NetView® Performance Monitor
Diagnosis
Version 2 Release 7
Contents

Figures............................................................................. xi

Tables ............................................................................ xiii

Preface ........................................................................... xv

  Who Should Read this Book ............................................ xv
  What this Book Contains .............................................. xv
  Publications .................................................................. xvi
    NPM Library ........................................................... xvi
    Related Publications ................................................ xvi
    Accessing Publications Online .................................. xvi
    Ordering Publications .............................................. xvi
    Providing Feedback about Publications .................... xvi
  Contacting Customer Support ..................................... xvii

1 Using LookAt to Look Up Message Explanations ........... xviii

1 Conventions Used in This Book ................................. xviii

1 Changes in This Edition ............................................. xix

1 Changes in the Previous Edition ................................. xix

1 Using Syntax Diagrams .............................................. xix
  Reading Syntax Diagrams ........................................... xix
  Abbreviating Keywords .............................................. xix
  Parameters ............................................................ xix
    Required Parameters ........................................... xx
    Default and Optional Parameters .......................... xx
  Repeating Parameters .............................................. xx
  Reading Fragments .................................................. xxi
  Long Syntax Diagrams ............................................. xxi

Part I. Classifying and Documenting Problems ............... 1

Chapter 1. NPM Diagnosis Overview ............................. 5

  Using Checklists ...................................................... 6
  Classifying Problems ............................................... 6
  Gathering Information about Problems ...................... 6
    Using Traces ....................................................... 6
    Using Traps ......................................................... 6
    Using FNMILOG and SYSPRINT Data Sets ............... 7
    Using the PANELID and PRINT Commands ............. 7
Chapter 5. Documenting Specific Function Problems ................. 47
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCP Definition Problems</td>
<td>53</td>
</tr>
<tr>
<td>Information Needed</td>
<td>53</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>53</td>
</tr>
<tr>
<td>NetWare Resources Collection Problems</td>
<td>54</td>
</tr>
<tr>
<td>Information Needed</td>
<td>54</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>54</td>
</tr>
<tr>
<td>Network Collection Problems</td>
<td>56</td>
</tr>
<tr>
<td>Information Needed</td>
<td>56</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>56</td>
</tr>
<tr>
<td>NetView Synergy Interface (NSI) Problems</td>
<td>57</td>
</tr>
<tr>
<td>Information Needed</td>
<td>57</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>57</td>
</tr>
<tr>
<td>ODLC LAN Resource Data Collection Problems</td>
<td>59</td>
</tr>
<tr>
<td>Information Needed</td>
<td>59</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>59</td>
</tr>
<tr>
<td>Online System Problems</td>
<td>60</td>
</tr>
<tr>
<td>Information Needed</td>
<td>60</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>60</td>
</tr>
<tr>
<td>Performance Alert Problems</td>
<td>61</td>
</tr>
<tr>
<td>Information Needed</td>
<td>61</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>61</td>
</tr>
<tr>
<td>RTM Collection Problems</td>
<td>62</td>
</tr>
<tr>
<td>Information Needed</td>
<td>62</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>62</td>
</tr>
<tr>
<td>Session Collection Problems</td>
<td>63</td>
</tr>
<tr>
<td>Information Needed</td>
<td>63</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>63</td>
</tr>
<tr>
<td>System Console Support Problems</td>
<td>65</td>
</tr>
<tr>
<td>Information Needed</td>
<td>65</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>65</td>
</tr>
<tr>
<td>SMF Collection Problems</td>
<td>66</td>
</tr>
<tr>
<td>Information Needed</td>
<td>66</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>66</td>
</tr>
<tr>
<td>Transit Analysis Program (TAP) Problems</td>
<td>68</td>
</tr>
<tr>
<td>Information Needed</td>
<td>68</td>
</tr>
<tr>
<td>Collecting the Information</td>
<td>68</td>
</tr>
<tr>
<td>VTAM Statistics Collection Problems</td>
<td>69</td>
</tr>
<tr>
<td>Information Needed</td>
<td>69</td>
</tr>
</tbody>
</table>
Collecting the Information ................................................................. 69

Part II. Using Diagnostic Tools ....................................................... 71

Chapter 6. Using PDI, FNMILOG, SYSPRINT, and PANELID .......... 73

Using the PDI Control Block ............................................................... 73
Using the FNMILOG and SYSPRINT Data Sets .................................. 74
FNMILOG Data Set ............................................................................. 74
  NPM Virtual Storage Allocation ..................................................... 74
  Virtual Storage by Subpool .......................................................... 75
  Configuration Summary from NCP RRT and Initialization Statement .. 76
  Configuration Hierarchy ............................................................... 77
  Totals by Resource Type Table ..................................................... 78
  Configuration Symbol Resource Table ........................................ 78
  Synonym Chain Distribution Table .............................................. 78
  Sample FNMILOG ........................................................................ 79
SYSPRINT Data Set ........................................................................... 84
  Diagnosing with the PANELID Command ....................................... 87
  Diagnosing with the PRINT Command ......................................... 87

Chapter 7. Using the Online Problem Determination Facilities ........ 89

Selecting the Problem Determination Option .................................... 89
Displaying Main Storage ................................................................. 90
  Displaying Virtual Storage ........................................................... 92
Using Display Commands .............................................................. 92
  Displaying Diagnostic Trace Options .......................................... 96
  Displaying Diagnostic Options .................................................... 98
Displaying the Operator Traces ....................................................... 100
Displaying Session Traces ............................................................ 101
Adding a Diagnostic Trap .............................................................. 102
Displaying the Status of Diagnostic Traps ....................................... 104
Displaying Trap Values ............................................................... 106
Taking an NPM Storage Dump ....................................................... 107
Ending NPM Abnormally ............................................................... 108

Chapter 8. NPM DEBUG and TRAP Initialization Statements ........... 109

DEBUG Statement ............................................................................ 110
TRAP Statement .............................................................................. 119
  Determining the Applicable Parameters for Your Trap ................. 125

NPM Diagnosis
Chapter 9. Using the NPM Internal Trace ........................................... 127

Finding the Internal Trace Table................................................. 127
Understanding Trace Table Entries.............................................. 129
  Dispatcher Trace Entry (Types QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, SR) .... 132
  Message Trace Entry (MG).................................................... 133
  Command Trace Entry (SC, RC)............................................ 134
  VSAM and VTAM Request Parameter List (RPL) Trace Entry (VE, VS).................. 135
  VTAM Exit Trace Entry (VX)............................................... 136
  Logic Trace Entry (LG).......................................................... 137
  LAN Service Trace Entry (LN).............................................. 138
  Presentation Service Trace Entry (PS)................................... 139
  NPM-to-NPM Trace Entry (LQ, LR, RQ, RR)................................ 140
  Session Trace Entry (SA).................................................... 141
  TASK Start/TASK End Trace Entry (KS, KE)................................ 142
  TASK Abend Trace Entry (KA)............................................. 143
  Installation-Wide Exit Trace Entry (IX).................................... 144
  Input/Output Buffer (IOB) Trace Entry (IO)................................ 145
  Intertask Pipe Data Trace Entry (IT)...................................... 146
  APPCIOE Trace Entry (AE).................................................. 147

Chapter 10. Using the NPM Message Trace .................................... 149

Finding a Message Trace........................................................ 149
Message Trace Entry............................................................. 151

Part III. Appendixes .................................................................. 153

Appendix A. Problem Worksheets .............................................. 155
  NPM Internal Trace............................................................ 156
  NPM Major Control Blocks.................................................. 157
  NPM Abend Dump............................................................... 158
  NPM Save Area Layout........................................................ 159
  Problem Determination Information Control Block.......................... 160
  RTM2 Work Area Layout.................................................... 161
  VM GCS Abends and Program Checks...................................... 162
  VM GCS State Blocks.......................................................... 163

Appendix B. Checkpoint Processing for NetWare Resources Collections .................................................. 165
Figures

1. Diagnostic Procedure Flowchart ..................................................... 5  
2. Example of RTM2WA Summary ................................................... 39  
3. Example of NPM Dump. .......................................................... 39  
4. Example of FNMILOG Data Set (Part 1 of 5) ......................................... 79  
5. Example of FNMILOG Data Set (Part 2 of 5) ......................................... 80  
6. Example of FNMILOG Data Set (Part 3 of 5) ......................................... 81  
7. Example of FNMILOG Data Set (Part 4 of 5) ......................................... 82  
8. Example of FNMILOG Data Set (Part 5 of 5) ......................................... 83  
9. Example of SYSPRINT Data Set (Part 1 of 2) ......................................... 85  
10. Example of SYSPRINT Data Set (Part 2 of 2) ......................................... 86  
11. Location of Panel ID ............................................................ 87  
12. FNM00PRI as Printed in SYSPRINT Using the PRINT Command ................. 88  
13. Problem Determination Facilities Panel (FNM09DBM) ......................... 89  
14. Default Format of Display Main Storage Panel (FNM09DMS) ..................... 90  
15. Display/Modify Diagnostic Trace Options Panel (FNM09DTF) .................. 96  
16. Display/Modify Options Panel (FNM09OPT) .......................................... 98  
17. Display/Modify Operator Trace Panel (FNM09OPR) .............................. 100  
18. Display/Modify Session Trace Panel (FNM09OPS) ............................... 101  
19. Add Diagnostic Trap panel (FNM09TRA) ........................................... 102  
20. Display/Modify Trap Status Panel (FNM09TRS) .................................... 104  
21. Diagnostic Trap - Message Panel (FNM09DT1) .................................... 106  
22. Take an NPM Storage Dump Panel (FNM09DMP) ................................. 107  
23. Abend NPM Panel (FNM09ABN) ................................................ 108  
24. Internal Trace Diagram .......................................................... 127  
25. Example of Locating NPM Internal Trace Table .................................... 128  
26. Command Trace Entry (Type: SC or RC) .......................................... 134  
27. VTAM/VSAM Trace RPL Entry (Type: VE or VS). ................................ 135  
28. VTAM Exit Trace Entry (Type: VX) ................................................ 136  
29. Logic Error Trace Entry (Type: LG) ................................................ 137  
30. LAN Service Trace Entry (Type: LN) ................................................ 138  
31. Presentation Service Trace Entry (Type: PS) ...................................... 139  
32. NPM to NPM Trace Entry (Type: LQ, LR, RQ, RR) .............................. 140  
33. Session Trace Entry (Type: SA) .................................................. 141  
34. Task Start/Task End Trace Entry (Type: KS or KE) .............................. 142  
35. TASK Abend Trace Entry (Type: KA) ...................................... 143  
36. Installation-Wide Exit Trace Entry (Type: IX) .................................... 144  
37. IOB Trace Entry (Type: IO) ..................................................... 145  
38. Intertask Pipe Data Trace Entry (Type: IT) ....................................... 146  
39. APPCIOE Trace Entry (Type: AE) ............................................. 147  
40. Message Trace Diagram ......................................................... 149  
41. Example of Locating Message Trace Table Entries ................................ 150  
42. Message Trace Entry ........................................................... 151
## Tables

1. PDI Structure ................................................................. 73
2. NPM Virtual Storage Allocation ............................................. 74
3. Fields on the Display Main Storage Panel (FNM09DMS) ................... 91
4. Display Main Storage Commands ............................................ 92
5. Display Main Storage Additional commands ................................ 92
6. Display Main Storage Control Block Names ................................ 93
7. Fields in the Diagnostic Trace Options Panel (FNM09DTF) ................ 96
8. Fields in the Display/Modify Options Panel (FNM09OPT) ................. 98
9. Fields in the Operator Trace Panel (FNM09OPR) ............................ 100
10. Fields in the Session Trace Panel (FNM09OPS) ............................. 101
11. Fields in the Add Diagnostic Trap Panel (FNM09TRA) .................... 102
12. Fields in the Trap Status Panel (FNM09TRS) ................................ 104
13. Fields in the Diagnostic Trap - Message Panel (FNM09DT1) .............. 106
14. Fields in the NPM Storage Dump Panel (FNM09DMP) ...................... 107
15. Trap Parameters ........................................................... 125
16. NPM Internal Trace Classes for Traps ........................................ 126
17. Trace Types and Class ....................................................... 129
18. Dispatcher Trace Entry (Basic Format of Each Trace Entry) ............... 132
19. Dispatcher Trace Entry Fields ............................................. 132
20. Message Entry Record (Type: MG) ........................................ 133
21. Message Trace Entry Fields ............................................... 133
22. Command Trace Record Fields ............................................ 134
23. VTAM/VSAM Trace Entry Fields .......................................... 135
24. VTAM Exit Trace Record Fields .......................................... 136
25. Logic Error Trace Record Fields ........................................ 137
26. LAN Service Trace Record Fields ...................................... 138
27. Presentation Service Trace Record Fields ................................ 139
28. NPM to NPM Trace Entry Fields ........................................ 140
29. Session Trace Entry Fields .............................................. 141
30. TASK Start/TASK End Trace Entry Fields ................................ 142
31. TASK ABEND Trace Entry Fields .................................... 143
32. Installation-Wide Exit Trace Entry ...................................... 144
33. IOB Trace Entry ........................................................... 145
34. Intertask Pipe Data Trace Entry .......................................... 146
35. APPCIOE Data Trace Entry ............................................... 147
36. Message Trace Entry Fields ............................................. 151
Preface

Tivoli® NetView Performance Monitor Diagnosis helps you analyze a NetView®
Performance Monitor (NPM) problem, classify it as a specific type of failure, collect data,
and describe the problem to the Support Center.

Although several types of problems create error conditions, this book is primarily concerned
with program errors that are related to NPM. Before using this book, you should be familiar
with the NetView Performance Monitor Installation and Customization book and the NetView
Performance Monitor User’s Guide.

Who Should Read this Book

This book is written for system programmers who use Operating System/390 (OS/390®),
Virtual Telecommunications Access Method (VTAM®), and network control programs
(NCPs) to operate a communication network.

What this Book Contains

This book contains the following sections:

Part I. Classifying and Documenting Problems

Part I contains the following chapters:

- Chapter 1, “NPM Diagnosis Overview”
  Describes the diagnostic process, which consists of classifying a problem and gathering
  information to report the problem to the Support Center. This chapter also describes how
  to search for a known solution to the problem and the steps to take if a known solution
  does not exist.

- Chapter 2, “Diagnosing with Checklists”
  Provides a set of checklists to use when troubleshooting a problem.

- Chapter 3, “Classifying Program Problems”
  Describes how to classify a problem by choosing the keyword that best describes it. It
  also lists additional search arguments and describes when to use them.

- Chapter 4, “Documenting Problems”
  Lists the information needed to understand problems and describes how to obtain this
  information.

- Chapter 5, “Documenting Specific Function Problems”
  Lists the information needed to understand problems for which there is no keyword and
  describes how to obtain this information.

Part II. Using Diagnostic Tools

Part II contains the following chapters:

- Chapter 6, “Using PDI, FNMILOG, SYSPRINT, and PANELID”
  Describes how to use the problem determination information (PDI) control block to
diagnose an abend, how to search the FNMILOG and SYSPRINT data sets for
information about a problem, and how to use the PANELID and PRINT commands to find information about a problem that involves online panels.

- Chapter 7, “Using the Online Problem Determination Facilities”
  Describes how to use the facilities to collect documentation for your problem.

- Chapter 8, “NPM DEBUG and TRAP Initialization Statements”
  Describes these statements and their use in diagnostic tasks.

- Chapter 9, “Using the NPM Internal Trace”
  Provides information about how to find the NPM internal trace table and describes the format of each field in this table.

- Chapter 10, “Using the NPM Message Trace”
  Provides information about how to find the NPM message trace table and describes the format of each field in this table.

Part III. Appendixes

Part III contains the following appendixes:

- Appendix A, “Problem Worksheets”
  Contains worksheets that you can use to record information about NPM problems.

- Appendix B, “Checkpoint Processing for NetWare Resources Collections”
  Contains worksheets that can be used for collecting information about problems.

A glossary and index are provided at the end of the book. A product bibliography is included in the NetView Performance Monitor Concepts and Planning book.

Publications

This section lists publications in the NPM library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

NPM Library

The following documents are available in the NPM library:

- *NetView Performance Monitor Concepts and Planning*, GH19-6961
  Provides information about the basic concepts of NPM and helps you plan for installing NPM and migrating from a previous release.

- *NetView Performance Monitor Installation and Customization*, SH19-6964
  Provides information about installing, customizing, and tuning NPM.

  Explains how to use the graphical interface for NPM.

- *NetView Performance Monitor Reference*, SH19-6965
  Provides information for operators, system programmers, and system planners who want to produce reports and write applications using data collected by NPM.

- *NetView Performance Monitor Messages and Codes*, SH19-6966
  Provides information to help operators and system programmers understand, interpret, and respond to NPM messages and codes.
Preface

  Provides information for new and experienced operators who use NPM on a daily basis to manage a communication network.

- NetView Performance Monitor Diagnosis, LY19-6381
  Provides information about analyzing an NPM problem, classifying it as a specific type of failure, collecting data, and describing the problem to the Customer Support.

- IBM Online Library Omnibus Edition OS/390 Collection Kit, SK2T-6700
  (available December 2001)
  CD containing all OS/390 documentation.

- IBM Online Library z/OS Software Products Collection Kit, SK3T-4270
  (available December 2001)
  CD containing all z/OS documentation.

Related Publications
A product bibliography is included in the NetView Performance Monitor Concepts and Planning book.

Accessing Publications Online
You can access many Tivoli publications online at the Tivoli Customer Support Web site:
http://www.tivoli.com/support/documents/

These publications are available in PDF or HTML format, or both. Translated documents are also available for some products.

Ordering Publications
You can order many Tivoli publications online at the following Web site:
http://www.ibm.com/shop/publications/order

You can also order by telephone by calling one of these numbers:
- In the United States: 800-879-2755
- In Canada: 800-426-4968
- In other countries, for a list of telephone numbers, see the following Web site:
  http://www.tivoli.com/inside/store/lit_order.html

Providing Feedback about Publications
We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:
- Send an e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at the following Web site:
  http://www.tivoli.com/support/survey

Contacting Customer Support
If you have a problem with any Tivoli product, you can contact Tivoli Customer Support. See the Tivoli Customer Support Handbook at the following Web site:
The handbook provides information about how to contact Tivoli Customer Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting support

**Note:** For Tivoli NetView for OS/390 customers only, additional support is available on the NETVIEW CFORUM (Customer Forum) through the IBMLink™ system. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to get resolution.

### Using LookAt to Look Up Message Explanations

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

You can access LookAt from the Internet at:


or from anywhere in z/OS where you can access a TSO command line (for example, TSO prompt, ISPF, z/OS UNIX® System Services running OMVS).

To find a message explanation on the Internet, go to the LookAt Web site and simply enter the message identifier (for example, IAT1836 or IAT*).

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from a disk on your z/OS Collection Kit, SK3T-4269 or from the LookAt Web site. To obtain the code from the LookAt Web site, do the following:

2. Click the News button.
3. Scroll to Download LookAt Code for TSO and VM.
4. Click the ftp link, which will take you to a list of operating systems. Select the appropriate operating system. Then select the appropriate release.
5. Find the lookat.me file and follow its detailed instructions.

To find a message explanation from a TSO command line, simply enter: `lookat message-id`. LookAt will display the message explanation for the message requested.

**Note:** Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in z/OS MVS Routing and Descriptor Codes, SA22-7624. For such messages, LookAt prompts you to choose which book to open.

### Conventions Used in This Book

The term *data set* is used in this book to refer to members of partitioned data sets.
The term z/OS is used in this book to mean z/OS and OS/390 operating systems. Where the term OS/390 does appear, the related information applies only to OS/390 operating systems.

Changes in This Edition

The following changes have been made to this book for NPM Version 2 Release 7:

- NPM version numbers have been updated throughout this book.
- LookAt information has been added to this preface.
- Minor updates have been made to the GTF trace information in the NetView Synergy Interface (NSI) Problems section of Chapter 5, “Documenting Specific Function Problems.”

Changes in the Previous Edition

The following changes were made to this book for NPM Version 2 Release 6:

- Minor updates were made to “Using the Online Problem Determination Facilities” on page 89.
- Minor updates were made to “NPM DEBUG and TRAP Initialization Statements” on page 109.

Using Syntax Diagrams

This book uses syntax diagrams to illustrate the required syntax of commands and statements. This section describes how to use these diagrams.

Reading Syntax Diagrams

The syntax diagrams start with double arrowheads on the left (➡) and move along the main line until you end with two arrowheads facing each other (⬅). To use a syntax diagram, follow any path from left to right. When you reach the end of a line, go to the beginning of the next line, if there is one. For whatever path you choose, code every item that appears on the path. All spaces, commas, and other characters are significant.

Abbreviating Keywords

In a syntax diagram, keywords are all or partly in uppercase. Where an abbreviation is possible, the abbreviation is shown in uppercase and the rest of the keyword is shown in lowercase. Variable values that you provide are shown in italics.

