attributes panel 37
Specify Node Authorization panel 41
Specify Node Class Authorizations of User panel 26
Specify Operation on Resource panel 71
Specify Phase Attributes of Plan panel 128
Specify Resource Assignment Operation panel 86
Specify Resource Class Authorizations of User panel 27
Specify User Defaults panel 28
Specify Windows Attributes of Phase 131
Submit Plan panel 145
Transmission Display and Control panel 502
Validation Errors of Plan panel 144
PARM= statement 474
PARTDEF operand
ADD RESOURCE 254
DEFINE RESOURCE 256
partial matching in transmission plans 68, 246
partial naming 254
partial naming and matching 67
partial naming for resource definitions 67, 245
partially defined name, GIX field 72, 89
password (a node attribute) 38, 220
password (a phase attribute) 131
PASSWORD operand 202
CHANGE/DEFINE NODE statement 220
CHANGE/DEFINE USER statement 213
DEFINE PHASE 322
password, IOF 487
password, user 21
PDOS resources, naming conventions for 571
PENDING operand (DELETE SPLAN) 411
PF keys 11
PGM operand 202
phase attributes 129
batch job name 130
batch job submission conditioning value 130
conditioning criteria 130
conditioning mode 130
conditioning phase 130
conditioning value 130
cut off 129
group of nodes 129
hold option 130
logon 131
node name 129
password 131
phase name 129
phase notes 131
scheduling 129
termination value 130
track option 131
window option 131
phase by node information
cut-off date 397
cut-off time 398
cut-off time interval 154, 398
DRD updated 154, 398
date/time 154, 398
first start date/time 155, 398
highest phase return code 155, 398
last start date/time 155, 398
max start date/time 155, 398
network change effect 155, 397
no. of completed/deleted functions 155, 398
node type 155, 398
scheduled start date/time 155, 398
sequence termination return code 155, 398
status 155, 398
total elapsed time 156, 400
total number of functions 156, 400
windows 400
phase by node information in the TCF, looking at 154
phase information
completed/deleted phases by node 152, 396
conditioning phase name 152, 396
conditioning return code 152, 396
cut-off date 396
cut-off time 396
cut-off time interval 152, 396
deleted by 152, 396
highest return code 153, 396
highest return code to invoke proc 396
highest return code to invoke procedure 153
max start date/time 397
max start date/time 153
node name 153, 397
node or group of nodes 153, 397
node type 153, 397
number of nodes 153, 397
phase notes 153, 397
phase status 153, 397
phase termination return code 153, 397
procedure invoked at end of phase 153, 397
scheduled start date/time 153, 397
phase information (continued)

TCP error messages 153, 397
windows 397

phase information in the TCF, looking at 151

browse phase information 152
select the plan 151

phase name (a phase attribute) 129

phase notes 153, 397
phase notes (a phase attribute) 131

PHASE operand (RESET PLAN) 419
phase start-up, controlling 477
phase status 153, 397
phase termination return code 153, 397

phases

attributes of 129
change the status of 507
conditioned 514
conditioning 513
definition of 119, 313
displaying 502
hold 507
holding 518
in a recursive plan 120, 314
list functions 510
list with status 510
name 129
release 507, 508
restart 509

physical sequential file 100, 261
physical sequential files 184

plan attributes

plan notes 128
plan priority 128
recursive 128
start time delay 128

plan information

deleted by 150, 395
highest return code 150, 395
max delay 150, 395
number of completed/deleted phases 150, 395
plan notes 150, 395
planer ID 150, 395
priority 150, 395
recursion schedule 150, 395
recrucesive plan 150, 396
status 150, 396
submission date and time 151, 396
submitted by 151, 396
total number of phases 151, 396

plan information in the TCF, looking at 148
review the plan information 149
specify the plan name 148

plan library 119, 141, 313
plan model name 29

plan name 126

plan notes 150, 395
plan notes (a plan attribute) 128
plan priority 121
plan priority (a plan attribute) 128
plan recursion 120, 313
plan summary 168
planner ID 150, 395