Parameters

The following are types of parameters used in syntax diagrams:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>Required parameters are displayed on the main path.</td>
</tr>
<tr>
<td>Optional</td>
<td>Optional parameters are displayed below the main path.</td>
</tr>
<tr>
<td>Default</td>
<td>Default parameters are displayed above the main path.</td>
</tr>
</tbody>
</table>
Parameters are classified as keywords or variables. Keywords are displayed in uppercase letters and can be typed in uppercase or lowercase. For example, a command is a keyword.

Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a file name is a variable.

In the following example, NSASOLCT is a command, the variable parameter is *ncp_name*, the keyword is CLOCK, and CLOCK’s variable is *time*. You replace the variables with your own values.

```
NSASOLCT ncp_name, CLOCK=time
```

**Required Parameters**

A stack of parameters with the first parameter on the main path means that you must choose only one from the stack.

In the following example, the required parameters are LU, GROUP, CDRM, or APPL.

```
SESSCOLL LU=luname, GROUP=gname, CDRM=cname, APPL=applname
```

**Default and Optional Parameters**

Items shown above the main line are defaults. Items shown below the main line are optional.

```
SEND 'message text', HOST=LOCAL, HOST=hostname
```

The previous diagram shows that if you do not specify a host, HOST=LOCAL is used. To send a message to a different host, for example NYC, code the SEND command as follows:

```
SEND 'message text', HOST=NYC
```

**Repeating Parameters**

Items that can be repeated are shown as follows:

```
CRITERIA, 'expression'
```

The previous diagram shows that the following are all valid ways of coding the CRITERIA statement:

- CRITERIA
- CRITERIA 'expression'
- CRITERIA 'expression1','expression2'
- CRITERIA 'expression1','expression2','expression3'
- CRITERIA 'expression1','expression2','expression3','expression4'
and so on.

**Reading Fragments**

Syntax diagrams can contain fragments. A fragment is indicated by vertical bars with the name of the fragment between the bars. The fragment appears after the main diagram, as shown in the following example.

```
>>> SE 'message text' Route
```

**Route:**

```
,ROUTE=GLOBAL
```

```
,ROUTE=ALL
  ,ROUTE=CONSOLE
  ,ROUTE=EXTERNAL
```

The previous diagram shows that the following are all valid ways of coding the SEND command:

- `SE 'message text'
- `SE 'message text',ROUTE=GLOBAL`
- `SE 'message text',ROUTE=ALL`
- `SE 'message text',ROUTE=CONSOLE`
- `SE 'message text',ROUTE=EXTERNAL`

**Long Syntax Diagrams**

When more than one row is needed for a syntax diagram, the continued line ends with a single arrowhead (▷) and following line begins with a single arrowhead (◁), as shown in the following example.

```
>>> FILE NAME=dd_name
  ,BUFND=2
  ,BUFND=nnn
  ,BUFNI=2
  ,BUFNI=nnn
  ,COUNT=99999999
  ,COUNT=nnnnnnnn

  ,ISTATUS=ACTIVE
  ,ISTATUS=INACTIVE

  ,MACRF=NSR
  ,MACRF=LSR

  ,USE=OUTPUT
  ,USE=READONLY

  ,WRAP=NO
  ,WRAP=YES
```


Part I. Classifying and Documenting Problems

Chapter 1. NPM Diagnosis Overview ................................................................. 5
Using Checklists ............................................................................................... 6
Classifying Problems ........................................................................................ 6
Gathering Information about Problems ............................................................. 6
   Using Traces ................................................................................................. 6
   Using Traps .................................................................................................. 6
   Using FNMILOG and SYSPRINT Data Sets .................................................. 7
   Using the PANELID and PRINT Commands .................................................. 7
Reporting Problems to the Support Center ....................................................... 7
Searching the Software Support Database for Possible Solutions .................... 7
Opening APARs with the Support Center .......................................................... 8

Chapter 2. Diagnosing with Checklists ............................................................. 9
Batch Reporting Utility Checklist ...................................................................... 10
Dynamic Definite Response (DDR) Function Checklist ...................................... 11
Dynamic Network Collection (DNC) Checklist .................................................. 12
LAN Bridge Collection Checklist ...................................................................... 13
LAN Segment Utilization Data Collection Checklist ......................................... 14
NCP Definition Checklist .................................................................................. 15
NetWare Resources Collection Checklist ........................................................ 16
Network Collection and Accounting Checklist .................................................. 18
NPM EXECs Checklist ...................................................................................... 19
NPM Performance Alerts Checklist ................................................................... 20
ODLC LAN Resource Data Collection Checklist .............................................. 21
RTM Collection Checklist (MVS) ...................................................................... 22
Session Collection Checklist ............................................................................ 23
System Console Support Checklist .................................................................... 24
System Management Facility (SMF) Collection Checklist .................................. 25
Transit Analysis Program (TAP) Checklist ........................................................ 26
VTAM Statistics Collection Checklist ................................................................ 27

Chapter 3. Classifying Program Problems ....................................................... 29
Using Keywords to Describe Problems ................................................................ 29
Choosing the Correct Keyword ......................................................................... 29
   Abnormal End Symptoms .......................................................................... 30
   Documentation Problem Symptoms ............................................................ 30
   Incorrect Output Symptoms ........................................................................ 31
   Loop Symptoms .......................................................................................... 32
   Message Symptoms ...................................................................................... 32
   Performance Problem Symptoms ................................................................. 33
   Wait Symptoms ............................................................................................ 34
Choosing Additional Keywords .......................................................................... 34

Chapter 4. Documenting Problems ................................................................... 35
Beginning to Document Problems ...................................................................... 35
   Information Needed for Most Problems ......................................................... 35
Chapter 5. Documenting Specific Function Problems .......................... 47

Archive Data Management Problems .............................................. 48
  Information Needed ................................................................. 48
  Collecting the Information ...................................................... 48
Batch Reporting Utility (FNMREPRT) Problems ............................. 49
  Information Needed ................................................................. 49
  Collecting the Information ...................................................... 49
Dynamic Definite Response (DDR) Problems .................................. 50
  Information Needed ................................................................. 50
  Collecting the Information ...................................................... 50
LAN Bridge Collection Problems .................................................. 51
  Information Needed ................................................................. 51
  Collecting the Information ...................................................... 51
LAN Segment Utilization Data Collection Problems ....................... 52
  Information Needed ................................................................. 52
  Collecting the Information ...................................................... 52
NCP Definition Problems ............................................................ 53
  Information Needed ................................................................. 53
  Collecting the Information ...................................................... 53
NetWare Resources Collection Problems ...................................... 54
  Information Needed ................................................................. 54
  Collecting the Information ...................................................... 54
Network Collection Problems ...................................................... 56
  Information Needed ................................................................. 56
  Collecting the Information ...................................................... 56
NetView Synergy Interface (NSI) Problems .................................... 57
  Information Needed ................................................................. 57
  Collecting the Information ...................................................... 57
This book is intended to help you isolate the cause of NPM program failures. Figure 1 illustrates the process for diagnosing program failures. Follow this process to classify your NPM problem, gather information about it, and report the problem to the Support Center.

**Figure 1. Diagnostic Procedure Flowchart**

This chapter provides an overview of the diagnostic process and describes what happens to your problem after you report it to the Support Center.
Using Checklists

Before you classify or collect information about an NPM problem, see the appropriate checklist in "Diagnosing with Checklists" on page 9. Use the checklists as a preliminary step in the diagnosis process.

Classifying Problems

The first step in the diagnosis process is to classify the type of problem you are experiencing. There are seven types of possible failures:

- Abnormal end
- Documentation
- Incorrect output
- Loop
- Message
- Performance
- Wait

These failures are described by keywords which you compile into a keyword string. To do this, follow the instructions in "Classifying Program Problems" on page 29. This keyword string is used by a Support Center representative to search the Information/Access (RETAIN®) database to discover if your problem, or a similar one, was previously reported.

Gathering Information about Problems

After you have classified your problem, you must gather information about it. "Documenting Problems" on page 35 lists the documentation you should have for each type of problem before calling the Support Center. The following tools are available to help you gather information and isolate the cause of program problems:

- Traces
- Traps
- FNMILOG and SYSPRINT data sets
- PANELID and PRINT commands

Using Traces

NPM provides several different types of trace options that you can enable or disable through the DEBUG initialization statement. These options are described in "NPM DEBUG and TRAP Initialization Statements" on page 109. You can also modify the trace options set at initialization using the Problem Determination panel (FNM09DBM). See "Using the Online Problem Determination Facilities" on page 89 for more information.

NetWare for SAA® provides trace facilities in addition to the ones provided by NetView. NetWare users should refer to the appropriate Novell documentation.

Using Traps

NPM also provides several types of traps that use the NPM internal trace table. These traps show the following:

- Type of trace event to be checked
- Data used for comparisons
- Number of times the trap has met the requirements
- Action to be taken if a trap is found to meet the requirements

Use these traps to provide dumps or to abnormally end NPM. For more information about each type of trap, see "NPM DEBUG and TRAP Initialization Statements" on page 109.
Using FNMILOG and SYSPRINT Data Sets

The FNMILOG data set contains a listing of the initialization statements from the initialization data set. You can use FNMILOG to check your coding of initialization statements and the values you used in them. Specify the type of report that you want printed in the NPM initialization statement.

If you specify LIST=YES, NPM lists each initialization statement and its parameters.

If you specify CONFIG=YES, NPM summarizes the configuration established during initialization.

See “Using the FNMILOG and SYSPRINT Data Sets” on page 74 for more information.

The SYSPRINT data set keeps a log of messages issued while NPM is running. Use this SYSPRINT to track error messages and identify problems. SYSPRINT also shows panels printed using the PRINT command.

Using the PANELID and PRINT Commands

NPM provides two commands to help you identify and print panels that seem to be causing NPM errors:

**PANELID**

Helps you identify the panel that NPM was displaying when the problem occurred. When you issue this command, NPM displays the panel name in the top left corner of the current panel and all subsequent panels. See “Diagnosing with the PANELID Command” on page 87 for more information.

**PRINT**

Prints data that appears in panels. When you use this command, NPM copies the contents of your current panel into the SYSPRINT data set. See “Diagnosing with the PRINT Command” on page 87 for more information.

Reporting Problems to the Support Center

After you have classified your problem and gathered the necessary information, report it to the Support Center. The Support Center is the first point of contact for NPM customers who need help with a program problem after installation is complete. If you encounter a problem during installation, contact your local marketing systems engineer for assistance.

When you call the Support Center, have your customer identification ready, such as your account name, access code, and program number. The Support Center determines the type of help that you need, assigns you a problem number, and refers you to a Support Center representative.

Searching the Software Support Database for Possible Solutions

When you reach your Support Center representative, describe your problem. The representative uses this information to form a keyword string and then searches RETAIN, an IBM® database containing problem symptoms and resolutions, for known problems and for problems currently under investigation. The representative might ask for more information to produce other keywords to help in the search.
If you have Information/Access, an IBM-licensed program, you can search the RETAIN database yourself for similar problems and their solutions based on your keyword string. See “Classifying Program Problems” on page 29 for more information about using keywords.

If a similar problem description is found in the database search, a solution is probably available. The Support Center representative can help initiate the solution. If the search does not produce a solution, the representative ensures that you have the necessary information to discuss the problem with an NPM specialist. Your call is then referred to an NPM specialist who asks you for more information and performs additional searches of the database. If the problem appears to be a new problem, the NPM specialist creates an authorized program analysis report (APAR).

**Opening APARs with the Support Center**

If an NPM specialist creates an APAR for your problem, you are asked to send documentation about your problem to IBM. Write the APAR number in the upper-right corner of each piece of documentation you send.

The APAR, problem description, and your documentation permit the NPM specialist to examine the problem in greater detail to develop a solution. If on-site assistance is necessary to develop a solution, IBM dispatches an operational systems engineer to your site.

Once a solution is found, the Support Center provides you with a temporary or bypass solution for you to test. If the solution resolved the problem, the support center builds a program temporary fix (PTF), and all the information about the solution is entered in the RETAIN database. This procedure keeps the database’s problem descriptions and solutions current and ensures that the information is available for future RETAIN searches.
Diagnosing with Checklists

This chapter provides the following set of checklists to help you resolve your problem. Before starting the diagnosis process, use the appropriate checklist for the NPM function that is causing your problem.

- Batch reporting utility (FNMREPRT)
- Dynamic definite response (DDR)
- Dynamic network collection (DNC)
- LAN bridge collection
- LAN segment utilization data collection
- NCP definition
- NetWare resources collection
- Network collection
- NPM EXECs
- ODLC LAN resource data collection
- Performance alerts
- Response time monitor (RTM) collection
- Session collection
- System console support
- System management facility (SMF) collection
- Transit analysis program (FNMTAP)
- VTAM statistics collection
Batch Reporting Utility Checklist

Use this checklist to ensure that batch reporting is correctly set up to report on your data.

- Did you set up default substitution parameters in the FNMRPPRC procedure? This should be done only once at installation.

- Did you override the default substitution parameters set in the FNMRPPRC procedure? You can override substitution parameters that are used in any of the OUTPUT, INPUT, or SYSIN DD statements.

- In the FNMRPPRC procedure, are OLRECL and OBLK set as shown in the following example?
  
  - OLRECL=linesize (if ASA=NO)
  - OLRECL=linesize+1 (if ASA=YES)
  - OBLK=multiple of OLRECL

- Did you check SYSMSG for any system messages?

- Did you check the RPTLOG data set? This data set holds the INFOR, WARNING, and ERROR messages logged by FNMREPRT. Check the data set’s contents, especially if the FNMREPRT JCL ends with RC=1. The message, if any, will explain the reason.

- If you are merging multiple logs, did you sort the input records by the date and time fields contained in the system management facilities (SMF) header portion of the records?

- If DELIMS=YES, is LINESIZE large enough to contain the additional columns added by the delimiters? Two columns (beginning and ending delimiters) are added for each nonnumeric field.

- If you are not getting any data records written to output:
  - Does the input data set contain records of the correct subtype for the report that you chose? You can check the subtype summary at the end of the output report.
  - Do the DATES and TIMES keywords of the SELECT statement in SYSIN correspond to the dates and times of the records in the input data set?
  - Do any records in the input data set meet the criteria coded on the CRITERIA statements in SYSIN?

- If you installed user exits, are they correct?

- Did you modify any IBM-supplied files other than SYSIN or LISTS? If so, run FNMREPRT using the unmodified version of the files.

- Is the syntax of the SYSIN data set correct?
Dynamic Definite Response (DDR) Function Checklist

Use this checklist to ensure that you correctly set up the DDR function.

- Is RSP=YES coded on the APPL statement?
- Is DDRBUF greater than or equal to the following?
  \[(\text{number of concurrent DDR sessions} \times \text{number of transactions per minute}) \div 10\]

  **Note:** DDRBUF is not used with VTAM V3R4.1 or later.

- Is SESSH=R specified on the SESSCOLL command? Is SESSION STATISTICS=R specified on the Start Session panel (FNM02SCL), fast path =2.1?
- Is SESSH=R coded on the GENERIC statement? If so, message FNM613E is issued.
Dynamic Network Collection (DNC) Checklist

Use this checklist to ensure that you correctly set up DNC to collect network data.

- Is FNMCAAX3 or FNMCAAX4 installed? For information about installing the required NPM exit routine, refer to the NetView Performance Monitor Reference book.
- Is the NPM VTAM session manager exit, ISTEXCAA, installed in SYS1.VTAMLIB?
- Is HOSTCOLL=YES specified on the NPM initialization statement?
- Is DNC=YES specified on the CONTROL initialization statement?
- Is DNC=YES specified on the NCP command for network resource collection?
- Is DYINT=n specified on the NCP command for interval collection? The default is DYINT=2.
- Are the terminals and LUs that are to have data collected generated with NPACOLL=YES in the NCP resource definitions?
- If you are using the GENERIC, LINE, PU, or LU statement in the FNMSCMDS data set, is DNC=YES or DNC=NO specified correctly?
- Is DNC=NO specified on the APPL initialization statements for applications, such as NETMON, for which you do not want data collected?
- Is DNC=(YES,PASS) specified on the APPL initialization statements for applications, such as TSO and the NetView program, which use CLSDST OPTCD=PASS to pass a session from one application to another?
- If you are using the installation-wide exit for DNC authorization, is the BEGIN function flag set to call it?
- If you are using the NPM installation-wide exit for DNC authorization, is it processing correctly?
- Are all primary NPALUs active?
LAN Bridge Collection Checklist

Use this checklist to ensure that you correctly set up LAN bridge collection.

- Is the NetView CNMLINK library concatenated to the STEPLIB DD statement in the NPM startup procedure JCL as shown in the following example?

  //STEPLIB DD DSN=&HLQ.V2R7M0.SFNMLMD1,DISP=SHR
  // DSN=SYS1.CNMLINK,DISP=SHR

- Is the NPM LINKLIB library concatenated to the STEPLIB DD statement in the NetView startup procedure JCL as shown in the following example?

  //STEPLIB DD DSN=&SQ1.VTAMLIB,DISP=SHR
  // DSN=NPM.V2R7M0.SFNMLMD1,DISP=SHR

- Is NetView properly installed? NetView must be set up correctly to send and receive data between NPM and the NetView program. For information about installing NetView, see the *NetView Installation and Administration Guide* for your operating system.

- Is NetView running on the same host as this NPM?

- Verify that the following message was logged in NPM’s SYSPRINT file:

  FNM170I LAN MANAGEMENT SUPPORT ENABLED

  This message confirms that the NetView CNMCNETV module was successfully loaded at NPM initialization.

- Are you using NetView V1R3 or a later release for MVS?

- Is the NPM-supplied NetView autotask (FNMAUT01) active? To display the status of the autotask, enter the following command in the NetView operator console:

  LIST FNMAUT01

- Is the NPM-supplied NetView DSI6DST task active? To display the status of the task, enter the following command in the NetView operator console:

  LIST DSI6DST

- Is the NPM-supplied NetView FNMSYRN task active? To display the status of the task, enter the following command in the NetView operator console:

  LIST FNMSYRN

- Is LANCOLL=YES specified in the NPM initialization statement in FNMINIT?
LAN Segment Utilization Data Collection Checklist

Use this checklist to ensure that you correctly set up LAN segment utilization data collection.

- Is the NetView CNMLINK library concatenated to the STEPLIB DD statement in the NPM startup procedure JCL as shown in the following example:
  ```
  //STEPLIB DD DSN=&HLQ.V2R7M0.SFNMLMD1,DISP=SHR
  // DSN=SYS1.CNMLINK,DISP=SHR
  ```

- Is the NPM LINKLIB library concatenated to the STEPLIB DD statement in the NetView startup procedure JCL as shown in the following example:
  ```
  //STEPLIB DD DSN=&SLOC1.VTAMLIB,DISP=SHR
  // DSN=NPM.V2R7M0.SFNMLMD1,DISP=SHR
  ```

- Verify that the following message was logged in NPM’s SYSPRINT file:
  ```
  FNM170I LAN MANAGEMENT SUPPORT ENABLED
  ```
  This message confirms that the NetView CNMCNETV module was successfully loaded at NPM initialization and that it is running on the same host as this NPM.

- Are you using NetView V1R3 or a later release?

- Is the NPM-supplied NetView autotask (FNMAUT01) active? To display the status of the autotask, enter the following command in the NetView operator console:
  ```
  LIST FNMAUT01
  ```

- Is the NPM-supplied NetView DS16DST task active? To display the status of the task, enter the following command in the NetView operator console:
  ```
  LIST DS16DST
  ```

- Is the NPM-supplied NetView FNMSYRN task active? To display the status of the task, enter the following command in the NetView operator console:
  ```
  LIST FNMSYRN
  ```

- Is LANCOLL=YES specified in the NPM initialization statement in FNMINIT?

- Is LAN Network Manager Version 1.1 set up to support the segment utilization command?

- Is a special hardware adapter installed on all of the token rings from which segment utilization data is collected?

  **Note:** You must have Operating System/2® 1.3 with Extended Service 1.0 (OS/2 1.3 ES 1.0) or a later version of OS/2®.

- Did you specify a LSEGCOLL command with a valid segment or LAN Manager name?

  **Note:** The SEGMENT and LANMGRNM parameters are mutually exclusive. For more information about the LSEGCOLL command, see the *NetView Performance Monitor Installation and Customization* book.
NCP Definition Checklist

Use this checklist to ensure that NCPs are properly defined.

- Is the NCP resource resolution table (RRT) used to define the NCP to the system in the NPM FNMLLIB or STEPLIB data set?

- Is the NCP VTAMLST member used to define the NCP to the system in one of the VTAMLST data sets?

**Note:** If you use OPTION NEWDEFN=YES to generate your NCP, the VTAMLST member used to define the NCP to NPM should be the NEWDEFN member, not the NCP source file.