PLCOPY 197, 447
PLCOPY utility

how to run 208

POSC resources, naming conventions for 572
post test 163, 404
posttest 162, 403

PRETEST operand

INSTALL MICROCODE 361
INSTALL RESOURCE 368
REMOVE MICROCODE 371
REMOVE RESOURCE 374

pre test 163, 404

PREPARE CLIST statement (MAINTDAT) 294
PREPARE DATASET statement (MAINTDAT) 296

embedded form 300

PREPARE JOB statement (MAINTDAT) 294
PREPARE MESSAGE statement (MAINTDAT) 303
PREPARE PRINT statement (MAINTDAT) 296

embedded form 300

PREPARE RESOURCE statement (MAINTDAT) 277

preparing job control statements 472
preparing messages using GIX 191

browsing messages 192
deleting messages 192
queueing messages 191

preparing resources for transmission 97, 261
appending resources 117

browsing the repository directory 117
deleting resources from the repository 117

editing CLIST and Job resources using ISPF/PDF 117
loading resources into the repository 100, 261
printing resources 118
printing the repository directory 117
querying space in the repository 118
replacing resources 117

preparing transmission plans 119
preparing transmission plans using batch 313
pretest 162, 403

PRETEST operand (INSTALL MICROCODE) 361
PRETEST operand (INSTALL RESOURCE) 368

primary and embedded statements 265
primary and secondary return codes 159, 401
primary commands 11

PRINT APIQENT statement (TCFMAINT) 415
PRINT CLIST statement (MAINTDAT) 304
PRINT CPLU statement 237
PRINT DATASET statement (MAINTDAT) 304
PRINT DEFINITION statement (SUBMIT) 253
PRINT DIRECTORY statement (MAINTDAT) 280
PRINT DUMP statement (MAINTDAT) 304
PRINT ERRORLOG statement (MAINTDAT) 304
PRINT GROUP statement 231
PRINT HISTORY statement (SUBMIT) 260
PRINT JOB statement (MAINTDAT) 304
PRINT NODE statement 224
PRINT PLAN statement (TCFMAINT) 416
PRINT REQUEST statement (TCFMAINT) 417
PRINT RESOURCE statement (MAINTDAT) 281
PRINT TPLU statement 238
PRINT USER statement 214