- If you are using an NPM EXEC to issue the NPM NCP command, is the command coded correctly?

- Did you receive any syntax error messages?
NetWare Resources Collection Checklist

Use this checklist to ensure that you correctly set up NetWare resources data collection.

- Is the NetView CNMLINK library concatenated to the STEPLIB DD statement in the NPM startup procedure JCL as shown in the following example?
  ```
  //STEPLIB DD DSN=&HLQ.V2R7M0.SFNMLMD1,DISP=SHR
  // DD DSN=SYS1.CNMLINK,DISP=SHR
  ```

- Is the NPM LINKLIB library concatenated to the STEPLIB DD statement in the NetView program startup procedure JCL as shown in the following example?
  ```
  //STEPLIB DD DSN=&HLQ.V2R7M0.SFNMLMD1,DISP=SHR
  // DD DSN=SYS1.CNMLINK,DISP=SHR
  ```

- Is NetView properly installed? NetView must be set up correctly to send and receive data between NPM and the NetView program. For information about installing the NetView program, see the NetView Installation and Administration Guide for your operating system.

- Is NetView running on the same host as this NPM?

- Verify that the following message was logged in NPM’s SYSPRINT file:
  ```
  FNMI701 NETWARE SUPPORT ENABLED
  ```

  This message confirms that the NetView CNMCNETV module was successfully loaded at NPM initialization.

- Are you using NetView V1R3 or a later release?

- Is the NPM-supplied NetView autotask (FNMAUT01) active? To display the status of the autotask, enter the following command in the NetView operator console:
  ```
  LIST FNMAUT01
  ```

- Is the NPM-supplied NetView task DSICRTR active? To display the status of the task, enter the following command in the NetView operator console:
  ```
  LIST DSICRTR
  ```

- Is NWCOLL=YES specified on the NPM initialization statement in FNMINIT?

- Is NetWare for SAA or the NetWare Management Agent for NetView properly installed? Enter the following command in the NetView operator console:
  ```
  RUNCMD SP=nwsaa_sp APPL=NETWARE OP=NPM SNAME=nw_sname QUERY SERVER TIME
  ```

  where:

  - `nwsaa_sp` is the service point corresponding to the server containing the NetWare for SAA software.
  - `nw_sname` is the name of the NetWare server containing the NPM NetWare Agent.

  If you get a message back giving the time on the server, the software is correctly installed. Otherwise, you obtain an error message and act accordingly.

- Is the NPM NetWare Agent properly installed? Enter the following command in the NetView operator console:
  ```
  RUNCMD SP=nwsaa_sp APPL=NETWARE OP=NPM SNAME=nw_sname LOAD NLM NLMNAME=IBMNPMA
  ```

  where:
The service point corresponding to the server containing the NetWare for SAA software is the \textit{nwsaa_sp}. The name of the NetWare server containing the NPM NetWare Agent is \textit{nw_sname}.

Check the message obtained against the list below and act accordingly:

- If the message states either that the NPM NetWare Agent is already loaded or was loaded successfully, unload it by entering:
  \begin{verbatim}
  RUNCMD SP=nwsaa_sp APPL=NETWARE OP=NPM SNAME=nw_sname UNLOAD NLM NLMNAME=IBMNPMAG
  \end{verbatim}
- Any other message indicates an error condition and you act accordingly.

Have you just started NPM after an ABEND or termination? If so, NPM performs checkpoint processing if it found any active NetWare collections. For more information, see \textit{"Checkpoint Processing for NetWare Resources Collections" on page 165}.

Have you received message \textit{FNM861I} with request type (RQ)=14 and return code (RC)=35? You receive this message when the program-to-program interface (PPI) buffer is full. The PPI enables NPM to send data buffers to or receive data buffers from NetView. The most likely explanation for receiving this message is that you still have active NetWare collections although NPM is down. The NPM NetWare Agent sends collection data to NPM on an unsolicited basis until the collection is completed or stopped. Issue an LWGSTOP command from the system console to stop all active NetWare collections when NPM is restarted. For more information about other request type and return code combinations, refer to the \textit{NetView Application Programming Guide}. 
Network Collection and Accounting Checklist

Use these checklists to ensure that you correctly set up network collection, network session accounting (NSA), or network gateway accounting (NGA).

Network Collection:
- Is the NPALU coded in the NCPGEN statements?
- Is the NPALU bound to NPM?
- Is NPACOLL=YES specified for resources for which you want network collection?
- Is NPA=YES specified in the BUILD macro for the NCP?
- Has the NCP been generated?

NSA and NGA:
- Is the name of the backup NPALU specified with the NSA option?
- Is NSA=SMF or NSA=NPMLOG specified on the NPM initialization statement?
- Is NSA=YES specified in the PROFILE statement to authorize operators to use online NSA panels?
- Is FNMSTRTR coded with NSACNTL, NSAMODY, or NSASOLCT to meet your installation requirements?
- For NSA collection, is the SESSACC option specified on the NCP BUILD statement and the NCP source?
- For NGA collection, is the GWSESAC option specified on the NCP BUILD statement and the NCP source?
- After all the updates were added to the NCP source, was the NCP regenerated and the CCU reloaded?
NPM EXECs Checklist

Use this checklist to ensure your NPM EXECs are correctly defined.

- Is the NPM EXEC member in one of the FNMSCMDS data sets?

   **Note:** If the FNMSCMDS data set is defined with extents, you might need to compress the data set before running the NPM EXEC.
NPM Performance Alerts Checklist

Use this checklist to verify that you correctly installed NPM performance alerts.

- Is NetView present in the same host CPU as the copy of NPM that provides performance alerts?

- Is NetView properly installed? NetView must be set up correctly to receive alerts from NPM. For more information about setting up the NetView program to receive alerts, see the *NetView Installation and Administration Guide* for your operating system.

- Is ALERT=YES specified on the NPM initialization statement?

- Is ALERT=YES specified on the TASK initialization statement, if present?

- If you are collecting bridge or segment data, specify LANX=ALERT on the NPM statement. You should also check the LAN Bridge Control panel (FNM02LBB), fast path=9.9.1, or the LAN Segment Control panel (FNM02LSB), fast path=9.9.2, to ensure that NetView is specified as a destination for monitor exceptions. Ensure that LANX=ALERT is specified as a destination for monitor exceptions.

- Does NPM have access to the NetView CNMLINK library?

  Is the NetView CNMLINK library concatenated to the STEPLIB DD statement in the NPM startup procedure JCL as shown in the following example?

  ```
  //STEPLIB DD DSN=&HLQ.V2R7M0.SFNMLMD1,DISP=SHR
  // DD DSN=SYS1.CNMLINK,DISP=SHR
  ```

  Ensure that the second line is added as shown here to the existing STEPLIB DD statement in the NPM startup procedure.

- Have you specified reasonable monitor thresholds? You can flood your system with alerts or effectively disable the alert function by specifying an extreme value. For guidelines, see the *NetView Performance Monitor User’s Guide*.

- Are you using an NPM installation-wide exit to process alerts? You can discard or modify performance alerts with the exit. Be careful when you modify the NMVT alert. NPM performs a simple verification on the alerts and discards any alerts that are not correctly defined.

**Note:** Message FNM173E is issued if problems occur during the processing of alerts. Check the performance measurement table (PMT) for the number of alerts as follows:

- **Total** The total number of alerts.
- **Sent** The number of alerts that the NetView program accepted.
- **Lost** The number of alerts that NPM attempted to send, but could not send.

The difference between the total number of alerts and the combination of alerts sent and lost is the number of alerts that were discarded.
ODLC LAN Resource Data Collection Checklist

Use this checklist to ensure that you correctly set up the collection of data from the ODLC LAN resources.

- Is NPACOLL=YES or NPACOLL=(YES,EXTENDED) coded in the NDFGEN definition for all resources on which the collection is taken? Also, NPACOLL=YES should be coded in the GROUP macro for all logical PUs.

- If you are collecting data for an ODLC LAN token ring, is the ECLTYPE parameter correctly specified on the GROUP statement? For more information about how to code the ODLC LAN token ring, see the *NCP, SSP, and EP Resource Definition Reference*.

- Ensure that the monitor types PUDSDISC and RTIMEOUT are coded in the START command.

- Ensure that the communication control unit (CCU) that you selected to monitor is active. Check the Session Display and Management panel (FNM02OSS), fast path =9.2.

- Is the 3746 Model 900 attachment properly set up and running NCP V6R2 or a later release? Are you running NPM V2R7?

- If you issued a NETCOLL command, did you specify an NPM time interval that is a multiple of the 3746 Model 900 attachment forwarding interval time (3 minutes and 45 seconds)?

- Is NPA=(YES,DR) coded in the BUILD macro of the NCP initialization statement?
Use this checklist to verify that you correctly set up response time monitor (RTM) collection.

- Is HOSTCOLL=YES coded on the NPM statement?
- Is RTM=YES specified on the CONTROL initialization statement?
- Is FNMU83X2 installed and at the current PTF level? Because RTM information is not written into SMF type 28 records, the NPM RTM Analysis panels do not operate without this user exit.
- Is RTM=SESSION or RTM=NPMLOG specified on the NPM initialization statement?
- Is RTM=YES specified on the SMFCNTL initialization statement? The default is NO.
- Have you checked that RTM events are not zero on the following panels:

<table>
<thead>
<tr>
<th>Fast Path</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.4</td>
<td>FNM020PM</td>
<td>Performance Measurement Table Display</td>
</tr>
<tr>
<td></td>
<td>FNM02OPR</td>
<td>PMT File Services Summary</td>
</tr>
<tr>
<td></td>
<td>FNM02OPC</td>
<td>Session Collection Performance.</td>
</tr>
</tbody>
</table>

You can access FNM02OPM by using fast path =9.4. You then access FNM02OPR by pressing PF8 (NEXT) from FNM02OPM, then PF8 (NEXT) from FNM02OPS. You access FNM02OPC by pressing PF8 (NEXT) from FNM02OPR, then PF8 (NEXT) from FNM02OPV.

- Was SMF collection initialized? You can check this by entering:

  =P.1;DUMP;MAT

  on the command line in NPM’s Online Problem Determination Facilities. Offset X’15’, with value either X’84’ or X’8C’, indicates whether SMF collection was initialized and RTM collection activated.
Session Collection Checklist

Use this checklist to verify that you correctly set up session collection.

- Have you followed the correct naming conventions when naming the NPM start task procedure? Do not name the procedure with the same name defined in the NPM initialization statements or the same name as an operator ID to identify initialization commands.

If the names are conflicting, you might see any or all of the following three messages:
- FNM729I
- FNM277I
- FNM007W

- If you are using VTAM 3.4.1 or a later release, is the NPM load library concatenated with the VTAM load library in the VTAM startup procedure? For more information, see the *NetView Performance Monitor Installation and Customization* book.

- Is HOSTCOLL=YES specified on the NPM initialization statement?

- If NPM is collecting data, but not showing it on the session analysis panels, verify that SESSION=SESSION is specified on the NPM initialization statement.

- Is NPM defined to VTAM as shown in the following example?

```
'name of NPM' APPL AUTH=(SPO,ACQ,CNM)
```

- Are the base INTERVAL and SESSINT values coded correctly on the NPM initialization statement? NPM writes session statistics by multiplying SESSINT by the base INTERVAL. Collection could be occurring but is written only once each hour.

- Is the application used in the APPL statement that defines this application the same as the application name used to open the VTAM access method control block (ACB)? For example, if the application statement used to open the VTAM ACB is:

```
VTAMLST: VBUILD TYPE=APPL
* PRODAPP1 APPL AUTH=(ACQ,PASS),
      MODETAB=AMODETAB
      ACBNAME=CICSPRD1
```

the NPM APPL statement must be:

```
APPL NAME=CICSPRD1
```

If you are using TSO or any application that opens an ACB (ACBNAME=) with a name other than the name specified on the NPM application statement, use the SYN parameter to permit NPM to resolve the application name. If TSO is accessed by a session manager using the NetView synergy interface (NSI), add a second definition of TSO for the network name of TSO. For more information about the APPL statement, see the *NetView Performance Monitor Reference* book.

- If you are using a session manager, such as NetView Access Services, is NSI=YES specified on the NPM initialization statement and is TYPE=NVAS specified on the APPL statement that defines the manager?
System Console Support Checklist

Use this checklist to verify that you correctly set up system console support.

- Is COMMAND=YES specified on the console statement to enable the MODIFY and STOP system console commands?

- Does the identifier in the command match the START command used when NPM was started, as shown in the following examples?
  
  **Example 1:**
  
  ```
  S FNMSNPM
  F FNMSNPM,D OPER
  ```
  
  **Example 2:**
  
  ```
  S FNMSNPM,NPMA01,INIT=A01INIT
  F NPMA01,D OPER
  ```
# System Management Facility (SMF) Collection Checklist

Use this checklist to verify that you correctly set up the System Management Facility (SMF).

- Is the NPALU active? Check the NPALU Display and Management panel (FNM02OCC), fast path =9.3, for a list of active NPALUs.
- Is HOSTCOLL= YES coded on the NPM statement?
- Is user-exit FNMU83X1 installed? The NPM SMF Management panels do not operate without this user exit. For installation information, see the *NetView Performance Monitor Reference*.
- Is user-exit FNMU83X2 installed? The NPM RTM Analysis panels do not operate without this user exit because RTM information is not written into SMF type 28 records.
- In SYS1.PARMLIB for SMFPRMxx, is record type 28 authorized to NPM in STC class?
- Is NPM defined correctly to SMF? Verify that SYSPARM and SUBSYSPARM have IEFU83 and IEFU84 coded as EXITS. For more information, see the *OS/390 MVS Extended Addressability Guide* and the SMF libraries.
- Does the SMF type initialization statement in the SMF control member of the FNMPARM partitioned data set specify the requested SMF record type number? For more information, see the *NetView Performance Monitor Installation and Customization* book.
- Is the IEFU83 installation-wide exit different from the IBM-supplied version? Is it screening out SMF record type 28?
- Is the FNMUEXIT installation-wide exit different from the IBM-supplied version? Is it screening out SMF record type 28?
- Have you checked the following panels for zero record counters?

<table>
<thead>
<tr>
<th>Fast Path</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=9.4</td>
<td>FNM02OSM</td>
<td>SMF User Records Display and Management</td>
</tr>
<tr>
<td></td>
<td>FNM02OPR</td>
<td>PMT File Services Summary</td>
</tr>
</tbody>
</table>

You access FNM02OPR by pressing PF8 (NEXT) from the Performance Measurement Table Display (FNM02OPM), fast path =9.4, then PF8 (NEXT) from FNM02OPS.

- Are the subtypes writing to SMF? Are NETWORK=SMF, SESSION=SMF, or both of these definitions specified in the NPM initialization statement?
- Is NETWORK=SMF specified when the NETCOLL command is issued for any of the following?

- NCP Causes subtype X’10’ to be written.
- LINE Causes subtype X’11’ to be written.
- PU Causes subtype X’12’ to be written.
- LU Causes subtype X’13’ to be written.
Transit Analysis Program (TAP) Checklist

Use this checklist to verify that the transit analysis program is correctly installed.

- Check the summary statistics at the end of the SYSPRINT file. If NPM or VTAM trace records are not found in the input, check the value of the INPUT keyword.

- If records were not selected for processing, use SUMMARY=YES on the TAP control statement to list all the session partners that are read from the input file. Check if the particular session is in the file.
VTAM Statistics Collection Checklist

Use this checklist to verify that the VTAM statistics collection function is correctly installed.

- Are the NPM APF-authorized? For example, FNMMAIN must be linked and all libraries from which NPM modules are loaded must be in the APF list.

- Have you specified VTAMCOLL=YES on the NPM initialization statement? VTAMCOLL=NO is the default.

- Have you added a TASK statement in FNMINIT for the FNMVDC00 subtask? Is its priority set to 255?

- Have you added the NETCOLL and NETANLY keywords to the PROFILE initialization statement?

- Have you specified the TNSTAT parameter in VTAM’s start option list? Ensure that you place the TIME=1440 parameter after the TNSTAT parameter.

- Ensure that you did not issue the MODIFY TRACE,TYPE=SMS,ID=VTAMBUF command while VTAM is running or place it in the ATCSTRxx member.

- Ensure that you did not issue the DISPLAY BFRUSE command while VTAM is running.
When you start the diagnosis process, the first step is to determine the type of problem you have. When you classify the problem, you know the type of information you need.

To classify a program problem, read the symptoms for each type of problem listed in this chapter and choose the keyword that best describes the problem you are experiencing. If you cannot identify the problem type, see “Beginning to Document Problems” on page 35 for more information.

Using Keywords to Describe Problems

A keyword is a set of alphanumeric characters that defines one aspect of a program failure. When you have a program problem, you need to choose a keyword based on the symptoms of your problem and use it to build a search string. The keyword should be the one that best describes the type of failure you are experiencing. To assign a keyword to your problem, match the symptoms of your problem with the symptoms listed for each keyword.

You can use a string of keywords to completely describe your problem. For example, include a program and release number keyword in your keyword string. These keywords identify the IBM-licensed program that failed. The program number that applies to NPM Version 2 is 5655-043. The release number is R7. Other related keywords are described later in this chapter.

Choosing the Correct Keyword

You can use one of the following seven keywords to classify your problem:

<table>
<thead>
<tr>
<th>This Keyword</th>
<th>Relates to</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABEND</td>
<td>Abnormal End</td>
</tr>
<tr>
<td>DOC</td>
<td>Documentation</td>
</tr>
<tr>
<td>INCORROUT</td>
<td>Incorrect Output</td>
</tr>
<tr>
<td>LOOP</td>
<td>Loop</td>
</tr>
<tr>
<td>MSG</td>
<td>Message</td>
</tr>
<tr>
<td>PERFM</td>
<td>Performance</td>
</tr>
<tr>
<td>WAIT</td>
<td>Wait</td>
</tr>
</tbody>
</table>

The remainder of this chapter describes the symptoms for each of these problem types and provides additional keywords to add to your keyword string.
Abnormal End Symptoms

Choose the ABEND keyword when one or more of the following symptoms occur:

- An abnormal end message is displayed at the system operator’s console indicating that the NPM main task (FNMMAIN) or one of its subtasks abnormally ended. The message, which contains the abend code, is found in both the system console log and the SYSPRINT data set.

- Message FNM325I (program name ENTERED ESTAE, CODE=abend code) is issued because the program abnormally ended and its extended specify task abnormal exit (ESTAE) routine was entered. This message identifies the program that failed and the abend code. Use each as part of your keyword string. For example, if you received the following message:

  FNM325I FNMONL00 ENTERED ESTAE, CODE=50C4

  you would assign the keyword string ABEND0C4 FNMONL00.

  Note: Only the last three characters of the code sequence are used. The S shown in this example message is not used in the keyword string.

Select one of the following keywords based on the type of abend:

- If the abend is a system abend, the keyword is ABENDxxx, where xxx is the abend code in hexadecimal, such as ABEND0C4, ABEND604, or ABEND13E. A system abend occurs when a problem is detected by the system. For example, an attempt to access protected storage would result in a system abend.

- If the abend is an NPM user abend, the keyword is ABENDUxxxx, where xxxx is the abend code in decimal. The NPM user abend might be an expected event because of circumstances in the system. A user abend occurs when NPM detects a problem that stops NPM from continuing to process normally. For example, when NPM runs out of buffer segments, it can no longer continue processing, and a user abend occurs.

For more information about abends, refer to the NetView Performance Monitor Messages and Codes book.

Some abend failures can be caused by incorrect JCL or definition statements, such as a reference to an incorrect library. System abend failures can result from an action such as issuing a system supervisor call (SVC) in a program with an incorrect event control block (ECB) address. Program check abend problems are hardware-detected error conditions. For example, an 0C4 or 0C1 abend generated by either a branch or a store to an address that is not valid. An abend problem can also result from a VSAM or VTAM error. If this occurs, check the allocation of VSAM or VTAM parameters. NPM user abend codes originate in NPM and are documented in the NetView Performance Monitor Messages and Codes book.

If you do not have information about how to recover from or prevent the abend, continue gathering documentation.

To document an ABEND problem, see “ABEND Problems” on page 38. Also, if this is the first time you have documented a problem, see “Beginning to Document Problems” on page 35.

Documentation Problem Symptoms

Choose the DOC keyword when one or more of the following symptoms occur:
There is incomplete or inaccurate information in the publications about the installation, operation, customization, messages, or diagnosis of NPM.

The documentation or online help panel is inconsistent in describing the use of an NPM function.

Select one of the following book order numbers for the NPM book in which there is a problem:

<table>
<thead>
<tr>
<th>Order Number</th>
<th>Book Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>GH19-6961-06</td>
<td>NetView Performance Monitor Concepts and Planning</td>
</tr>
<tr>
<td>SH19-6962-06</td>
<td>NetView Performance Monitor User’s Guide</td>
</tr>
<tr>
<td>SH19-6964-06</td>
<td>NetView Performance Monitor Installation and Customization</td>
</tr>
<tr>
<td>LY19-6381-06</td>
<td>NetView Performance Monitor Diagnosis</td>
</tr>
<tr>
<td>SH19-6966-06</td>
<td>NetView Performance Monitor Messages and Codes</td>
</tr>
<tr>
<td>SH19-6965-06</td>
<td>NetView Performance Monitor Reference</td>
</tr>
</tbody>
</table>

When you use the DOC keyword, you must include the last two numbers of the book order number. The format for the DOC keyword string is:

```
xxxxxxx DOC yyyyyyyyyy
```

where:

- `xxxxxxx` Specifies the program number.
- `yyyyyyyy` Specifies the book order number.

For example, if a problem exists in *NetView Performance Monitor Diagnosis*, whose order number is LY19-6381-06, you would use the following keyword string:

```
5655043 DOC LY19638106
```

Use the panel ID as a keyword for problems with online help panels. Report all problems with online help panels to the Support Center. Report a publication problem only if it affects the operation or use of NPM. If the problem does not affect the operation or use of NPM, send information about the problem as detailed in “Providing Feedback about Publications” on page xvii. You can report the following types of comments:

- Suggestions about the format or content of the publication
- General comments about the publication

To document a DOC problem, see “DOC Problems” on page 41. If this is the first time you have documented a problem, also see “Beginning to Document Problems” on page 35.

### Incorrect Output Symptoms

Choose the INCORROUT keyword when one or more of the following symptoms occur:

- The output appears incorrect or incomplete
- The output is formatted incorrectly
- You receive unexpected output, such as a garbled message on a panel, and the program does not appear to be in a loop
- You receive an NPM logic error.
Choosing Keywords

If the incorrect output involves a message other than FNM799S, see "Message Symptoms" for a description of the problem. If the incorrect output is on a panel, use the PANELID command to display the panel name and use that name as part of your keyword string. The format for the INCORROUT keyword string is:

```
xxxxxxx INCORROUT yyyyyyyyy
```

where:

- `xxxxxxx` Specifies the MVS program number.
- `yyyyyyyy` Specifies the panel ID.

For example, if the panel name is FNM0C204 and you are running NPM under MVS, use the following keyword string:

```
5655043 INCORROUT FNM0C204
```

If message FNM799S is issued, an internal NPM logic error has occurred. For detailed information about error message FNM799S, refer to the NetView Performance Monitor Messages and Codes book.

To document an INCORROUT problem, see "INCORROUT Problems" on page 42.

If this is the first time you have documented a problem, also see "Beginning to Document Problems" on page 35.

Loop Symptoms

Choose the LOOP keyword when one or more of the following symptoms occur:

- A part of the program is repeating itself
- A command has not completed after an extended period of time and processor utilization is very high
- There is high processor utilization or console (operator terminal) lockout

A loop can be either enabled or disabled. The symptom of an enabled loop is high processor utilization by a specific task. An enabled loop still allows system commands to be executed, and responses are returned to the console.

The symptoms of a disabled loop are similar to those of an enabled loop; however, system commands are not accepted. The system cannot be interrupted from an operator’s console.

To document a LOOP problem, see "LOOP Problems" on page 43. If this is the first time you have documented a problem, also see "Beginning to Document Problems" on page 35.

Message Symptoms

Choose the MSG keyword when one or more of the following symptoms occur:

- The message received is not normally expected.
- The message is issued with an incorrect format (misspelled words or unprintable characters in the message).
- The message does not appear as it is documented in the NetView Performance Monitor Messages and Codes book.
- A message appears to contain incorrect data.
A message was issued under conditions that should not have caused it to be issued.

The message is missing data.

A program problem can cause an NPM message to be issued at the system console, in the FNMILOG data set, in the SYSPRINT data set, or at an NPM operator’s terminal. NPM messages appear in one of the following formats:

- \text{FNM}.xxxz
- \text{FNM}yyyyz
- \text{FNM}1.xxxz

where:

- **FNM** Is an identifier for NPM. If the message associated with your problem does not have an FNM prefix, the problem is probably not with NPM.
- **FNM1** Is an identifier for the NPM batch reports function:
  - FNMI001 through FNMI099 are NPM batch report messages
  - FNMI100 through FNMI149 are general NPM messages
  - FNMI150 through FNMI169 are NPM NetWare Agent messages
  - FNMI170 through FNMI9999 are general NPM messages

- **nnn** Is the message number.
- **xxx** Is the three-digit message number.
- **yyyy** Is the four-digit message number.
- **z** Is the type of message, indicated by one of the following letters:
  - I An informative message
  - A An action needs to be taken
  - E An error condition
  - W A warning condition
  - S A severe error
  - T A termination error

Use the message as a keyword in your keyword string. For example, use MSGFNMI018 as a keyword if you receive the following message:

\text{FNM1018T OUT OF VECTOR VALUE STORAGE}

You do not need to include a leading zero for three-digit NPM messages. For example, use MSGFNME as a keyword if you receive the following message:

\text{FNM008E NCPNCP01 NOT DEFINED}

To document a MSG problem, see “MESSAGE Problems” on page 44. If this is the first time you have documented a problem, also see “Beginning to Document Problems” on page 33.

### Performance Problem Symptoms

Choose the PERFM keyword when one or more of the following symptoms occur:

- NPM commands take an excessive amount of time to complete
- NPM performance characteristics are below expectations
Choosing Keywords

NPM’s performance can be affected as a result of applying a program temporary fix (PTF) or a service-level update. You should gather documentation for a PERFM problem if you determine that NPM is performing poorly.

To document a PERFM problem, see "PERFM Problems" on page 45. If this is the first time you have documented a problem, also see "Beginning to Document Problems" on page 35.

Wait Symptoms

Choose the WAIT keyword when one or more of the following symptoms occur:

- The operator cannot enter commands or communicate with NPM, and NPM does not appear to be in a loop.
- There is no response to NPM at the system console.

A program problem can cause processing to stop for NPM with no abnormal completion codes and no excess processor utilization reported. For example, you should gather documentation for a WAIT problem if you enter an NPM command and do not receive a response although the processor and all other jobs are starting and ending normally.

To document a WAIT problem, see "WAIT Problems" on page 46. If this is the first time you have documented a problem, also see "Beginning to Document Problems" on page 35.

Choosing Additional Keywords

In addition to a problem type keyword, the following search arguments can be added to your keyword string to further classify your problem.

<table>
<thead>
<tr>
<th>Search For...</th>
<th>When...</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAF</td>
<td>The failure occurred in the archive data management programs.</td>
</tr>
<tr>
<td>DR</td>
<td>The dynamic reconfiguration associated with boundary nodes is incorrect.</td>
</tr>
<tr>
<td>FIL</td>
<td>The file service task appears to be writing incorrect output.</td>
</tr>
<tr>
<td>LOP</td>
<td>The failure occurred while you were trying to format and print the network log.</td>
</tr>
<tr>
<td>NET</td>
<td>The network collection function is incorrect.</td>
</tr>
<tr>
<td>NGA</td>
<td>The problem occurred with network gateway accounting.</td>
</tr>
<tr>
<td>NSA</td>
<td>The problem occurred with network session accounting.</td>
</tr>
<tr>
<td>NSI</td>
<td>The failure concerns the NetView synergy interface.</td>
</tr>
<tr>
<td>ONL</td>
<td>The failure occurred while you were using the NPM online facility.</td>
</tr>
<tr>
<td>PA</td>
<td>The failure occurred while sending alerts to the NetView program.</td>
</tr>
<tr>
<td>RTM</td>
<td>The RTM collection function is incorrect.</td>
</tr>
<tr>
<td>SES</td>
<td>The session collection function is incorrect.</td>
</tr>
<tr>
<td>SMF</td>
<td>The failure occurred while you were recording or collecting SMF data.</td>
</tr>
</tbody>
</table>
After you classify your problem using a keyword or keyword string, you must describe and document the problem. For a checklist related to your problem, see "Diagnosing with Checklists" on page 9. Follow the steps described in the checklist to ensure that all procedures were performed correctly. If the problem persists, collect the necessary documentation and report the problem to the Support Center.

This chapter lists the information you need to collect before contacting the Support Center. Procedures are also provided to help you find a solution for your problem, or to determine as much as possible about the failure to describe it accurately to the Support Center. You can use the following functions to help you collect the recommended documentation for your problem:

<table>
<thead>
<tr>
<th>This function ...</th>
<th>Is described in ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online problem determination</td>
<td>&quot;Using the Online Problem Determination Facilities&quot; on page 89</td>
</tr>
<tr>
<td>Initialization statements</td>
<td>&quot;NPM DEBUG and TRAP Initialization Statements&quot; on page 109</td>
</tr>
<tr>
<td>Debugging options</td>
<td>&quot;DEBUG Statement&quot; on page 110</td>
</tr>
<tr>
<td>Traps</td>
<td>&quot;TRAP Statement&quot; on page 119</td>
</tr>
<tr>
<td>NPM internal traces</td>
<td>&quot;Using the NPM Internal Trace&quot; on page 127</td>
</tr>
</tbody>
</table>

Use the worksheets in "Problem Worksheets" on page 155 to record the data you collect about your problem.

If you know the specific function that is failing, you might need to collect more documentation than is listed in this chapter. See "Documenting Specific Function Problems" on page 47.

### Beginning to Document Problems

Each problem that you encounter requires basic documentation. Collect the following information for problems for which you cannot identify the problem type. Collect this information before you gather information for a specific type of problem. Then, report the problem under the terms and conditions of your service agreement.

### Information Needed for Most Problems

The following list summarizes the information that you should collect for most program problems:

- The NPM program number
- The maintenance level and release number for the operating system
A record of PTFs or APARs for your operating system
A record of PTFs or APARs for your level of NPM
The VTAM and NCP version and release numbers
A record of PTFs or APARs for your level of VTAM and NCP
The NPM FNMILOG and NPM SYSPRINT data sets
A console dump of NPM
A copy of the system console log
A description of the error condition
Any unique information about your problem or system
NetWare users
- The release number of your NetWare operating system and the release number of
  your NetWare for SAA or NetWare Management Agent for NetView software.
- A record of patches applied to your operating system and NetWare for SAA or
  NetWare Management Agent for NetView
- A copy of the server console messages
- A description of the error condition
- Any unique information about your problem or system

Collecting the Information

The following steps describe how to collect the necessary information:

1. Record the NPM program number, or component ID number, that indicates the
   operating system you are using. The component ID number should be the first keyword
   in a keyword string, preceding the problem type and other modifier keywords.

2. Record the operating system release number.

3. Record any PTFs or APARs that are applied to your operating system.
   Record the maintenance level (your Program Update Tape level) for the following
   components of your operating environment:
   - The operating system
   - VTAM
   - NPM
   - NCP

4. Record any PTF, APARs, or patches that are applied to your level of NPM.
   - For systems other than NetWare: List PTFs or AMASPZAPs by module to show
     all the PTFs that apply to NPM. Also obtain a System Modification Program (SMP)
     control data set (CDS) listing by module of the NPM’s function modification
     identifier (FMID).
   - For NetWare: Find out which patches have been applied by typing the following
     command on the NetView program console:
     PATCHES

5. Record the VTAM and NCP version and release numbers. Record any PTFs or APARs
   that apply to these programs.

6. Print the NPM FNMILOG and NPM SYSPRINT data sets. For information about these
   data sets, see "Using the FNMILOG and SYSPRINT Data Sets" on page 74.

7. Obtain a dump of NPM based on your operating system.
   Use the following DUMP command with options CSA, NUC, PSA, and RGN to obtain
   a console dump of NPM:
These options produce a dump of NPM’s address space and the common storage area (CSA). Do not use the DUMP option of the CANCEL command. The DUMP option produces a dump that is not sufficient for problem determination. For more information, see the *OS/390 MVS System Commands* book.

You can also create a dump using the NPM Online Problem Determination Facilities. For more information, see “Taking an NPM Storage Dump” on page 107.

8. Print a copy of the system console log. The system console log is a data set or file that stores job-related information, operational data, commands, descriptions of unusual occurrences, and messages to or from the operator. Use the console log to examine what the operator was doing when the error occurred.

9. *NetWare users*, make a copy of the server console messages.

10. Research the sequence leading to the problem, including the commands and panels entered before the problem occurred. Use the PANELID command to identify the panels you were using as the problem occurred. You can print these panels with the PRINT command. For information about these commands, see “Diagnosing with the PANELID Command” on page 87 and “Diagnosing with the PRINT Command” on page 87.

Try to answer the following questions as you research the problem:

- What was the first indication of the problem?
- What were you trying to do?
- What should have happened?
- What did happen?
- Has the function worked before?
- Can you recreate the problem?

11. Record any unique information about the problem or your system. For example, answer the following questions:

- Were other applications running when the problem occurred?
- Are the programs that are running modified in any way?
- Is the problem limited to only cross-domain or cross-network sessions?
- Are the installation members different from the sample members?
- Have you recently added new software or hardware to your system?

12. Continue collecting information according to the requirements of your specific problem type. See the appropriate sections in this chapter and in “Documenting Specific Function Problems” on page 47.

13. Report your problem to the Support Center. If requested by IBM, you might need to submit the collected documentation to create an APAR.

To put the required files on tape, use the VSAM IDCAMS utility for VSAM files. Use the IEBGENER utility for all other files.

For more information about reporting problems to the Support Center and creating an APAR, see “NPM Diagnosis Overview” on page 5.
ABEND Problems

An ABEND problem can result from:

- Incorrect JCL or definition statements
- A program with an incorrect event control block (ECB) address
- A VSAM or VTAM error.

For information about classifying ABEND problems, see "Abnormal End Symptoms" on page 30.

For more information about diagnosing ABEND problems, see "Using PDI, FNMILOG, SYSPRINT, and PANELID" on page 73.

If you have determined that you have an ABEND problem, first ensure that you have enough storage. (You might not have allocated enough storage if this is the first time you have initialized NPM.) Use the formulae in the NetView Performance Monitor Installation and Customization book to ensure the allocation is adequate.

Information Needed

If you have allocated enough storage, collect the following information. Use the worksheets in "Problem Worksheets" on page 153 to record your information.

- Basic documentation
- Abend code
- Program status word (PSW) at the time of the abend
- Contents of the general registers at the time of the abend
- Name of the module containing the failing instruction
- The SYSMOD level of the failing module
- Offset into the module of the instruction pointed to by the PSW

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. This information is listed in "Beginning to Document Problems" on page 35.

   If a dump occurs with the abend, save the unformatted dump data set on a tape or cartridge for the Support Center. If multiple abends occur, as is typical in out-of-storage errors, you need to save only the dump from the first abend.

   Look for a message that shows only a partial dump was taken in the system console log. The Support Center usually requires a complete dump to solve abend problems. Therefore, you might need to increase the size of your dump data set and recreate your problem to get a complete dump.

   You can view or print the dump data set without altering it by using an interactive dump viewing utility, such as the Interactive Problem Control System (IPCS). The IPCS is a component of OS/390 MVS operating systems that permits online problem management, interactive problem diagnosis, online debugging for disk-resident control program abend dumps, problem tracking, and problem reporting.

   You can also use a formatting utility on the unformatted dump to create a formatted file and print the dump. However, because this formatted file contains printer control characters, the Support Center cannot use it. You must keep a copy of the unformatted dump.
2. Locate and record the abend code.
If message FNM325I was issued at the time of the abend, see the console log for the exact message text. This message contains the abend code and the name of the load module. If the abend code is a user-abend code (Uxxxx), see the *NetView Performance Monitor Messages and Codes* book for the cause of the abend.

**Note:** Use the abend code as part of your keyword string when searching the RETAIN database for solutions to your problem. For example, if the abend code is 0C1, a keyword string might be 566533301 ABEND0C1. For more information about keyword strings, see "Classifying Program Problems" on page 29.

3. Locate the PSW in the NPM dump. The PSW is the pointer to the instruction that would be processed next if the abend did not occur.
Using a dump formatting utility, find the work area labeled RTM2WA Summary. This summary contains the completion code, the registers at the time of the error, and the PSW.

4. Locate the failing instruction using the PSW.
The second word in the extended control (EC) PSW points to the instruction that would be processed next if the failure did not occur. Find this location in the dump. For example, in the RTM2WA summary in Figure 2, the PSW is X'076C2000 80085A36'. Therefore, you can locate the failing instruction in the dump just before the address X'00085A36'. In the example shown in Figure 3, the instruction pointed to by the PSW is X'0000C9B0'.

5. Locate the failing module.
To discover which module was running when the abend occurred, scan backward in the right margin from the PSW to the module name. If the module begins with FNM, it is an...
NPM module. If the module name does not begin with FNM, the problem is probably not with the NPM program. In the example dump shown in Figure 3 on page 39, the failing module is FNMCMSS0.

You can also find the name of the failing module or CSECT by looking in Register 12 or at offset X'28' of the PDI. To find the PDI control block, see “Using PDI, FNMILOG, SYSPRINT, and PANELID” on page 73.

After you find the module name, locate the coded branch at the beginning of the module just before the module name and date compilation. Use this to determine the starting location and instruction of the failing module. For example, in Figure 3 on page 39, module FNMCMSS0 begins at location X'000859B0' with instruction X'47F0F050'.

Use the failing module name as part of your keyword string when searching the RETAIN database for solutions to your problem. For example, if the failing module is FNMCMSS0, a keyword string could be 56653301 ABEND0C1 FNMCMSS0 REGF. For more information about keyword strings, see “Classifying Program Problems” on page 29.

6. Record the SYSMOD level of the module. If maintenance was applied to the module containing the failing instruction, the latest SYSMOD level appears just after the module name and date compilation. For example, module FNMCMSS0 shown in Figure 3 on page 39 was compiled on February 28, 1992 at 4:38 p.m. and is at PTF level UY40329. If you found the module name using the PDI control block, you can find the SYSMOD level at offset X'38' in the control block.

7. Calculate the hexadecimal offset into the failing module.

Subtract the contents of Register C from the second word of the PSW to obtain the hexadecimal offset. For example, if the PSW address is X'00085A36' and the module address is X'000859B0', the offset is X'00000086'.

```
00085A36 [PSW address]
- 000859B0 [address of module]
--------
00000086 [offset]
```

8. Report the problem to the Support Center after collecting all of the required information. For more information about reporting the problem, see “Reporting Problems to the Support Center” on page 7.
DOC Problems

A DOC problem can result from an error or omission in the system documentation.

Information Needed

Collect the following information to document a DOC problem:

- The order number and revision level of the book or the identification and panel print of the online help panel
- The location of the error in the book or panel
- A description of how the documentation caused the problem

Collecting the Information

The following steps describe how to collect the necessary information.

1. Identify the order number and revision level of the book or the name of the online help facility panel involved. The format for book numbers that appear on the front covers of the NPM books is:
   xxxx-xxxx-yy

   where:

   xxxx-xxxx  Specifies the book order number
   yy         Specifies the two-digit revision number.

   The panel name is displayed in the upper left corner of the screen and begins with FNM. See “Diagnosing with the PANELID Command” on page 87 for more information about panel names. Also, obtain a copy of the panel using the PRINT command. See “Diagnosing with the PRINT Command” on page 87 for information about the PRINT command.

2. Locate the pages in the document or the panels that contain incorrect or incomplete information and prepare a description of the problem.

3. Report the problem to the Support Center after collecting all of the required information. See “Reporting Problems to the Support Center” on page 7 for more information about reporting the problem.

Note: Report a documentation problem to the Support Center only if the problem affects the operation of NPM, or if the problem involves online help panels. Otherwise, send information about the problem as detailed in “Providing Feedback about Publications” on page xvii.
INCORROUT Problems

An INCORROUT problem can result from output that is:
- Incorrect or incomplete,
- Formatted incorrectly
- Unexpected
- An NPM logic error.

Information Needed

To document an INCORROUT problem, collect the following information:
- Basic documentation
- A description of the incorrect output and the expected output
- The specific output that is incorrect

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. This information is listed in “Beginning to Document Problems” on page 35.
2. Record the type of output that is incorrect (for example, the panel, message, or data record). Explain how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
   Include the options chosen in the panel, if appropriate.
3. Use the PANELID command to identify the panel in use at the time of the incorrect output. Record the panel ID and generate a copy of the panel using the PRINT command. For more information, see “Diagnosing with the PANELID Command” on page 87 and “Diagnosing with the PRINT Command” on page 87.
4. In the case of a logic error message, note the error code, module name, and offset. For more information about why the error was generated, see the NetView Performance Monitor Messages and Codes book.
5. After collecting all the required information, report the problem to the Support Center. For more information about reporting the problem, see “Reporting Problems to the Support Center” on page 7.
LOOP Problems

A LOOP problem can occur when:

- A part of the program repeats itself
- A command has not completed after an extended period of time and processor utilization is very high
- There is high processor utilization or console (operator terminal) lockout

Information Needed

Collect the following information to document a LOOP problem:

- Basic documentation
- Messages associated with the loop (if any)
- A TRAP dump on the message (if any)

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. This information is listed in “Beginning to Document Problems” on page 33. The console log shows the display PSW commands that indicate the address range of the loop.