printer name 29
printer, IOF 487

printer name 29
printer, IOF 487
<table>
<thead>
<tr>
<th>Resource Attributes (continued)</th>
<th>Resource Identifiers (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>data already compressed 115</td>
<td>modification 108</td>
</tr>
<tr>
<td>data set name 113, 188</td>
<td>origin 108</td>
</tr>
<tr>
<td>disposition 189</td>
<td>resource name 108, 111, 187</td>
</tr>
<tr>
<td>filemode number 114</td>
<td>sublibrary name 109</td>
</tr>
<tr>
<td>FS compression 115</td>
<td>type attributes 111</td>
</tr>
<tr>
<td>FS parms 116</td>
<td>version 108</td>
</tr>
<tr>
<td>job name 113, 188</td>
<td>resource model 72</td>
</tr>
<tr>
<td>label sequence number 114, 189</td>
<td>resource name 71, 88</td>
</tr>
<tr>
<td>label type 114, 189</td>
<td>resource name (a resource identifier) 108, 111, 187</td>
</tr>
<tr>
<td>logical block length 114</td>
<td>resource repository 100</td>
</tr>
<tr>
<td>logical record length 114</td>
<td>maintaining the 430</td>
</tr>
<tr>
<td>name at node 73</td>
<td>resource type codes 566</td>
</tr>
<tr>
<td>object size 115</td>
<td>resource type names 566</td>
</tr>
<tr>
<td>processing size 115</td>
<td>resource types</td>
</tr>
<tr>
<td>restriction 73</td>
<td>as a batch operand 213, 220, 231, 252, 253</td>
</tr>
<tr>
<td>scratch 189</td>
<td>as a GIX field 25, 42, 70, 80, 87</td>
</tr>
<tr>
<td>to be compressed 115</td>
<td>available at each node 566</td>
</tr>
<tr>
<td>unit 113, 188</td>
<td>functions available for 567</td>
</tr>
<tr>
<td>user compression algorithm 116</td>
<td>initial values for 62</td>
</tr>
<tr>
<td>user compression parameters 116</td>
<td>list of 61, 240</td>
</tr>
<tr>
<td>volume 113, 188</td>
<td>list of NetView DM for MVS 566</td>
</tr>
<tr>
<td>VSAM catalog name 114</td>
<td>LU 6.2 62, 240</td>
</tr>
<tr>
<td></td>
<td>LU0 62, 240</td>
</tr>
<tr>
<td>resource class</td>
<td>REST operand (INITIATE statements) 330</td>
</tr>
<tr>
<td>as a batch field 221</td>
<td>RESTRICT operand</td>
</tr>
<tr>
<td>as a GIX field 27, 42, 73</td>
<td>DEFINE RESOURCE 256</td>
</tr>
<tr>
<td>default 28</td>
<td>restriction (a resource attribute) 73</td>
</tr>
<tr>
<td>resource definition</td>
<td>RESTTYP parameter 481</td>
</tr>
<tr>
<td>browsing a 77</td>
<td>RESTYPE operand</td>
</tr>
<tr>
<td>changing a 76</td>
<td>ACCEPT RESOURCE 347</td>
</tr>
<tr>
<td>definition of 61, 239</td>
<td>ADD RESOURCE 254</td>
</tr>
<tr>
<td>deleting 252</td>
<td>CHANGE/DEFINE NODE statement 220</td>
</tr>
<tr>
<td>deleting a 76</td>
<td>CHANGE/DEFINE USER statement 213</td>
</tr>
<tr>
<td>printing 253</td>
<td>DEASSIGN RESOURCE 255</td>
</tr>
<tr>
<td>printing a 77</td>
<td>DEFINE RESOURCE 256</td>
</tr>
<tr>
<td>resource group</td>
<td>DEFINE/DELETE GROUP statements 229</td>
</tr>
<tr>
<td>as a GIX field 82</td>
<td>DELETE DEFINITION 252</td>
</tr>
<tr>
<td>browsing a 83</td>
<td>DELETE HISTORY 257</td>
</tr>
<tr>
<td>changing a 82</td>
<td>DELETE RESOURCE 353</td>
</tr>
<tr>
<td>defining a 79, 228</td>
<td>INSTALL RESOURCE 364</td>
</tr>
<tr>
<td>deleting a 83</td>
<td>LOAD RESOURCE 288</td>
</tr>
<tr>
<td>model 81</td>
<td>PRINT DEFINITION 253</td>
</tr>
<tr>
<td>name 81, 89</td>
<td>PRINT GROUP statement 231</td>
</tr>
<tr>
<td>organizing resources into a 79, 225</td>
<td>PRINT HISTORY 260</td>
</tr>
<tr>
<td>printing a 84</td>
<td>REMOVE RESOURCE 373</td>
</tr>
<tr>
<td>restrictions 82</td>
<td>RETRIEVE RESOURCE 378</td>
</tr>
<tr>
<td>resource history information for resource types</td>
<td>SEND RESOURCE 385</td>
</tr>
<tr>
<td>resource history record</td>
<td>ULOCK RESOURCE 311</td>
</tr>
<tr>
<td>changing a 94</td>
<td>UNINSTALL COMPONENT 392</td>
</tr>
<tr>
<td>contents of 93, 94, 248</td>
<td>UNLOAD RESOURCE 306</td>
</tr>
<tr>
<td>defining a 92</td>
<td>RESYNC operand (RECEIVE/RETRIEVE statements) 336</td>
</tr>
<tr>
<td>definition of 90, 247</td>
<td>RESYNC operand (SEND/REPLACE statements) 340</td>
</tr>
<tr>
<td>deleting a 95</td>
<td>RETINT field 527</td>
</tr>
<tr>
<td>resource identifiers 108</td>
<td>RETREIVE MICROCODE statement (SUBMIT) 376</td>
</tr>
<tr>
<td>class codes 112</td>
<td>RETRIEVE RESOURCE statement (SUBMIT) 378</td>
</tr>
<tr>
<td>completeness 108</td>
<td>RETRIEVE statements (SUBMIT) 332</td>
</tr>
<tr>
<td>file control name 108</td>
<td>RETRY field 527</td>
</tr>
<tr>
<td>generable attributes 111</td>
<td>rety interval 522</td>
</tr>
<tr>
<td>group name 108</td>
<td>return codes, batch utility 201</td>
</tr>
<tr>
<td>library name 109</td>
<td>reviewing the plan information 149</td>
</tr>
<tr>
<td>match attributes 111</td>
<td></td>
</tr>
<tr>
<td>matching indicators 111</td>
<td></td>
</tr>
</tbody>
</table>
Index 613