2. Document any messages that might be displayed on the terminal at the time of the loop. Also, record any messages issued to the system or NPM console.

3. Obtain a TRAP dump on the message, if one was issued. For information about TRAP statements, see “NPM DEBUG and TRAP Initialization Statements” on page 109. Ensure that the FNMSDUMP DD statement points to a dump data set, such as the SYSMDUMP DD statement.

4. Report the problem to the Support Center after collecting all of the required information. For more information about reporting the problem, see “Reporting Problems to the Support Center” on page 3.
MESSAGE Problems

A MESSAGE problem can occur when, for example, you receive an unexpected message, or when the message is undocumented or incomplete.

Information Needed

Collect the following information to document a MESSAGE problem:
- Basic documentation
- A description of the expected and incorrect output
- A copy of the message showing the issuing module
- A copy of a TRAP dump on the message
- A copy of the specific output that is incorrect

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. This information is listed in "Beginning to Document Problems" on page 35.

2. Record the type of output that is incorrect (for example, the panel, message, or data record). Explain how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
   Include the options chosen in the panel, if appropriate.

3. Code a DEBUG statement with MSGMOD=YES. This option adds the name of the issuing module to the message text. For more information about DEBUG statements, see "DEBUG Statement" on page 110 for more information about DEBUG statements.

4. Obtain a TRAP dump on the message. For information about TRAP statements, see "NPM DEBUG and TRAP Initialization Statements" on page 109.
   Ensure that the FNMSDUMP DD statement points to a dump data set, such as the SYSMDUMP DD statement.

5. Use the PANELID command to record the panel in use at the time of the incorrect output. Copy these panels using the PRINT command. For further instructions, see "Diagnosing with the PANELID Command" on page 87 and "Diagnosing with the PRINT Command" on page 87.

6. Report the problem to the Support Center after collecting all of the required information. For more information about reporting the problem, see "Reporting Problems to the Support Center" on page 7.
PERFM Problems

A PERFM problem can occur when:
- NPM commands take an excessive amount of time to complete
- NPM performance characteristics are below expectations

Information Needed

Collect the following information to document a PERFM problem:
- Basic documentation
- System or network modifications
- A description of the operation attempted, the results expected, and the results received

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. This information is listed in "Beginning to Document Problems" on page 35.

2. Record any modifications to your system or your network. For example, do you have installation exits, command lists, and command processors processing under NPM and does the performance degradation relate to any user-installed code? Performance problems can be related to various system and networking constraints. Your IBM marketing representative can help you identify possible causes of a performance problem.

3. Record the actual performance, the expected performance, and the source of information for the expected performance. If a document is the source, record the order number and page number of the document.

4. Report the problem to the Support Center after collecting all of the required information. For more information about reporting the problem, see "Reporting Problems to the Support Center" on page 7.
WAIT Problems

A WAIT problem can occur when:

- The operator cannot enter commands or communicate with NPM, and NPM does not appear to be in a loop.
- There is no response to NPM at the system console.

Information Needed

Collect the following information to document a WAIT problem:

- Basic documentation
- The activities of the task leading up to the wait

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. This information is listed in "Beginning to Document Problems" on page 35.

2. Use FNMILOG and SYSPRINT to research the activity before the wait and to identify which operation is in the wait state.

3. Report the problem to the Support Center after collecting all of the required information. For more information about reporting the problem, see "Reporting Problems to the Support Center" on page 7.
This chapter describes how to document specific function problems that might not be classified with keywords. If your problem does not fit one of the keywords described in "Documenting Problems" on page 35, follow the instructions for documenting the specific function:

- Archive data management
- Batch reporting utility (FNMREPRT)
- Dynamic definite response (DDR)
- LAN bridge collection
- LAN segment utilization data collection
- NCP definition
- NetWare resources collection
- Network collection
- NetView synergy interface (NSI)
- ODLC LAN resource data collection
- Online system
- Performance alerts
- Response time monitor (RTM) collection
- Session collection
- System console support
- System management facility (SMF) collection
- Transit analysis program (FNMTAP)
- VTAM statistics collection

**Note:** Before collecting information, see "Diagnosing with Checklists" on page 9 for a checklist related to your problem. Follow the steps described in the checklist to ensure that all procedures were performed correctly. Collect the information listed in this chapter if the problem persists.

This chapter lists the information you need to describe program problems, document failures, and report unresolved problems. This chapter also describes the procedures to either find a solution to your problem or to determine as much as you can about the failure and accurately describe it to the Support Center. You can use the following functions to help you collect the recommended documentation for your problem:

<table>
<thead>
<tr>
<th>This function ...</th>
<th>Is described in ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online problem determination</td>
<td>&quot;Using the Online Problem Determination Facilities&quot; on page 89</td>
</tr>
<tr>
<td>Initialization statements</td>
<td>&quot;NPM DEBUG and TRAP Initialization Statements&quot; on page 109</td>
</tr>
<tr>
<td>Debugging options</td>
<td>&quot;DEBUG Statement&quot; on page 110</td>
</tr>
<tr>
<td>Traps</td>
<td>&quot;TRAP Statement&quot; on page 119</td>
</tr>
<tr>
<td>NPM internal traces</td>
<td>&quot;Using the NPM Internal Trace&quot; on page 127</td>
</tr>
</tbody>
</table>

For a summary of the information you should collect for most program problems, see "Beginning to Document Problems" on page 35.
Archive Data Management Problems

Information Needed

Collect the following information to document a problem with the archiving procedure:

- Basic documentation
- A SYSABEND dump (if an abend occurred)
- The normal output listing and the failing output listing from the run
- A listing of the FNMARCHV JCL control statements
- A VSAM utility listing of both the input and output files

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.
2. If an abend occurred, use the SYSABEND DD statement to obtain a dump (see “ABEND Problems” on page 38). Record the type of output that is incorrect, for example, the field in the output record.
3. Explain how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
4. Verify that the control statements in the FNMARCHV JCL are set up correctly.
5. Use the VSAM REPRO command to create a copy of the input and output files.
6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
Batch Reporting Utility (FNMREPRT) Problems

Information Needed

Collect the following information to document a problem with the batch reporting facility (FNMREPRT):

- Basic documentation
- A description of the incorrect and expected output
- A copy of the following data sets:
  - INPUT
  - OUTPUT
  - LOG
  - SYSIN
  - Any other user-modified or user-supplied data sets (LISTS, REPORTS, SYNONYMS, and so on)
- A copy of any installation-wide exits used
- A copy of the calling procedure or EXEC

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems” on page 33.

2. Explain how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is there more output than expected?
   - Is the information inaccurate?
   - Is the information formatted incorrectly?

3. Make a copy of the following data sets:
   - INPUT
   - OUTPUT
   - LOG
   - SYSIN
   - Any other user-modified or user-supplied data set, such as LISTS, REPORTS, or SYNONYMS

   Record any messages found in the LOG data set.

4. If you are using installation-wide exits, submit copies of the exits to the Support Center.

5. Make a copy of the procedure or EXEC used to call FNMREPRT.

6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center” on page 7.
Dynamic Definite Response (DDR) Problems

Information Needed

Collect the following information to document a DDR problem:

- Basic documentation
- Data from a session with the LU experiencing the problem
- A copy of the FNMVLOGx data sets that contain the trace data of the LU session passing data to NPM
- Copies of panel FNM03SMN and the panel that was displayed when the problem occurred
- The number of entries added to and deleted from PMT

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.

2. Using the NPM session collection function, start collecting data from the LU experiencing the problem, and begin a trace on this session. Use the following parameters to start the collection:
   SESSCOLL RESOURCE=lu_name, VLOG=1, MIN=N0, SESSH=R

3. Ensure that the FNMVLOGx data sets are empty, then re-create the problem. Submit the unformatted and machine-readable FNMVLOGx data set to the Support Center.

4. Use the PANELID command to identify the panel in use at the time of the incorrect output. Copy these panels using the PRINT command. In addition, copy the following panel:

   Fast Path | Panel ID | Panel Name
   =2.6.4    | FNM03SMN | NPM Session Analysis Summary - Logical Unit

   For more information, see "Diagnosing with the PANELID Command" on page 87 and "Diagnosing with the PRINT Command" on page 87.

5. Using the NPM Online Problem Determination Facilities, enter:
   =P.1;DUMP;CPMT

   Record the following values:

   Offset | Description
   X'78'  | Overflow counter, indicating whether DDR was turned on
   X'7C'  | Number of entries added to the table
   X'80'  | Number of entries deleted from the table

6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
LAN Bridge Collection Problems

Information Needed

Collect the following information to document a LAN bridge collection problem:

- Basic documentation
- A copy of the procedure used to start NPM
- A copy of the procedure to start NetView (the network log)
- A description of the incorrect and expected output. This information includes the name and a copy of the panel, message, or data record that is incorrect.
- A dump of the CSA address space, subsystem interface address space, and the NetView address space

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.

2. Make a copy of the procedure used to start NPM. This is the startup JCL.

3. Make a copy of the network log from NetView.

4. Record the type of output that is incorrect, for example, the panel, message, or data record. Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

5. Obtain a dump of the CSA address space, subsystem interface address space, and the NetView program address space. For more information, see “Beginning to Document Problems” on page 35.

6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
LAN Segment Utilization Data Collection Problems

Information Needed
Collect the following information to document a LAN segment utilization data collection problem:
- Basic documentation
- A copy of the procedure used to start NPM
- A copy of the procedure to start NetView (the network log)
- A description of the incorrect and expected output
- Copies of the SMF log files, VSAM files, or the NPM sequential log file
- A trace on the data transfers across the program-to-program interface
- A dump of the CSA address space, subsystem interface address space, and the NetView address space

Collecting the Information
The following steps describe how to collect the necessary information:
1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.
2. Make a copy of the procedure used to start NPM. This is the startup JCL.
3. Make a copy of the network log from NetView.
4. Record the type of output that is incorrect (for example, the panel, message, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
5. Obtain copies of the SMF log files, review files, or the NPM sequential log file for submission to the Support Center.
6. Obtain a copy of a trace on the data transfers that cross the program-to-program interface. Ensure that LAN=YES is coded on the DEFAULT statement in the FNMPDET member of the FNMPARM partitioned data set.
7. Obtain a dump of the CSA address space, subsystem interface address space, and the NetView address space.
8. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
NCP Definition Problems

Information Needed

Collect the following information if a problem occurs while defining NCP to NPM, or if NCP resources are missing or defined incorrectly:

- Basic documentation
- A listing of the procedure used to start NPM
- A copy of the NCP resource resolution table (RRT)
- A copy of the NCP VTAMLST file

Collecting the Information

The following steps describe how to collect the necessary information.

1. **Collect the basic information required for most problems.** See "Beginning to Document Problems" on page 35.

2. Use the following procedure if NCP resources are missing or defined incorrectly:
   a. Code an NCP command for the NCP error in the FNMSTRT member or file.
   b. Code CONFIG=YES, LIST=YES on the NPM initialization statement.
   c. Restart NPM.

   These steps cause the NCP and its resources to be included in the configuration report written to FNMILOG. For more information, see "Using the FNMILOG and SYSPRINT Data Sets" on page 74.

3. Make a copy of the procedure used to start NPM. This is the startup JCL.

4. Verify that the initialization statements and the procedures used to start NPM were correct.

5. Copy the NCP RRT to tape.

6. Copy the NCP VTAMLST file to tape.

   If you used OPTION NEWDEFN=YES when generating the NCP, copy the NCP source file and the NCP NEWDEFN file.

7. Report the problem to the Support Center after collecting the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
NetWare Resources Collection

NetWare Resources Collection Problems

Information Needed

Collect the following information to document a NetWare resources collection problem:

- Basic documentation
- A copy of the procedure used to start NPM
- A copy of the procedure to start NetView (the network log)
- A description of the incorrect and expected output. This information includes the name and a copy of the panel, message, or data record that is incorrect.
- A dump of the CSA address space, subsystem interface address space, and the NetView address space
- A copy of the server AUTOEXEC.NCF file
- A copy of the server STARTUP.NCF file
- A copy of the file server system log
- A NetWare for SAA trace obtained locally using the NetWare facilities or the generalized trace facility (GTF) trace data set showing the VTAM buffer trace.

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.
2. Make a copy of the procedure used to start NPM. This is the startup JCL.
3. Make a copy of the network log from NetView.
4. Record the type of output that is incorrect (for example, the panel, message, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
5. Obtain copies of the SMF log files, review files, or the NPM sequential log file for submission to the Support Center.
6. Obtain a copy of a trace on the data transfers that cross the program-to-program interface. Ensure that LAN=YES is coded on the DEFAULT statement in the FNMPDET member of the FNMPARM partitioned data set.
7. Obtain a dump of the CSA address space, subsystem interface address space, and the NetView program address space. For more information, see "Beginning to Document Problems" on page 35.
8. Obtain a copy of the server AUTOEXEC.NCF file from the NetWare file server’s SYSTEM directory.
9. Obtain a copy of the server STARTUP.NCF file from the directory containing your NetWare file server’s software.
10. Obtain a copy of the NetWare for SAA trace produced locally using the NetWare facilities or a GTF trace.
    To obtain a copy of the VTAM buffer trace through GTF:
    a. Clear the GTF trace data set.
    b. Start GTF.
c. Start the VTAM buffer trace specifying the options TYPE=BUF and ID=sp_name
   where sp_name is the service point (SP) physical unit (PU) name of the server
   where the problem occurred.
   If you do not know the SP™ PU name, then specify the option ID=VTAM to trace
   all the inbound and outbound buffers for the service points from which data is being
   collected.

11. Record the version and release of the NetWare server you are using.

12. Record the version and release of the NetWare for SAA software you are using.

13. Report the problem to the Support Center after collecting all the required information.
    For more information about reporting a problem, see "Reporting Problems to the
    Support Center" on page 7.
Network Collection Problems

Information Needed
Collect the following information to document an NPM network collection problem:
- Basic documentation
- A listing of the procedure used to start NPM
- Data from a session between the NPALU and the NCP
- A trace on the NPALU
- A copy of the FNMVLOGx data sets that contain the trace data of the NPALU session that is passing the network data to NPM

Collecting the Information
The following steps describe how to collect the necessary information:
1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.
2. Make a copy of the procedure used to start NPM. This is the startup JCL.
3. Verify the procedure used to start NPM to ensure that the FNMLLIB DD statement points to the correct library containing the NCP RRT.
4. Start collecting data from the NPALU session to the NCP and begin a trace on this session using the NPM session collection function. This must be the first command in the FNMSTRT member. Use the following parameters to start the collection:
   SESSCOLL RESOURCE=npalu_name, VLOG=1, MIN=N0, SESSH=R
5. Ensure that the FNMVLOGx data sets are empty and then recreate the problem. Submit the unformatted and machine-readable FNMVLOGx data set to the Support Center.
6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
NetView Synergy Interface (NSI) Problems

Information Needed
Collect the following information to document an NSI problem:
- Basic documentation
- A description of the incorrect output and the expected output
- The name and a copy of the panel, message, or data record that is incorrect
- The generalized trace facility (GTF) trace data set showing the NSI trace
- A dump of the NSI address space

Collecting the Information
The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.

2. Record the type of output that is incorrect (for example, the panel, message, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

3. Use the PANELID command to identify the panel in use at the time of the incorrect output. Copy this panel using the PRINT command. For further instructions, see "Diagnosing with the PANELID Command" on page 87 and "Diagnosing with the PRINT Command" on page 87.

4. Obtain a copy of the NSI trace through GTF by taking the following steps:
   a. Clear the GTF trace data set.
   b. Start GTF from the OS/390 console, specifying the option TRACE=USRP and NPMNSI.TRACE as the GTF trace data set. GTF creates NPMNSI.TRACE as new. Its space allocation may need to be modified depending on the conditions. For example, issue the following command:

   \[ S \text{ GTF.GTF,DSN=NPMNSI.TRACE,SPACE=(CYL,(50,100)),DISP=(,CATLG),UNIT=SYSDA} \]

   GTF requests trace options:
   \[ 01 \text{ AHL100A SPECIFY TRACE OPTIONS} \]
   c. Issue the following command:

   \[ R \text{ 01,TRACE=USRP} \]

   The system responds with:
   \[ IEE6001 \text{ REPLY TO 01 IS; TRACE=USRP} \]
   \[ \text{TRACE=USR} \]
   \[ 02 \text{ AHL102A CONTINUE TRACE DEFINITION OR END} \]
   d. If you want only to trace NSI (EID=800), issue the following command:

   \[ R \text{ 02,USR=(800),END} \]

   - OR -
   e. If you also want to include VTAM Buffer Trace records (records X'FEF' and X'FF1'), issue the following command:

   \[ R \text{ 02,USR=(FEF,FF1,800),END} \]
The system responds with:
03 AHL125A RESPECIFY TRACE OPTIONS OR REPLY U

f. Issue the following command:
R 03,U

GTF completes the startup, generating the message:
AHL031I GTF INITIALIZATION COMPLETE

g. Start NSI. NSI has to be started before starting either NPM or the Session Manager.
h. When NSI is active, modify NSI to write to GTF by issuing the following command:
F nsi_proc_name,TRACE,START

The NSI responds with the following messages:
FNM932I MODIFY COMMAND ACCEPTED
FNM1275I START TRACE COMMAND COMPLETED

i. Start the Session Manager and NPM and reproduce the problem.
j. When the problem has been reproduced, stop the NSI trace:
F nsi_proc_name,TRACE,STOP

The NSI responds with the following messages:
FNM932I MODIFY COMMAND ACCEPTED
FNM1275I STOP TRACE COMMAND COMPLETED

k. Turn the GTF off:
P GTF,GTF

GTF responds with the following message:
AHL006I GTF ACKNOWLEDGES STOP COMMAND

5. Obtain a dump of the NSI address space.

Use the following system commands to dump the NSI address space:
DUMP COMM=(NSI,NPM)
R nn,JOBNAME=(NPMV2R7,NSI),SDATA=(NUC,RGN,CSA)

These commands produce a dump of the NSI address space and CSA. Do not use the dump option of the CANCEL command. This option produces a dump that is not sufficient for problem determination.

Keep a copy of the unformatted dump. The NPM specialist needs an unformatted copy for APAR documentation. For more information about the OS/390 MVS operating system, see the OS/390 MVS System Commands book.

6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see Reporting Problems to the Support Center on page 7.

7. To format the raw GTF file NPMNSI_TRACE, use IPCS by issuing the following command:
GTF USR(800)

This command formats the NSI trace in IPCS.
ODLC LAN Resource Data Collection Problems

Information Needed

Collect the following information to document an ODLC LAN resource data collection problem:

- Basic documentation
- A listing of the procedure used to start NPM
- A description of the incorrect output and the expected output
- The name and a copy of the panel, message, or data record that is incorrect
- A dump of the CSA address space
- A copy of the NCP definition generation member or file

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.
2. Make a copy of the procedure used to start NPM. This is the startup JCL.
3. Record the type of output that is incorrect (for example, the panel, message, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
4. Make a copy of the network log from NetView.
5. Obtain a dump of the CSA address space. For more information, see “Beginning to Document Problems” on page 35.
6. Obtain a copy of the NCP definition generation member or file. Ensure that NPA=(YES,DR) is coded in the BUILD macro of the NCP definition generation member or file.
   Also, verify that NPACOLL=(YES,EXTENDED) is coded in the GROUP macro. For more information about coding the NPACOLL parameter, see the NetView Performance Monitor Installation and Customization book.
7. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
Online System Problems

Information Needed

Collect the following information to document an NPM online system problem:

- Basic documentation
- A listing of the procedure used to start NPM
- The panel identification
- A copy of any NPM EXECs involved in the problem

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.

2. Verify that the initialization statements and the procedures used to start NPM were correct.

3. Use the PANELID command to identify the panel in use at the time of the incorrect output. Copy this panel using the PRINT command. For more information, see "Diagnosing with the PANELID Command" on page 87 and "Diagnosing with the PRINT Command" on page 87 for more information.