RGN (a node attribute) 40
RGN operand
CHANGE/DEFINE NODE statement 219
RPS resources, naming conventions for 572
running the Batch Utilities 202

S
S (restart a phase) line command 509
SAVE APIQENT statement (TCFMAINT) 420
SAVE PLAN statement (TCFMAINT) 421
SAVE REQUEST statement (TCFMAINT) 422
saving a plan 140
saving transmission plans using GIX 168
SC operand (RECEIVE/RETRIEVE statements) 336
scheduled start date/time 153, 155, 397, 398
scheduling (a phase attribute) 129
scratch (a resource attribute) 189
SCRATCH operand (RECEIVE/RETRIEVE statements) 336
SCRATCH operand (SEND/REPLACE statements) 341
SCRATCH operand (UNLOAD RESOURCE) 307
scroll field 10
selecting from the GIX Main Menu 9
selecting resources of a specific type 107, 185
selecting the plan 151
selecting the plan name 123
send 163, 404
SEND MICROCODE statement (SUBMIT) 383
SEND operand (INITIATE PROCEDURE) 357
SEND operand (INSTALL MICROCODE) 362
SEND operand (INSTALL RESOURCE) 368
SEND RESOURCE statement (SUBMIT) 385
SEND statements (SUBMIT) 337
sending 94
sequence termination return code 155, 398
server group name 229
server name (a node attribute) 40
server report code 162, 403
service provider
problems with the 543
SHR operand
ADD statements 250
PREPARE DATASET/PRINT 299
SEND/REPLACE statements 341
shutdown of TCP 538
side effect resource names 162, 403
side effects 162, 403
skeletons library 464
SNA sessions, current number 480
SNA sessions, maximum number 480
SNA/File Services class codes 584
Snads report code 162, 403
snap data sets 464
Software Profile Management Facility (SPMF) xviii
SPACE statement 210
specify node class authorizations of user 26
specify resource class authorizations of user 27
specify user defaults 28
specifying an operation on a node 36
specifying an operation on a resource 71
specifying node attributes 37
specifying node authorizations 41
specifying resource selection criteria 107, 186
specifying the connection profile name 50
specifying the job attributes 113, 188
specifying the node and selecting the resource type 86
specifying the node and the resource 92
specifying the node type 35
specifying the node type of a group 56
specifying the phase attributes of a plan 128
specifying the plan name 148
specifying the resource assignment operation 86
specifying the resource attributes 72
specifying the transmission profile name 47
SPMF (Software Profile Management Facility), xviii
SRCTYPE operand (PREPARE CLIST/JOB/PROGRAM) 294
SRCTYPE operand (PREPARE DATASET/PRINT) 299
SSP resources, naming conventions for 572
START parameter 478
start time delay (a plan attribute) 128
start transmission 552
start-time delay 120, 313
starting the transmission control program 471
starting transmission 477, 494
status 150, 155, 159, 161, 396, 398, 400, 402, 511
status (a node attribute) 40, 220
status of transmission 491
STATUS operand
CHANGE/DEFINE NODE statement 220
DEFINE PHASE 322
ERASE RESOURCE 274
STI 65, 243
stopping
the IOF 532
the TCP 530
transmission 529
structured names, definition of 66, 244, 582
SUBCAT operand
CONVERT RESOURCE 271
DELETE HISTORY 258
DELETE RESOURCE 354
DELETE statements 329
PREPARE RESOURCE 278
PRINT RESOURCE 282
RETRIEVE RESOURCE 380
SEND RESOURCE 387
SEND/REPLACE statements 341
subcategories 250
subcategory updates 250
SUBLIB operand
DEFINE/DELETE GROUP statements 229
DELETE HISTORY 258
DELETE RESOURCE 354
RECEIVE/RETRIEVE statements 336
RETRIEVE RESOURCE 380
SEND RESOURCE 387
SEND/REPLACE statements 341
sublibrary name 229
sublibrary name (a resource identifier) 109
SUBLNAME operand
CONVERT RESOURCE 271
ERASE RESOURCE 275
PREPARE RESOURCE 278
submission date and time 151, 396
SUBMIT control statements
for groups 227
for logical units 233
for LU 6.2 resource types 254
for LU0 resource types 248
for nodes 217
for tracking resources 257
for transmission plans 319
for user profiles 211
SUBMIT PLAN statement (SUBMIT) 326
SUBMIT transmission functions
for LU 6.2 node types 343
for LU0 node types 327
SUBMIT utility
definition of 197
how to run 204
sample JCL to run 206
submitted by 151, 396
submitting a transmission plan using batch 318
submitting JCL 483
submitting transmission plans 119, 142
submitting transmission plans using batch 313
submitting transmission plans using GIX 144
subtree identifier (STI), definition of 65, 243
subtrees 575
suppressing unsolicited messages 479
SVP
See service provider
SYNC operand (INITIATE statements) 330
SYSOUT class 29
T
TCAM, log on to IOF from 486
TCF 147
TCFMAINT control statements 409
TCFMAINT utility
definition of 198
how to run 206
sample JCL to run 208
TCP 147
TCP error message 159, 401
TCP error message number 162, 403
TCP error messages 153, 397
termination value (a phase attribute) 130
termination values 121, 315
at the function level 122
at the phase for node level 121, 315
at the phase level 121, 315
at the sequence level 121, 315
at the sequence or function levels 140
defining 139
for batch jobs 122
TIME operand
ACTIVATE NODE 349
DEFINE PHASE 322
INITIATE PROCEDURE 358
INSTALL MICROCODE 362
INSTALL RESOURCE 369
UNINSTALL COMPONENT 392
TIMEIND operand (continued)
INITIATE PROCEDURE 358
INSTALL MICROCODE 362
INSTALL RESOURCE 369
UNINSTALL COMPONENT 392
timzoffs (a node attribute) 40
TITLE statement 210
to be compressed (a resource attribute) 115
to be deleted matching indicator 163, 403
to be deleted resource name 162, 403
TOCOMPRS operand (LOAD RESOURCE) 290
TODA operand (RECEIVE/RETRIEVE statements) 335
TODATE operand (RECEIVE/RETRIEVE statements) 335
tokens, definition of 65, 243
total elapsed time 156, 400
total number of functions 156, 400
total number of phases 151, 396
TOTIME operand (RECEIVE/RETRIEVE statements) 335
TP name field 527
TPNAME operand
CHANGE LUTP statement 235
PRINT TPLU statement 238
TPRRETRY parameter 522
TPRTINT parameter 522
TRACK operand
ACCEPT MICROCODE 346
ACCEPT RESOURCE 347
DEFINE PHASE 323
DELETE MICROCODE 352
DELETE RESOURCE 355
INITIATE PROCEDURE 358
INSTALL MICROCODE 362
INSTALL RESOURCE 369
REMOVE MICROCODE 372
REMOVE RESOURCE 374
SEND MICROCODE 384
SEND RESOURCE 388
UNINSTALL COMPONENT 392
track option (a phase attribute) 131
tracking at the phase level 131
tracking information 11
tracking resources 90, 247
transmission
hold 515
immediate quiesce 530
problems 537
quiesce 529
stopping 529
transmission control file (TCF) 119, 147, 313, 395
contents of 147, 395
function information 156, 400
looking at information in the 147, 395
phase by node information 154, 397
phase information 151, 396
plan information 148, 395
transmission control master menu 488
transmission control program
cold start 478
definition of 471
job control statements, running 472
log file 537
operator control 471
profile 476
transmission control program (continued)
  shutdown 538
  starting the 471
  startup parameters 475
  stopping the 530
  termination 478
    using multiple TCPs 471
    warm start 478
transmission control program (TCP) 119, 147, 313
transmission error identification 159, 401
transmission failures, recovering from 180, 405
  NetView DM for MVS host errors 181, 406
  node errors 182, 407
  VTAM errors 182, 406
transmission function termination return codes 181, 406
transmission functions for LU0 node types 327
transmission plan
  attributes 128
  browsing a 123
  changing 126, 140
  creating 126
  definition of 119, 313
  deleting 141, 163
  erasing 163
  name 126
  printing a 123
  recursion 120, 313
  recursive 120, 313
  renaming 142
  resetting 169
  saving 140, 168
  structure 119, 313
  submitting 142
  validating 142
transmission plans, partial matching of resources 68, 246
transmission profile
  attributes 46, 232
    browsing logical unit assignments 49
    changing 317, 521
    changing the assignment of logical units to a 47, 235
    creating 317
    defaults 46
    definition of 46, 232
    displaying or changing 524
    name 235, 238
    printing logical unit assignments 49
    printing the assignment of logical units to a 238
    specifying which ones to use 481
transmission requests 471
transmission requests, controlling 496
transmission requests, selecting 497
  deleting a 506
  NSRs 497
  phases 497
  refid=D&CC API requests 497
    with a specific status 499
    within a date or time range 498
transmission return code 161, 402
transmission schedule, changing 503
transmission schedules, controlling 495
  schedule display and control 496
transmission status 491
transmission windows
  cutoff date 323
  cutoff time 323
  defining (using batch) 323
  defining (using GIX) 131
  resetting (using batch) 419
  resetting (using GIX) 148
TSO
  logging on to GIX from 5
type attributes (a resource identifier) 111
TYPE field 527
TYPE operand
  CONVERT RESOURCE 270
  ERASE RESOURCE 273
  INSERT/REPLACE CLIST 286
  PREPARE RESOURCE 277
  PRINT DIRECTORY 280
  PRINT RESOURCE 281
TYPEAT operand
  LOAD RESOURCE 289