4. If one or more NPM EXECs were involved in the problem, send a copy of these EXECs along with the other information.

5. Report the problem to the Support Center after collecting the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
Performance Alert Problems

Information Needed
Collect the following information to document NPM performance alert problems:

- Basic documentation
- A listing of the procedure used to start NPM
- A copy of the FNMVLOGx data sets that contain the trace data of the NPALU session that is passing the network data to NPM

Collecting the Information
The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.
2. Verify that the initialization statements and the procedures used to start NPM are correct.
3. Start collecting data from the NPALU session to the NCP using the NPM session collection function. Use the following parameters:
   \[ \text{SESSCOLL} \text{ RESOURCE=} \text{npalu\_name}, \text{VLOG=1, MIN=NO, SESSH=V} \]
4. Ensure that the FNMVLOGx data sets are empty and then recreate the problem. Submit the unformatted and machine-readable FNMVLOGx data set to the Support Center.
5. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
RTM Collection Problems

Information Needed

Collect the following information to document RTM collection problems:

- Basic documentation
- A description of the incorrect and expected output
- A copy of the type 28 and type 39 records from the SMF data set on tape
- A copy of the assembled listing of FNMU83 and FNMU84 modules
- AMBLIST or a AMASPZAP dump of IEFU83 and IEFU84 load modules
- A copy of the link-edit JCL used to create the IEFU83 and IEFU84 load modules
- Copies of the FNMO2OPR, FNMO2OPM, FNMO2OPC, and FNMO2OSM panels

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.

2. Record the type of output that is incorrect, for example, the panel, message, or data record. Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

3. Copy the type 28 and 39 records from the SMF data set to tape.

4. For the IEFU83 and IEFU84 load modules, obtain a copy of the link-edit JCL used to create them and the AMBLIST or AMASPZAP dump. Also, copy the assembled listings of the FNMU83 and FNMU84 load modules.

5. Use the PANELID command to identify the panel in use at the time of the incorrect output. Copy these panels using the PRINT command. In addition, copy the following panels:

<table>
<thead>
<tr>
<th>Fast Path</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=9.4</td>
<td>FNMO2OPM</td>
<td>Performance Measurement Table Display</td>
</tr>
<tr>
<td></td>
<td>FNMO2OPR</td>
<td>PMT File Services Summary</td>
</tr>
<tr>
<td></td>
<td>FNMO2OPC</td>
<td>Session Collection Performance</td>
</tr>
<tr>
<td>=9.6</td>
<td>FNMO2OSM</td>
<td>SMF User Records Display and Management</td>
</tr>
</tbody>
</table>

   Note: You can access FNMO2OPM by using fast path =9.4. You then access FNMO2OPR by pressing PF8 (NEXT) from FNMO2OPM and then PF8 (NEXT) from FNMO2OPS. You access FNMO2OPC by pressing PF8 (NEXT) from FNMO2OPR and then PF8 (NEXT) from FNMO2OPV.

For more information, see "Diagnosing with the PANELID Command” on page 87 and "Diagnosing with the PRINT Command” on page 87.

6. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center” on page 7.
Session Collection Problems

Information Needed

Collect the following information to document session collection and analysis problems:

- Basic documentation
- A copy of a DEBUG statement with PRINTMBR=YES
- A description of both the incorrect and expected output
- A copy of the following panels:
  - FNM03SSA
  - FNM02OPM
  - FNM02OPR
- The NPM internal trace written to the GTF data set
- A session trace on one of the LUs experiencing the problem
- A VTAM log trace of the LU session written to the FNMVLOGx data sets
- A console dump of NPM when the session collection is active
- A copy of the NPM panel (with the panel identification) or a copy of the NPM data set showing the incorrect information

Collecting the Information

The following steps describe how to collect the necessary information:

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.

2. Code the NPM DEBUG statement with GTF=YES and PRINTMBR=YES. For more information, see "DEBUG Statement" on page 110.

3. Record the type of output that is incorrect (for example, the panel, message, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

4. Use the PANELID command to identify the panels in use at the time of the incorrect output. Copy these panels using the PRINT command. In addition, copy the following panels:

<table>
<thead>
<tr>
<th>Fast Path</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=2.2</td>
<td>FNM03SSA</td>
<td>Session Status/Stop (Active) Node Selection</td>
</tr>
<tr>
<td>=9.4</td>
<td>FNM02OPM</td>
<td>Performance Measurement Table Display</td>
</tr>
<tr>
<td></td>
<td>FNM02OPR</td>
<td>PMT File Services Summary</td>
</tr>
</tbody>
</table>

   Note: You can access FNM02OPM by using fast path =9.4. You then access FNM02OPR by pressing PF8 (NEXT) from FNM02OPM and then PF8 (NEXT) from FNM02OPS.

   For more information, see "Diagnosing with the PANELID Command" on page 87 and "Diagnosing with the PRINT Command" on page 87.

5. If the transit time values are incorrect, collect trace information on one or more representative logical units using the session collection command or the Start Session panel (FNM02SCL), fast path =2.1. Use the following parameters:
SESCOLL RESOURCE=luname,VLOG=1,MIN=NO

Note: VLOG=1 collects path information units (PIUs) to the unformatted and machine-readable FNMVLOGx data sets. A copy of the FNMVLOGx data set is required by the Support Center.

Use the NPM Online Problem Determination Facilities (fast path =PD) to trace the session collection processing for a specific LU. For more information, see “Displaying Session Traces” on page 101.

6. Clear and reset the FNMVLOGx data sets. Start a VTAM log trace of the LU session to the FNMVLOGx data sets by adding VLOG=1 to the SESSCOLL command in the FNMSTRT data set or by typing 1 on panel FNM02SCL. For more information, see the NetView Performance Monitor Installation and Customization book and the NetView Performance Monitor User’s Guide.

7. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
System Console Support Problems

Information Needed

Collect the following information to document system console support problems:
- Basic documentation
- A description of both the incorrect and expected output
- A TRAP dump of the message issued (if any)

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.
2. Record the type of output that is incorrect (for example, the panel, messages, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
3. Obtain a TRAP dump on the message, if one was issued. For information about TRAP statement, see “NPM DEBUG and TRAP Initialization Statements” on page 109.
   Ensure that the FNMSDUMP DD statement points to a dump data set, such as the SYSMDUMP DD statement.
4. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
SMF Collection Problems

Information Needed

Collect the following information to document SMF collection problems:

- Basic documentation
- A description of the incorrect output
- The panel identification
- Copies of the following panels:
  - FNM02OPR
  - FNM02OSM
- A copy of the SMFPRMxx member in SYS1.PARMLIB that is active
- A copy on tape of the type 28 records from the SMF data set
- A copy of the assembled listing of FNMU83 and FNMU84 modules
- AMBLIST or a AMASPZAP dump of IEFU83 and IEFU84 load modules
- A copy of the link-edit JCL used to create the IEFU83 and IEFU84 load modules

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.

2. Record the type of output that is incorrect for example, the panel, messages, or data record. Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

3. Use the PANELID command to identify the panel in use at the time of the incorrect output. Copy these panels using the PRINT command. In addition, copy the following panels:

<table>
<thead>
<tr>
<th>Fast Path</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=9.6</td>
<td>FNM02OPR</td>
<td>PMT File Services Summary</td>
</tr>
<tr>
<td>=9.6</td>
<td>FNM02OSM</td>
<td>SMF User Records Display and Management.</td>
</tr>
</tbody>
</table>

   **Note:** You can access FNM02OPR by pressing PF8 (NEXT) from the Performance Measurement Table Display (FNM02OPM), fast path =9.4, and then PF8 (NEXT) from FNM02OPS.

   For more information, see “Diagnosing with the PANELID Command” on page 87 and “Diagnosing with the PRINT Command” on page 87.

4. Obtain a copy of the SMFPRMxx member in the SYS1.PARMLIB that is active.

5. Copy the type 28 records from the SMF data set to tape.

6. For the IEFU83 and IEFU84 load modules, obtain a copy of the link-edit JCL used to create them and the AMBLIST or AMASPZAP dump. Also, copy the assembled listings of the FNMU83 and FNMU84 load modules.
7. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
Transit Analysis Program (TAP) Problems

Information Needed

Collect the following information to document transit analysis program (FNMTAP) problems:
- Basic documentation
- A description of both the incorrect and expected output
- A copy of the JCL or EXEC and the control statements used to execute FNMTAP00
- A copy of the input file to FNMTAP00

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. See "Beginning to Document Problems" on page 35.
2. Record the type of output that is incorrect (for example, the panel, messages, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?
3. List the FNMTAP JCL or EXEC and the control statements for submission to the Support Center.
4. Copy the input file to tape for submission to the Support Center.
5. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see "Reporting Problems to the Support Center" on page 7.
VTAM Statistics Collection Problems

Information Needed

Collect the following information to document VTAM statistics collection problems:

- Basic documentation
- A description of the incorrect output and the expected output
- The level of VTAM being monitored
- A description of the VTAM configuration related to the incorrect output or abend
- A copy of the FNMVLOGx data sets

Collecting the Information

The following steps describe how to collect the necessary information.

1. Collect the basic information required for most problems. See “Beginning to Document Problems” on page 35.

2. Record the type of output that is incorrect (for example, the panel, messages, or data record). Describe how the output differs from what was expected by answering the following questions:
   - Is all or part of the output missing?
   - Is the output duplicated?
   - Is there more output than expected?
   - Is the information inaccurate?

3. Record the version and release of the VTAM you are using.

4. Obtain a copy of the VTAM configuration by issuing the `D NET,MAJNODES` command from the system console.

5. Clear and reset the FNMVLOGx data sets.

6. Code the NPM DEBUG statement with EUITEST=YES, and also activate the VTAM trace by specifying VTAM=YES on the DEBUG statement. See “DEBUG Statement” on page 110 for more information about the DEBUG statement. You can also use fast path =P.2 to access the NPM online problem determination facilities and activate the EUI RU trace. For more information about using NPM Diagnostic trace options, see “Displaying Diagnostic Trace Options” on page 96.

7. Recreate the problem. A copy of the FNMVLOGx data sets is required by the Support Center.

8. Report the problem to the Support Center after collecting all the required information. For more information about reporting a problem, see “Reporting Problems to the Support Center” on page 7.
Part II. Using Diagnostic Tools

Chapter 6. Using PDI, FNMILOG, SYSPRINT, and PANELID .......... 73
Using the PDI Control Block .......................................................... 73
Using the FNMILOG and SYSPRINT Data Sets .............................. 74
  FNMILOG Data Set ...................................................................... 74
  NPM Virtual Storage Allocation ................................................... 74
  Virtual Storage by Subpool ........................................................... 75
  Configuration Summary from NCP RRT and Initialization Statement .............................................................. 76
  Configuration Hierarchy ............................................................. 77
  Totals by Resource Type Table ....................................................... 78
  Configuration Symbol Resource Table ........................................ 78
  Synonym Chain Distribution Table .............................................. 78
  Sample FNMILOG ...................................................................... 79
  SYSPRINT Data Set ...................................................................... 84
Diagnosing with the PANELID Command ........................................ 87
Diagnosing with the PRINT Command ............................................. 87

Chapter 7. Using the Online Problem Determination Facilities ......... 89
Selecting the Problem Determination Option .................................... 89
Displaying Main Storage ............................................................. 90
  Displaying Virtual Storage .......................................................... 92
  Using Display Commands ........................................................... 92
Displaying Diagnostic Trace Options .............................................. 96
Displaying Diagnostic Options ......................................................... 98
Displaying the Operator Traces ....................................................... 100
Displaying Session Traces ........................................................... 101
Adding a Diagnostic Trap ........................................................... 102
Displaying the Status of Diagnostic Traps ....................................... 104
Displaying Trap Values ............................................................. 106
Taking an NPM Storage Dump ...................................................... 107
Ending NPM Abnormally ........................................................... 108

Chapter 8. NPM DEBUG and TRAP Initialization Statements ......... 109
DEBUG Statement ....................................................................... 110
TRAP Statement ......................................................................... 119
  Determining the Applicable Parameters for Your Trap .................. 125
  Activating NPM Internal Trace Classes for Traps ........................ 126

Chapter 9. Using the NPM Internal Trace ....................................... 127
Finding the Internal Trace Table ..................................................... 127
Understanding Trace Table Entries ................................................. 129
  Dispatcher Trace Entry (Types QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, SR) ........................................... 132
  Message Trace Entry (MG) ........................................................ 133
  Command Trace Entry (SC, RC) ................................................... 134
  VSAM and VTAM Request Parameter List (RPL) Trace Entry (VE, VS) ................................................................. 135
  VTAM Exit Trace Entry (VX) ....................................................... 136
  Logic Trace Entry (LG) ........................................................... 137
  LAN Service Trace Entry (LN) ................................................... 138
  Presentation Service Trace Entry (PS) ........................................ 139
Chapter 10. Using the NPM Message Trace ............................... 149
Finding a Message Trace .......................................................... 149
Message Trace Entry ............................................................... 151
Using PDI, FNMILOG, SYSPRINT, and PANELID

This chapter describes additional tools that can help you diagnose program problems:

- Problem determination information (PDI) control blocks. You can use PDI control blocks to diagnose abends.
- The FNMILOG and SYSPRINT data sets.
- The PANELID and PRINT commands. You can use these commands to help you identify panels.

Using the PDI Control Block

Use the PDI control block when diagnosing an abend. The following abend information is stored in the PDI each time an extended specify task abnormal exit (ESTAE) is entered:

- Module ID
- Abend code
- Load module name
- Abend registers
- Environmental data

Locate the common address table (CAT) to find the first PDI. Register 11 points to the CAT, or you can locate the CAT as follows:

1. PSA control block is at location 0
2. PSA + X’10’ points to the CVT
3. CVT + X’148’ points to the CVT2
4. CVT2 + X’14’ points to the MAT
5. MAT + X’B4’ points to the CAT

Use the worksheets in "Problem Worksheets" on page 155, to record this information.

The four-byte address at X’24’ is the pointer to the PDI control block, FNMPDI or PDI. Use the information in the PDI to build keyword strings to further classify the problem.

Table 1 shows the structure of the PDI.

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>4</td>
<td>PDICBID</td>
<td>Control block identifier (PDI)</td>
</tr>
<tr>
<td>04</td>
<td>4</td>
<td>PDINEXT@</td>
<td>Address of next PDI</td>
</tr>
<tr>
<td>08</td>
<td>8</td>
<td>PDITIME</td>
<td>Time of abend in store-clock format</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>PDIABCC</td>
<td>Abend code</td>
</tr>
</tbody>
</table>
To find the second, third, and fourth PDI (if multiple ESTAEs were entered) follow the PDI chain at offset X’04’, PDINEXT®.

### Using the FNMILOG and SYSPRINT Data Sets

The information contained in the FNMILOG and SYSPRINT data sets is determined by the parameters coded in the NPM initialization statement. Use this information for tuning NPM. For complete information about coding the NPM statement and tuning NPM, see the *NetView Performance Monitor Installation and Customization* book.

#### FNMILOG Data Set

When you code the NPM initialization statement with LIST=YES, the FNMILOG data set contains a listing of the initialization statements from the NPM initialization data sets. NPM lists each statement with the parameters entered in the initialization data sets. It also lists the default parameters and default statements provided by NPM. You can use FNMILOG to verify the coding of initialization statements and to look for error messages.

**Note:** If WARNING flags appear to the left of incorrectly coded statements in the FNMILOG, note the statement number and search the FNMILOG for another occurrence of the statement number to locate the associated NPM error message.

NPM also provides a configuration report. Code the NPM initialization statement with CONFIG=YES to receive the report. The configuration report is in the FNMILOG data set and provides a summary of the configuration established during NPM initialization. You can use this report to verify the values you used in NPM initialization statements. The following sections describe the information contained in the report.

#### NPM Virtual Storage Allocation

Table 2 shows the amount of storage allocated during NPM initialization and where the storage is located. The numbers in the table are rounded to the nearest whole number. For information about how NPM uses this storage, see the *NetView Performance Monitor Installation and Customization* book.

<table>
<thead>
<tr>
<th></th>
<th>Private</th>
<th>CSA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Below 16 Mb</td>
<td>268 K</td>
<td>5 K</td>
<td>273 K</td>
</tr>
<tr>
<td>Above 16 Mb</td>
<td>876 K</td>
<td>105 K</td>
<td>981 K</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1144 K</td>
<td>110 K</td>
<td>1254 K</td>
</tr>
</tbody>
</table>
Virtual Storage by Subpool

The following list shows the abbreviations of the subpools used in FNMILOG to identify the NPM private virtual storage allocation:

<table>
<thead>
<tr>
<th>Name</th>
<th>Subpool</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUF</td>
<td>NPM buffer pool</td>
</tr>
<tr>
<td>CAT</td>
<td>Common address table and related control blocks</td>
</tr>
<tr>
<td>CBS</td>
<td>Control block pool</td>
</tr>
<tr>
<td>CCT</td>
<td>Communication control task</td>
</tr>
<tr>
<td>CPT</td>
<td>Configuration control blocks pool</td>
</tr>
<tr>
<td>FST</td>
<td>File service task buffer</td>
</tr>
<tr>
<td>INIT</td>
<td>Initialization storage (initialization storage is freed when initialization is complete)</td>
</tr>
<tr>
<td>PNL</td>
<td>Presentation services panel buffer pool</td>
</tr>
<tr>
<td>RACF®</td>
<td>Related control blocks</td>
</tr>
<tr>
<td>VDT</td>
<td>VTAM data table</td>
</tr>
</tbody>
</table>

Use storage allocation information in conjunction with the control table from the Performance Measurement Table Display panel (fast path =9.4). For example, if the number of FST control blocks is high in the Virtual Storage Allocation by SUBPOOL Report, you should look at the number of free FSTs in the control table on the Performance Measurement Table Display panel (FNM02OPM). You can use the performance measurement table to help tune NPM. If this number is high, you should define fewer FSTs at initialization. For more information, see the NetView Performance Monitor Installation and Customization book.
Configuration Summary from NCP RRT and Initialization Statement

The header and data fields for this report are taken from the NCP resource resolution table and the initialization statements. The following list describes the fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>The resource class (see “Configuration Hierarchy” on page 77)</td>
</tr>
<tr>
<td>NAME</td>
<td>The name of the resource</td>
</tr>
<tr>
<td>EA</td>
<td>The element address</td>
</tr>
<tr>
<td>NPA</td>
<td>Indicates whether the resource is enabled for NPM network data collection (Yes</td>
</tr>
<tr>
<td>DNC</td>
<td>Indicates whether the resource is enabled for dynamic network collection (Yes</td>
</tr>
<tr>
<td>DR</td>
<td>Indicates whether the resource is enabled for dynamic reconfiguration (Yes</td>
</tr>
<tr>
<td>SCL</td>
<td>The session collection status (Y</td>
</tr>
<tr>
<td>UP</td>
<td>The name of the resource above this resource in the network hierarchy</td>
</tr>
<tr>
<td>NEXT</td>
<td>The name of the next resource in the network hierarchy</td>
</tr>
<tr>
<td>DOWN</td>
<td>The name of the first resource below this resource in the network hierarchy</td>
</tr>
<tr>
<td>DWCT</td>
<td>The down count is the number of resources defined directly below this resource. An example of this count is the number of PUs directly below a link. This count does not include the LUs below the PU. The PU down count would include all LUs directly below it.</td>
</tr>
<tr>
<td>MAXL</td>
<td>The maximum number of LUs on which NPM can collect session data, plus the number of LUs to be excluded.</td>
</tr>
<tr>
<td>SPEED1</td>
<td>The sending speed on a line in bits/sec</td>
</tr>
<tr>
<td>SPEED2</td>
<td>The receiving speed on a line in bits/sec</td>
</tr>
<tr>
<td>CPB</td>
<td>The address of the NPM control block for that resource</td>
</tr>
<tr>
<td>SCB</td>
<td>The address of the session collection control block</td>
</tr>
</tbody>
</table>
Configuration Hierarchy
The configuration hierarchy is based on the NCP RRTs used during initialization and the NPM initialization statements. The configuration hierarchy is defined by the following list:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST</td>
<td>Host name from the HOST initialization statement</td>
</tr>
<tr>
<td>APPL</td>
<td>Application name from the APPL initialization statement</td>
</tr>
<tr>
<td>APPL/S</td>
<td>Application synonym from the APPL initialization statement</td>
</tr>
<tr>
<td>NCP</td>
<td>NCP name from the NPM NCP command</td>
</tr>
<tr>
<td>LINK</td>
<td>Link name from the NCP RRT</td>
</tr>
<tr>
<td>PU</td>
<td>Physical unit name from the NCP RRT</td>
</tr>
<tr>
<td>LU</td>
<td>Logical unit name from the NCP RRT and LUGROUP definition</td>
</tr>
<tr>
<td>LINE</td>
<td>Line name from the NCP RRT</td>
</tr>
<tr>
<td>CLUSTER</td>
<td>Cluster name from the NCP RRT</td>
</tr>
<tr>
<td>TERM</td>
<td>Terminal name from the NCP RRT</td>
</tr>
<tr>
<td>CDRM</td>
<td>Cross-domain resource name from the CDRM initialization statement</td>
</tr>
<tr>
<td>LUGROUP</td>
<td>LU group name as defined in the FNMSCMDS data set</td>
</tr>
<tr>
<td>PHYLINK</td>
<td>NTRI physical link name from the NCP RRT</td>
</tr>
<tr>
<td>LOGLINK</td>
<td>NTRI logical link name from the NCP RRT</td>
</tr>
<tr>
<td>NX25LINK</td>
<td>NPSI link name from the NCP RRT</td>
</tr>
<tr>
<td>XX25LINK</td>
<td>XI link name from the NCP RRT</td>
</tr>
<tr>
<td>NEOLINK</td>
<td>NEO link name from the NCP RRT</td>
</tr>
<tr>
<td>NX25PU</td>
<td>NPSI physical unit name from the NCP RRT</td>
</tr>
<tr>
<td>XX25PU</td>
<td>XI physical unit name from the NCP RRT</td>
</tr>
<tr>
<td>NX25VC</td>
<td>NPSI virtual circuit name from the NCP RRT</td>
</tr>
<tr>
<td>NEOPU</td>
<td>NEO physical unit name from the NCP RRT</td>
</tr>
<tr>
<td>ODLCLNLK</td>
<td>ODLC physical link name from the NCP RRT</td>
</tr>
<tr>
<td>ODLCLNPU</td>
<td>ODLC physical unit name from the NCP RRT</td>
</tr>
</tbody>
</table>
**Totals by Resource Type Table**
The fields in the Total by Resource Type table provide a summary listing of the resources in the configuration listing. The LU count is for local and remote LUs.