U
UDPROCSIZE operand
  INSTALL RESOURCE 365
ULOCK RESOURCE statement (MAINTDAT) 311
unattended mode 471
underscores, note on usage 199
uniform text 185
uniform text (a resource attribute) 116
UNINSTALL COMPONENT statement (SUBMIT) 392
uninstallation 94
unit (a resource attribute) 113, 188
UNIT operand (UNLOAD RESOURCE) 310
UNLOAD RESOURCE statement (MAINTDAT) 306
unloading resources from the repository 183, 267
  into physical sequential files 184
  into VSAM files 184
  using GIX 185
  with uniform text 185
unsolicited messages 479
unsolicited messages, handling 533
routing 533
viewing 534
user authorizations 21
  change node profile authorizations 22
  delete phase from IOF 23
  IOF 22
    maintain repository and plan library 22
    Manage other users' plans 23
    network configuration update 22
    plan preparation 22
    resource repository deletion and update 22
    submit and manage submitted plans 22
user compression algorithm (a resource attribute) 116
user defaults 28
  block size 28
  JCL cards 29
  lines per page 28
  node class 28
  page allocation 28
  plan model name 29
  printer name 29
user defaults (continued)
  resource class 28
  sysout class 29
  user device type 29
user device type 29
user ID 20
user parameters (a resource attribute) 116
user problems 541
user profile model 20
user profiles
  changing 29
  defining 20, 211
  definition of 19, 211
  deleting 30
  GIX authorizations 21
USERCOMP operand (LOAD RESOURCE) 290
USEREXIT operand
  CONVERT RESOURCE 269
  PREPARE RESOURCE 277
USERID operand 202
USERINFO operand
  CONVERT RESOURCE 271
  PREPARE RESOURCE 278
USERPARM operand (LOAD RESOURCE) 291
using the Get Plan command 140

V
  validating transmission plans 142
  validating transmission plans using GIX 143
  validation errors in a plan 142
variable-length records 101, 262
  format in the repository 102, 263
  format in user files 101, 262
variable-length records in the repository 102, 263
variables 199
VARLEN operand (PREPARE DATASET/PRINT) 299
VER operand
  CONVERT RESOURCE 271
  DELETE HISTORY 258
  DELETE RESOURCE 354
  DELETE statements 328
  ERASE RESOURCE 274
  INITIATE statements 330
  PREPARE CLIST/JOB/PROGRAM 294
  PREPARE RESOURCE 278
  PRINT CLIST/DATASET/DUMP/ERRORLOG/JOB 304
  PRINT RESOURCE 282
  RECEIVE/RETRIEVE statements 336
  RETRIEVE RESOURCE 380
  SEND RESOURCE 387
  SEND/REPLACE statements 340
verb 199
VERIFY operand (PREPARE DATASET/PRINT) 300
version (a resource identifier) 108
viewing unsolicited messages 534
VOLSER operand
  ADD statements 250
  PREPARE DATASET/PRINT 300
  SEND/REPLACE statements 341
volume (a resource attribute) 113, 188
VOLUME operand (UNLOAD RESOURCE) 309
VSAM catalog name (a resource attribute) 114
VSAM files 184
VSE 250
  CATLG operand 329
  CATPSW operand 329
  DSORG operand 329
  FILEPSW operand 329
  LIBNAME operand 329
  LMTYPE operand 329
members 250
NODELNAME operand 329
NODELNA operand 329
panels 250
programs 250
SUBLIB operand 329
sublibraries 250
VSE resources, naming conventions for 574
VTAM errors 182, 406
VTAM, log on to IOF from 486

W
  WAIT status, of functions 511
  warm starting the TCP 478
  when effective 161, 402
WINDOW operand
  DEFINE PHASE 323
  window option (a phase attribute) 131
  windows, transmission 397, 400
    cutoff date 323
    cutoff time 323
  defining (using batch) 323
  defining (using GIX) 131
  resetting (using batch) 419
  resetting (using GIX) 148

X
  XMFUNC operand
    CHANGE/DEFINE NODE statement 220
Communicating Your Comments to IBM

Tivoli® NetView® Distribution Manager for MVS™
User's Guide

Release 7
Publication No. SH19-6795-06

If you especially like or dislike anything about this book, 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 book. 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 prefer to send comments by mail, use the reader's comment form (RCF) at the back of this book. If you wish, you can give the RCF to the local branch office or IBM representative for postage-paid mailing.

If you prefer to send comments by FAX, use this number, which is in Italy:
39+06+596+62077

If you prefer to send comments electronically, use this network ID:
ROMERCF@VNET.IBM.COM

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies
Help us help you!

Tivoli® NetView® Distribution Manager for MVS™
User’s Guide

Release 7
Publication No. SH19-6795-06

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?

<table>
<thead>
<tr>
<th>Satisfied</th>
<th>Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

How satisfied are you that the information in this book is:

<table>
<thead>
<tr>
<th>Satisfied</th>
<th>Dissatisfied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
</tr>
<tr>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>Easy to find</td>
<td></td>
</tr>
<tr>
<td>Easy to understand</td>
<td></td>
</tr>
<tr>
<td>Well organized</td>
<td></td>
</tr>
<tr>
<td>Applicable to your task</td>
<td></td>
</tr>
</tbody>
</table>

Specific Comments or Problems:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Please tell us how we can improve this book:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name ___________________________________________ Address ________________________________

Company or Organization _____________________________________________________________

Phone No. ______________________________________________