**Configuration Symbol Resource Table**
The fields in the Configuration Symbol Resource table provide the following data:
- Number of entries in the symbol table
- Number of synonym chains
- Average length of synonym chain
- Maximum length of synonym chain
- Minimum length of synonym chain

The number of synonym chains is determined by the value specified by the CSRTSIZE parameter of the BUFFERS initialization statement.

**Synonym Chain Distribution Table**
The fields on the Synonym Chain Distribution Table include:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent</td>
<td>Percentage of the maximum length of the synonym chain</td>
</tr>
<tr>
<td>Value</td>
<td>Percent times the maximum length of the synonym chain, rounded upward</td>
</tr>
<tr>
<td>Count</td>
<td>Number of elements on the chain. For optimal CPU utilization, the distribution should be skewed toward the low values, for example, the synonym chains with the shortest length.</td>
</tr>
</tbody>
</table>
Sample FNMILOG

Figure 4 to Figure 8 on page 83 is a sample FNMILOG data set.

Figure 4. Example of FNMILOG Data Set (Part 1 of 5)
Figure 5. Example of FNMILOG Data Set (Part 2 of 5)
FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGMT=YES, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FILEMGFTY=NO, FILE MANAGEMENT
RTMMGMT=YES, RTM MANAGEMENT

FIGURE 6. EXAMPLE OF FNMILOG DATA SET (PART 3 OF 5)
PFKEYS PF1=HELP,PF2=LIST,PF3=END,PF4=TRANSIT,PF5=VOLUME,
PF6=DISTRIB,PF7=BKWD,PF8=FORWARD,PF9=SUMMARY,
PF10=TOP,PF11=BOTTOM,PF12=RETURN,PF13=HELP,PF14=LIST,
PF15=END,PF16=TRANSIT,PF17=VOLUME,PF18=DISTRIB,
PF19=TOP,PF20=BOTTOM,PF21=RETURN,PF22=TOP,
PF23=BOTTOM,PF24=RETURN

DEFAULTS DATE=MDY,GTF=NO,INTERVAL=1,MAXRUSZ=32764,MIN=YES,
MINRUSZ=59,SESSH=Y,VLOG=0,VTAMINT=1

BUFFERS BLDVRP=FNMBLVRP,BUFFNO=0,BUFSEGS=5024,CSRTSIZE=397,
DDBBUF=0,FSTBUFS=3488,MMSGCOUNT=63,PNLBUF=144,QCB=2106,
QCBCHN=BOTTOM,SESSNO=0,SSBHASH=0,SSRTSIZE=13,
TPRB=1024,TPRCHN=BOTTOM,VSRP=S

TASK ALERT=NO,NAME=FNMCCT00,NPMLOG=NO,PRTY=230,REVIEW=NO,
SESSION=NO,SMF=NO,VTAMLOG=NO

TASK ALERT=NO,NAME=FNVDC00,NPMLOG=NO,PRTY=255,REVIEW=NO,
SESSION=NO,SMF=NO,VTAMLOG=NO

TASK ALERT=NO,NAME=FNMLN00,NPMLOG=NO,PRTY=250,REVIEW=NO,
SESSION=NO,SMF=NO,VTAMLOG=NO

TASK ALERT=NO,NAME=FNMDT00,NPMLOG=NO,PRTY=254,REVIEW=NO,
SESSION=NO,SMF=NO,VTAMLOG=NO

TASK ALERT=NO,NAME=FNMFST00,NPMLOG=NO,PRTY=245,REVIEW=NO,
SESSION=NO,SMF=NO,VTAMLOG=NO

TASK ALERT=NO,NAME=FNMFST00,NPMLOG=YES,PRTY=240,REVIEW=YES,
SESSION=YES,SMF=YES,VTAMLOG=YES

CONSOLE CNSID=0,COMMAND=NO,DESC=7,NATLANG=ENGLISH,
OPERATOR=NPMUSER1,PROFILE=FNMDFLT,PROMPT=NO,
ROUTCODE=8,SECURITY=NO

OPERATOR NAME=MIKE,EXTERNAL=NO,PROFILE=DOITALL,RPROFILE=

PROFILE NAME=DOITALL,CONSOLE=YES,CONTROL=YES,DFM=YES,
GLOBAL=YES,GROGMN=YES,LAN=YES,ANALYSIS=YES,
COLLADM=YES,COLLCET=YES,CNTRLFNC=YES,FILEMGT=YES,
ACNTMGT=YES,RTMCGMT=YES,MUXUSER=999,NETANLY=YES,
NETCOLL=YES,NSA=YES,PD=YES,RTMCOLL=YES,SESSANLY=YES,
SESSCOLL=YES,SMF=YES

PRIVATE CSA TOTAL

BELOW 16 MEG 4214K OK 4214K

ABOVE 16 MEG 19K 4K 23K

TOTAL 4232K 4K 4236K

VIRTUAL STORAGE ALLOCATION BY SUBPOOL

CAT 16K CBS 1268K BUF 987K FST 1760K SPS 5K

INIT 15K PNL 148K CPT 18K CCT 1K RACF OK

TOTALS BY RESOURCE TYPE

TYPE TOTAL NPACOLL

HOST 1 0

APPL 2 0

APPL/S 0 0

CDRM 0 0

LUGROUP 0 0

NCP 0 0

NEOLINK 0 0

NX25LIN 0 0

XX25LIN 0 0

FRLMIPU 0 0

FRLMIPU 0 0

ETHERPL 0 0

FRLMIPU 0 0

FRLMIPU 0 0

Figure 7. Example of FNMILOG Data Set (Part 4 of 5)
Figure 8. Example of FNMILOG Data Set (Part 5 of 5)
SYSPRINT Data Set

The SYSPRINT data set shows messages produced when NPM processes FNMSTRT to start collection commands. A special character included in front of each message written in the SYSPRINT data set indicates the type of the message as follows:

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Type of Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>Information</td>
</tr>
<tr>
<td>*</td>
<td>Warning</td>
</tr>
<tr>
<td>@</td>
<td>Error</td>
</tr>
<tr>
<td>!</td>
<td>Severe</td>
</tr>
<tr>
<td>!</td>
<td>Termination</td>
</tr>
</tbody>
</table>

The message may also be prefixed by a number. This number represents the statement number of the command that caused the message to be issued.

Look for errors during this processing to see why data is not being collected or for other data collection errors.

The SYSPRINT data set also shows any panels printed using the PRINT command. A sample SYSPRINT data set is shown in Figure 9 on page 85 and Figure 10 on page 86.
Figure 9. Example of SYSPRINT Data Set (Part 1 of 2)
Figure 10. Example of SYSPRINT Data Set (Part 2 of 2)
Diagnosing with the PANELID Command

If you type PANELID on any command line or in the Select Option field, NPM displays the panel name in the top left corner of the current panel and all subsequent panels. This is helpful if you need to report a problem that involves the panel. Use the PANELID OFF command to suppress the display of the panel name.

Figure 11. Location of Panel ID

Use the panel name as a keyword in the symptom string to communicate to the Support Center the exact panels you were using when you began experiencing problems.

Diagnosing with the PRINT Command

The PRINT command enables you to copy the contents of your current screen to the SYSPRINT file. This can be done in one of two ways:

- You can use the command line or Select Option field to issue a PRINT command and print your current panel.
- You can assign the print function to a PF key so that the command is invoked by pressing the appropriate key. For information about the NPM PF key statement, see the NetView Performance Monitor Installation and Customization book.

Using the PRINT command enables the Support Center to see your panels as you re-create the problem. You can copy the SYSPRINT data set to tape without having to send a hard copy.

The PRINT command adds the date, time, operator ID, terminal ID, and panel ID to each panel that is printed. For example, if your current panel looks like Figure 11 and you type PRINT on the command line or use a PF key assigned to the PRINT command, a copy of the panel, including the date, time, and NPM operator ID, is written to the SYSPRINT data set. The copy of this panel sent to the SYSPRINT data set looks like Figure 12 on page 88.
Diagnosing with PRINT

Figure 12. FNM00PRI as Printed in SYSPRINT Using the PRINT Command

---

88

Version 2 Release 7
Using the Online Problem Determination Facilities

The NPM Online Problem Determination Facilities can help you collect additional data and classify your problem or collect data requested by the Support Center.

Selecting the Problem Determination Option

Use the following procedure to access the Online Problem Determination Facilities:

1. Select option P from the Primary Options panel (FNM00PRI) and press ENTER to display the Problem Determination Facilities panel (FNM09DBM), shown in Figure 13.

       Note: Your profile must specify PD=YES to use the NPM Online Problem Determination Facilities.

       The Problem Determination Facilities panel has nine options. Type one of the following option numbers in the Select Option field and press ENTER:


Figure 13. Problem Determination Facilities Panel (FNM09DBM)
Problem Determination Option

<table>
<thead>
<tr>
<th>Enter ...</th>
<th>For this function ...</th>
<th>Described in ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Display the main storage</td>
<td>“Displaying Main Storage”</td>
</tr>
<tr>
<td>2, 3, 4, 5</td>
<td>Set trace options</td>
<td>“Displaying Diagnostic Trace Options” on page 96</td>
</tr>
<tr>
<td>6</td>
<td>Add a trap</td>
<td>“Adding a Diagnostic Trap” on page 102</td>
</tr>
<tr>
<td>7</td>
<td>Determine status of a trap</td>
<td>“Displaying the Status of Diagnostic Traps” on page 104</td>
</tr>
<tr>
<td>8</td>
<td>Take NPM dumps</td>
<td>“Taking an NPM Storage Dump” on page 107</td>
</tr>
<tr>
<td>9</td>
<td>Abend NPM</td>
<td>“Ending NPM Abnormally” on page 108</td>
</tr>
</tbody>
</table>

Each of these options is described in this chapter.

2. If necessary, update the Host field with the name of any available NPM host, or type LOCAL to refer to the current NPM.

Displaying Main Storage

To display the main storage areas:

1. Select option 1 from the Problem Determination Facilities panel (FNM09DBM) and press ENTER to display the Display Main Storage panel (FNM09DMS), shown in Figure 14.

The Default Format of Display Main Storage panel can display four areas of main storage, labeled A, B, C, and D. The panel displays these areas in windows. The first line of each window is highlighted, and the Area field identifies the window that is active. The Lines/Area field shows how many lines each window contains.

You can format each area of main storage as one of the following:

- A
- B
- C
- D

Figure 14. Default Format of Display Main Storage Panel (FNM09DMS)
DUMP  This format displays 16 bytes in hexadecimal and character format with the storage address and an offset. DUMP is the default.

HEX  This format displays 32 bytes as eight hexadecimal words with the storage address.

CHAR  This format displays 64 bytes in character format with the storage address.

Figure 14 on page 91 shows the default format (DUMP) of the Display Main Storage panel (FNM09DMS). Only window A contains lines on the screen.

The storage area that can be displayed is limited to the common shared storage of the operating system that is not fetch protected and the private virtual storage of the NPM address space. When NPM is executing in key 6, you can inspect VTAM control blocks. However, these control blocks must reside in the common storage area (CSA). Table 3 describes the fields on the Display Main Storage panel (FNM09DMS):

<table>
<thead>
<tr>
<th>Field</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>Character</td>
<td>The Command field. You can enter Display Main Storage commands in this field. These commands control the format of each area and the contents of each area. The commands are described in Using Display Commands on page 93.</td>
</tr>
<tr>
<td>Address:</td>
<td>Hex</td>
<td>The address of the storage being displayed in the current area.</td>
</tr>
<tr>
<td>Symbol:</td>
<td>Character</td>
<td>The name of a symbol that is associated with the current address in the Address field. You can save a maximum of 32 different symbols and associated addresses. To display the storage for a symbol, type the symbol name in the Command field.</td>
</tr>
<tr>
<td>Area:</td>
<td>Character</td>
<td>The name of the current area where commands are active. The areas are labeled A, B, C, and D. For example, if the Area field contained C, the command DUMP would cause the storage in area C to be displayed in DUMP format.</td>
</tr>
<tr>
<td>Lines/Area: A</td>
<td>Decimal</td>
<td>The number of rows of the panel to be reserved for area A. The amount of storage displayed is determined by the format of the area. The maximum number of lines that can be displayed is 24 rows or less depending on the lines available on a terminal. If the number is zero, the area is not displayed but the address for that area is retained. The area can be displayed at a later time by setting the number of lines for the area to a value greater than zero.</td>
</tr>
<tr>
<td>Lines/Area: B</td>
<td>Decimal</td>
<td>The number of rows of the panel to be reserved for area B. The amount of storage displayed is determined by the format of the area.</td>
</tr>
<tr>
<td>Lines/Area: C</td>
<td>Decimal</td>
<td>The number of rows of the panel to be reserved for area C. The amount of storage displayed is determined by the format of the area.</td>
</tr>
<tr>
<td>Lines/Area: D</td>
<td>Decimal</td>
<td>The number of rows of the panel to be reserved for area D. The amount of storage displayed is determined by the format of the area. If the total number of lines to be displayed exceeds the depth of the panel, then the number of lines in each area starting with D is reduced until the maximum number of lines is reached.</td>
</tr>
<tr>
<td>A:</td>
<td>Hex</td>
<td>The current address associated with area A.</td>
</tr>
</tbody>
</table>
Displaying Virtual Storage

If you press the ENTER key when the cursor is positioned on a word displayed in hexadecimal, the content of the word is moved into the Address field, and the storage at that address is displayed in the current area.

Using Display Commands

There are several commands that control the display of storage. The following example shows the general format for the commands:

\[ \text{command}_{\text{name}} | \text{symbol\ area}\_id \]

where:

- **command\_name** is the name of the command to be executed. The commands are described in Table 4.
- **symbol** is the symbol name associated with the storage area.
- **area\_id** is the area where the action occurs. If **area\_id** is not specified, the default is the area identified in the Area field.

**Note:** You cannot specify an **area\_id** with the BACKWARD or FORWARD commands.

Table 4 lists the available display commands:

### Table 4. Display Main Storage Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACKWARD</td>
<td>Scrolls backward.</td>
</tr>
<tr>
<td>CHAR</td>
<td>Displays the current area in character format with 64 bytes displayed for each line.</td>
</tr>
<tr>
<td>DUMP</td>
<td>Displays the current area in dump format with 16 bytes displayed in four hexadecimal words and in character format.</td>
</tr>
<tr>
<td>FORWARD</td>
<td>Scrolls forward.</td>
</tr>
<tr>
<td>HEX</td>
<td>Displays the current area in hexadecimal format with 32 bytes displayed in eight hexadecimal words.</td>
</tr>
</tbody>
</table>

In addition to these commands, symbol, resource name, segment number, and control block names can also be entered as follows:

### Table 5. Display Main Storage Additional commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbol</td>
<td>Is the name of the symbol that was previously defined in the Symbol field of the panel.</td>
</tr>
<tr>
<td>resource_name</td>
<td>Specifies the name of an NPM internal resource (file name, operator ID, profile name, and so on) or the name of a resource defined for collection.</td>
</tr>
</tbody>
</table>
Table 5. Display Main Storage Additional commands (continued)

<table>
<thead>
<tr>
<th>segment_number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the three-digit hexadecimal number that identifies the segment.</td>
<td></td>
</tr>
<tr>
<td>The list of LAN bridges is searched before the list of LAN segments. If a LAN bridge hexadecimal identifier and a LAN segment hexadecimal identifier are identical, the LAN bridge control blocks are displayed.</td>
<td></td>
</tr>
<tr>
<td>Note: Use unique names to identify LAN bridges and LAN segments.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>control_block_name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indicates a control block to be displayed on the current area.</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 lists the names of the control blocks that can be displayed.

**Note:** Some resources are defined as both an internal resource (NPALUs and NPM applications) and as a collectable resource. The internal resource entry is displayed.

Table 6. Display Main Storage Control Block Names

<table>
<thead>
<tr>
<th>Control Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCB</td>
<td>NPM address space control block</td>
</tr>
<tr>
<td>ATCV</td>
<td>VTAM communication vector table</td>
</tr>
<tr>
<td>BBH</td>
<td>Primary NPM buffer block header. BBH replaced BMAP.</td>
</tr>
<tr>
<td>BUF</td>
<td>NPM private buffer pool</td>
</tr>
<tr>
<td>BUFS</td>
<td>Buffer management section of the CAT</td>
</tr>
<tr>
<td>CAT</td>
<td>Common address table</td>
</tr>
<tr>
<td>CBAT</td>
<td>Control block address table</td>
</tr>
<tr>
<td>CCA</td>
<td>Console communication area</td>
</tr>
<tr>
<td>CCT0</td>
<td>FNMCCT00</td>
</tr>
<tr>
<td>COCB</td>
<td>Current OCB</td>
</tr>
<tr>
<td>COE</td>
<td>Configuration object table</td>
</tr>
<tr>
<td>CPMT</td>
<td>CSA performance measurement table</td>
</tr>
<tr>
<td>CSRT</td>
<td>Configuration symbol resource table</td>
</tr>
<tr>
<td>CVT</td>
<td>Communication vector table</td>
</tr>
<tr>
<td>DFM</td>
<td>Data file management section of the CAT</td>
</tr>
<tr>
<td>DSP</td>
<td>Dispatcher section of the CAT</td>
</tr>
<tr>
<td>DSSST</td>
<td>NSI Tables</td>
</tr>
<tr>
<td>DTF</td>
<td>Diagnostic trace facility area in the CAT</td>
</tr>
<tr>
<td>DTH</td>
<td>Diagnostic trace header</td>
</tr>
<tr>
<td>ECB</td>
<td>FNMONL00 ECB list</td>
</tr>
<tr>
<td>EVL</td>
<td>Environment vector list</td>
</tr>
<tr>
<td>FAT</td>
<td>File address table</td>
</tr>
<tr>
<td>FSB</td>
<td>File services buffer pool</td>
</tr>
<tr>
<td>FST0</td>
<td>FNMFST00</td>
</tr>
<tr>
<td>FVT</td>
<td>NPM function vector table</td>
</tr>
<tr>
<td>GCB</td>
<td>Graphics control blocks</td>
</tr>
<tr>
<td>GMS</td>
<td>Graphics section of the CAT</td>
</tr>
<tr>
<td>HOST</td>
<td>Host collection point block</td>
</tr>
<tr>
<td>ITS</td>
<td>Initialization section of the CAT</td>
</tr>
<tr>
<td>LAN</td>
<td>LAN control block</td>
</tr>
<tr>
<td>LBK</td>
<td>LAN bridges by key</td>
</tr>
<tr>
<td>LBN</td>
<td>LAN bridges by status</td>
</tr>
</tbody>
</table>
### Table 6. Display Main Storage Control Block Names (continued)

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBS</td>
<td>LAN bridges by name and status</td>
</tr>
<tr>
<td>LGT</td>
<td>Logon table</td>
</tr>
<tr>
<td>LMN</td>
<td>LAN Managers by adapter</td>
</tr>
<tr>
<td>LPN</td>
<td>LAN bridge pending list</td>
</tr>
<tr>
<td>LSF</td>
<td>LAN segment deferred queue</td>
</tr>
<tr>
<td>LSP</td>
<td>LAN segment pending queue</td>
</tr>
<tr>
<td>LSS</td>
<td>LAN segment started queue</td>
</tr>
<tr>
<td>LST</td>
<td>LAN bridge started list</td>
</tr>
<tr>
<td>MAIN</td>
<td>FNMMAIN</td>
</tr>
<tr>
<td>MAT</td>
<td>Monitor address table</td>
</tr>
<tr>
<td>MSTR</td>
<td>MASTER OCB</td>
</tr>
<tr>
<td>MTCA</td>
<td>FNMMAIN TCA</td>
</tr>
<tr>
<td>MTRC</td>
<td>Message trace table</td>
</tr>
<tr>
<td>NAE</td>
<td>NPM host application entry</td>
</tr>
<tr>
<td>NAT</td>
<td>NPM application table</td>
</tr>
<tr>
<td>NCB</td>
<td>Node control blocks</td>
</tr>
<tr>
<td>NDC</td>
<td>Network data collection section of the CAT</td>
</tr>
<tr>
<td>NLS</td>
<td>FNMNLS00</td>
</tr>
<tr>
<td>NOT</td>
<td>Network object table</td>
</tr>
<tr>
<td>NSIV</td>
<td>NSI vector table</td>
</tr>
<tr>
<td>NVT</td>
<td>NPM services vector table</td>
</tr>
<tr>
<td>NWAB</td>
<td>NetWare address block</td>
</tr>
<tr>
<td>OCB</td>
<td>Operator control blocks</td>
</tr>
<tr>
<td>ONL0</td>
<td>FNMONL00</td>
</tr>
<tr>
<td>OSS</td>
<td>Operating system services section of the CAT</td>
</tr>
<tr>
<td>PCB</td>
<td>Presentation control block</td>
</tr>
<tr>
<td>PCT</td>
<td>Presentation control table</td>
</tr>
<tr>
<td>PMT</td>
<td>Performance measurement table</td>
</tr>
<tr>
<td>PSH</td>
<td>Presentation services header</td>
</tr>
<tr>
<td>PPA</td>
<td>Program-to-program autotask command entry</td>
</tr>
<tr>
<td>PPQ</td>
<td>Program-to-program interface pending list</td>
</tr>
<tr>
<td>PPS</td>
<td>Program-to-program interface sent list</td>
</tr>
<tr>
<td>PTCH</td>
<td>NPM patch module</td>
</tr>
<tr>
<td>QCB</td>
<td>Queue control blocks</td>
</tr>
<tr>
<td>RTP</td>
<td>Resource table prefix</td>
</tr>
<tr>
<td>SCA</td>
<td>Session communications area</td>
</tr>
<tr>
<td>SCB</td>
<td>Session control blocks</td>
</tr>
<tr>
<td>SCT</td>
<td>SMF control table</td>
</tr>
<tr>
<td>SDB</td>
<td>Session data buffers</td>
</tr>
<tr>
<td>SDC</td>
<td>Session data collection section of the CAT</td>
</tr>
<tr>
<td>SDC0</td>
<td>FNMSDC00</td>
</tr>
<tr>
<td>SDH</td>
<td>Session data headers</td>
</tr>
<tr>
<td>SDT</td>
<td>Session data table</td>
</tr>
<tr>
<td>SNT</td>
<td>Subarea number table</td>
</tr>
<tr>
<td>SSBH</td>
<td>SSB hash table</td>
</tr>
<tr>
<td>SSBV</td>
<td>SSB NSI vectors</td>
</tr>
</tbody>
</table>
Table 6. Display Main Storage Control Block Names (continued)

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSCT</td>
<td>NSI SS control table</td>
</tr>
<tr>
<td>SSRT</td>
<td>System symbol resource table</td>
</tr>
<tr>
<td>SVT</td>
<td>System services vector table</td>
</tr>
<tr>
<td>SYS</td>
<td>System OCB</td>
</tr>
<tr>
<td>TCA</td>
<td>Task control areas</td>
</tr>
<tr>
<td>TPEX</td>
<td>TPEX section of the CAT</td>
</tr>
<tr>
<td>TPRB</td>
<td>Teleprocessing request blocks</td>
</tr>
<tr>
<td>TRC</td>
<td>Next trace entry</td>
</tr>
<tr>
<td>TRCB</td>
<td>First trace entry</td>
</tr>
<tr>
<td>TXB</td>
<td>Transaction program blocks</td>
</tr>
<tr>
<td>UIB</td>
<td>User identification blocks</td>
</tr>
<tr>
<td>UPB</td>
<td>User profile blocks</td>
</tr>
<tr>
<td>USER</td>
<td>User common area. This command works if at least one exit is installed and HSTCOLL=YES is specified.</td>
</tr>
<tr>
<td>VDC0</td>
<td>FNMVDC00 control block</td>
</tr>
<tr>
<td>VDT</td>
<td>FNMVDT VTAM statistic data table</td>
</tr>
<tr>
<td>VSAM</td>
<td>VSAM application control blocks</td>
</tr>
<tr>
<td>VSRB</td>
<td>VSAM request blocks</td>
</tr>
<tr>
<td>VSX</td>
<td>VSAM section of the CAT</td>
</tr>
<tr>
<td>VTAM</td>
<td>VTAM application control block</td>
</tr>
<tr>
<td>VVE</td>
<td>VTAM version entry</td>
</tr>
<tr>
<td>WTG</td>
<td>FNMONL00 Where-To-Go list</td>
</tr>
</tbody>
</table>
Displaying Diagnostic Trace Options

To display diagnostic trace options:

1. Select option 2 from the Problem Determination Facilities panel (FNM09DBM) to display the Display/Modify Diagnostic Trace Options panel (FNM09DTF), shown in Figure 15.

2. Use Table 7 to complete the Diagnostic Trace Options panel. If you do not type OFF on an entry, the system defaults to the current entry. For the formats of trace entries, see “Using the NPM Internal Trace” on page 127.

3. When you complete your entries, press ENTER to submit your changes to NPM.

You can cancel any changes to the trace options using the END function.

Table 7. Fields in the Diagnostic Trace Options Panel (FNM09DTF)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Internal Trace</td>
<td>Required</td>
<td>ON  Enable the internal trace. OFF Disable the internal trace.</td>
</tr>
<tr>
<td>Dispatcher</td>
<td>Required</td>
<td>ON  Trace dispatcher events. OFF Do not trace dispatcher events.</td>
</tr>
<tr>
<td>Enable Message Trace</td>
<td>Required</td>
<td>ON  Enable the update of the message trace table. OFF Stop the update of the message trace table.</td>
</tr>
<tr>
<td>Dispatcher Return</td>
<td>Required</td>
<td>ON  Trace dispatcher return events. OFF Do not trace dispatcher return events.</td>
</tr>
<tr>
<td>Enable EUI RU Trace</td>
<td>Required</td>
<td>ON  Enable the EUI RU trace. OFF Disable the EUI RU trace.</td>
</tr>
<tr>
<td>Messages</td>
<td>Required</td>
<td>ON  Trace messages. OFF Do not trace messages.</td>
</tr>
<tr>
<td>Enable TCP/IP Trace</td>
<td>Required</td>
<td>Trace levels for FNMSNM00, FNMSNM06, FNMTLN00, and FNMTGUI tasks. See DEBUG Statement on page 110 for an explanation of the possible values.</td>
</tr>
<tr>
<td>VTAM Errors</td>
<td>Required</td>
<td>ON  Trace VTAM request errors and exit routines. OFF Do not trace VTAM request errors and exit routines.</td>
</tr>
</tbody>
</table>
Table 7. Fields in the Diagnostic Trace Options Panel (FNM09DTF) (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Required</th>
<th>ON</th>
<th>OFF</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable GTF Trace</td>
<td>Required</td>
<td>ON</td>
<td>OFF</td>
<td>Write NPM internal trace entries to GTF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not write NPM internal trace entries to GTF.</td>
</tr>
<tr>
<td>VTAM SPO</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace the secondary program operator (SPO) application program commands and messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace the commands and messages.</td>
</tr>
<tr>
<td>Trace Control Blocks</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace control blocks to GTF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not write control blocks to GTF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The GTF option must also be set to ON to trace control blocks to GTF.</td>
</tr>
<tr>
<td>VSAM Errors</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace VSAM request errors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace VSAM request errors.</td>
</tr>
<tr>
<td>Trace I/O Buffers</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Write I/O buffers to GTF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not write I/O buffers to GTF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The GTF option must also be set to ON to trace control blocks to GTF.</td>
</tr>
<tr>
<td>OS Services</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace calls and returns through the operating system services.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace requests of the operating system services.</td>
</tr>
<tr>
<td>Router</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace NPM-to-NPM router requests and responses.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace NPM-to-NPM router requests and responses.</td>
</tr>
<tr>
<td>Enable Traps</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Enable all traps.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Disable all traps.</td>
</tr>
<tr>
<td>Installation-Wide Exits</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace installation-wide exit calls.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace installation-wide exit calls.</td>
</tr>
<tr>
<td>LAN Services</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace LAN services calls.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace LAN services calls.</td>
</tr>
<tr>
<td>VTAM Statistics</td>
<td>Required</td>
<td>ON</td>
<td></td>
<td>Trace VTAM statistics data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF</td>
<td></td>
<td>Do not trace VTAM statistics data.</td>
</tr>
<tr>
<td>Host</td>
<td>Display Only</td>
<td></td>
<td></td>
<td>The name of the host for which diagnostic options are displayed.</td>
</tr>
</tbody>
</table>
Displaying Diagnostic Options

To display or modify the diagnostic options:

1. Select option 3 from the Problem Determination Facilities panel (FNM09DBM) to display the Display/Modify Options panel (FNM09OPT), shown in Figure 16.

2. Use Table 8 to complete the Display/Modify Options panel.

3. When you complete your entries, press ENTER to submit your changes to NPM.
   
   You can cancel any changes to the trace options using the END function.

Table 8. Fields in the Display/Modify Options Panel (FNM09OPT)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test NPM</td>
<td>Display Only</td>
<td>Shows the TEST option chosen on the NPM DEBUG initialization statement:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON: NPM is being tested.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: NPM is not being tested.</td>
</tr>
<tr>
<td>Print Members</td>
<td>Required</td>
<td>ON: Print source statement as read from FNMSCMDS on SYSPRINT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Do not print source statement.</td>
</tr>
<tr>
<td>Message Module</td>
<td>Required</td>
<td>ON: Insert module name into message text.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Do not modify messages with the issuing module name.</td>
</tr>
<tr>
<td>TELNET Trace Output</td>
<td>Required</td>
<td>ON: SYSPRINT print TELNET trace to SYSPRINT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: SYSLOG print TELNET trace to SYSLOG.</td>
</tr>
<tr>
<td>Print Message</td>
<td>Required</td>
<td>ON: Print all messages to SYSPRINT.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Print only those messages that are routed to SYSPRINT.</td>
</tr>
<tr>
<td>Save Control Blocks</td>
<td>Display Only</td>
<td>Shows the option chosen on the NPM DEBUG initialization statement.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON: Control blocks are saved and not released.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Control blocks are released.</td>
</tr>
<tr>
<td>Panel Test</td>
<td>Required</td>
<td>ON: Test panel definitions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Does not test NPM panels.</td>
</tr>
</tbody>
</table>

Note: You can reset this option using the PNLTEST command.
Table 8. Fields in the Display/Modify Options Panel (FNM09OPT) (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMODE=31</td>
<td>Display Only</td>
<td>Shows the AMODE in which NPM is operating:</td>
</tr>
<tr>
<td>ON</td>
<td>NPM is executing in AMODE=31.</td>
<td></td>
</tr>
<tr>
<td>OFF</td>
<td>NPM is executing in AMODE=24.</td>
<td></td>
</tr>
<tr>
<td>Host</td>
<td>Display Only</td>
<td>The name of the host for which diagnostic options are displayed.</td>
</tr>
</tbody>
</table>
Displaying the Operator Traces

To display or modify the presentation services trace for an operator:

1. Select option 4 from the Problem Determination Facilities panel (FNM09DBM) to display the Display/Modify Operator Trace panel (FNM09OPR), shown in Figure 17.

```
FNM09OPR       NPM V2R7 5655-043
PROBLEM DETERMINATION
DISPLAY/MODIFY OPERATOR TRACE

Command ===>          
User ID ===> OPER01   
Trace ===> ON  (ON|OFF)

Host: LOCAL

Press ENTER to set Trace or END to terminate request

PF 1=HELP  2=PRINT  3=END  12=RETURN  15=NCP  16=ROUTER
PF17=LINE  20=APPL  24=TERMINAL
```

*Figure 17. Display/Modify Operator Trace Panel (FNM09OPR)*

2. Use Table 9 to complete the Operator Trace panel. To determine an operator’s current status, enter the operator ID and press ENTER (with the TRACE field blank).

You can cancel any changes to the trace options using the END function.

```
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID</td>
<td>Required</td>
<td>The user ID that is to have its presentation trace status changed.</td>
</tr>
<tr>
<td>Trace</td>
<td>Optional</td>
<td>blank, ON, OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Return the current trace status of the operator. Set the operator trace status to ON. Set the operator trace status to OFF.</td>
</tr>
<tr>
<td>Host</td>
<td>Display Only</td>
<td>The name of the host on which the operator resides.</td>
</tr>
</tbody>
</table>
```

*Table 9. Fields in the Operator Trace Panel (FNM09OPR)*

Presentation services trace entries are placed in the NPM internal trace table when the operator trace status is set to ON. Using the display option, these trace entries can be viewed online if you enter the TRC symbol and scroll backward. The diagnostic traps for presentation services are then checked.
Displaying Session Traces

To display or modify the session trace for a resource:

1. Select option 5 from the Problem Determination Facilities menu (FNM09DBM) to display the Display/Modify Session Trace panel (FNM09OPS), shown in Figure 18:

![Display/Modify Session Trace Panel (FNM09OPS)](image)

2. Use Table 10 to complete the Session Trace panel. To determine the trace status of a specific resource, enter the resource name and press ENTER (with the STATUS field blank).

   You can cancel any changes to the trace options using the END function.

Table 10. Fields in the Session Trace Panel (FNM09OPS)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Required</td>
<td>The resource name that is to have its session trace status changed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The resource must be able to accept a start session collection command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Valid resources are application names (not synonyms) and logical units.</td>
</tr>
<tr>
<td>Status</td>
<td>Optional</td>
<td><strong>blank</strong> Return the current trace status of the resource.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ON</strong> Set the session trace status to ON.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>OFF</strong> Set the session trace status to OFF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With session trace status set to ON for a resource, session trace entries are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>placed in the NPM internal trace table and session data is written to GTF for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the resource. The GTF option must also be set to ON before starting this option.</td>
</tr>
<tr>
<td>Host</td>
<td>Display Only</td>
<td>The name of the host of the resource.</td>
</tr>
</tbody>
</table>
Adding a Diagnostic Trap

To add a diagnostic trap:

1. Select option 6 from the Problem Determination Facilities panel (FNM09DBM) to display the Add Diagnostic Trap panel (FNM09TRA), shown in Figure 19:

   ![Figure 19. Add Diagnostic Trap panel (FNM09TRA)](image)

   **Table 11. Fields in the Add Diagnostic Trap Panel (FNM09TRA)**

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trap ID</td>
<td>Required</td>
<td>Four characters identifying the trap being added.</td>
</tr>
<tr>
<td>Trap Type</td>
<td>Required</td>
<td>The type of trap: \n   MSG: Message diagnostic trap \n   CALL: Dispatcher call diagnostic trap \n   RETURN: Dispatcher call return diagnostic trap \n   SVC: Supervisor call diagnostic trap \n   SVC: Supervisor call return diagnostic trap \n   VTAM: VTAM error diagnostic trap \n   VSAM: VSAM error diagnostic trap \n   PSR: Presentation services diagnostic trap \n   RTR: Router diagnostic trap \n   LOGIC: Logic error diagnostic trap.</td>
</tr>
<tr>
<td>Status</td>
<td>Required</td>
<td>ACTIVE: The initial status of the trap is active. \n   INACTIVE: The initial status of the trap is inactive.</td>
</tr>
<tr>
<td>Action</td>
<td>Required</td>
<td>SDUMP: Cause NPM to take a system dump of NPM. \n   SNAP: Cause NPM to take a snap dump of NPM. \n   SUSPEND: Cause NPM to suspend the update of the NPM internal trace table. \n   NOOP: Take no action other than to write message FNM990I. \n   ABEND: Cause NPM to abend with a user 3999 abend code.</td>
</tr>
<tr>
<td>Count</td>
<td>Required</td>
<td>A value from 1 to 255. When the conditions of this trap are met the indicated number of times, the status of the trap is set to disabled, and message FNM992I is issued.</td>
</tr>
</tbody>
</table>

2. Use **Table 11** to complete the Add Diagnostic Trap panel.

3. When you complete your entries, press ENTER to submit your changes to NPM.

   You can cancel any changes to the trace options using the END function.

   ![Figure 19. Add Diagnostic Trap panel (FNM09TRA)](image)
Table 11. Fields in the Add Diagnostic Trap Panel (FNM09TRA) (continued)

<table>
<thead>
<tr>
<th>Field</th>
<th>Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message ID</td>
<td>Optional</td>
<td>The message for a message-type trap.</td>
</tr>
<tr>
<td>Calling Module</td>
<td>Optional</td>
<td>The name of an NPM module for a dispatcher or SVC type trap.</td>
</tr>
<tr>
<td>Called Module</td>
<td>Optional</td>
<td>The name of an NPM module being called in a dispatcher or SVC type trap.</td>
</tr>
<tr>
<td>Return Code</td>
<td>Optional</td>
<td>The return code to be checked for a dispatcher or SVC return type trap.</td>
</tr>
<tr>
<td>Node/File Name</td>
<td>Optional</td>
<td>The name of a VTAM node or a VSAM file to be checked.</td>
</tr>
<tr>
<td>Feed Back</td>
<td>Optional</td>
<td>Four bytes of hexadecimal data for a VTAM or VSAM type trap.</td>
</tr>
<tr>
<td>Sense Codes</td>
<td>Optional</td>
<td>Four bytes of hexadecimal data for a VTAM type trap.</td>
</tr>
<tr>
<td>Request code</td>
<td>Optional</td>
<td>VTAM code for the API macro.</td>
</tr>
<tr>
<td>User ID</td>
<td>Optional</td>
<td>The name of an operator ID to be checked for a presentation services trap.</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Optional</td>
<td>The last four characters of the panel name to be checked in a presentation services trap.</td>
</tr>
<tr>
<td>Flags</td>
<td>Optional</td>
<td>Four hexadecimal bytes of data containing the flags to be checked with presentation services flags.</td>
</tr>
<tr>
<td>Mask</td>
<td>Optional</td>
<td>Four hexadecimal bytes of data containing the mask to be added to the presentation services flags before being checked.</td>
</tr>
<tr>
<td>User ID</td>
<td>Optional</td>
<td>The operator ID that is to be checked for a router request.</td>
</tr>
<tr>
<td>Module Name</td>
<td>Optional</td>
<td>The module name that is being called for a router request.</td>
</tr>
<tr>
<td>Message Number</td>
<td>Optional</td>
<td>A decimal message number that will be checked when an error response is returned on a router request.</td>
</tr>
<tr>
<td>Logic code</td>
<td>Optional</td>
<td>A decimal number that corresponds to a logic error code.</td>
</tr>
<tr>
<td>Module name</td>
<td>Optional</td>
<td>The name of the module that issued the logic error.</td>
</tr>
</tbody>
</table>
Displaying the Status of Diagnostic Traps

To display the status of diagnostic traps:

1. Select option 7 from the Problem Determination Facilities panel (FNM09DBM) to display the Display/Modify Trap Status panel (FNM09TRS), shown in Figure 20.

2. Use Table 12 to complete the Trap Status panel.

Table 12. Fields in the Trap Status Panel (FNM09TRS)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Display Only</td>
<td>The name of the host for which a trap status is to be displayed:</td>
</tr>
<tr>
<td>Status</td>
<td>Display Only</td>
<td>The current status of diagnostic traps.</td>
</tr>
<tr>
<td>Traps Defined</td>
<td>Display Only</td>
<td>The number of diagnostic traps that were defined at initialization or dynamically.</td>
</tr>
<tr>
<td>Traps Available</td>
<td>Display Only</td>
<td>The number of diagnostic trap slots available to define additional traps.</td>
</tr>
<tr>
<td>Option</td>
<td>Optional</td>
<td>/ Display details about the trap.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A Change the status of the trap to ACTIVE.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>I Change the status of the trap to INACTIVE.</td>
</tr>
<tr>
<td>ID</td>
<td>Display Only</td>
<td>The identifier assigned to the diagnostic trap.</td>
</tr>
<tr>
<td>Status</td>
<td>Display Only</td>
<td>The current status of the diagnostic trap:</td>
</tr>
</tbody>
</table>

Figure 20. Display/Modify Trap Status Panel (FNM09TRS)
<table>
<thead>
<tr>
<th>Type</th>
<th>Display Only</th>
<th>The type of trap:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG</td>
<td>Display Only</td>
<td>Message diagnostic trap</td>
</tr>
<tr>
<td>CALL</td>
<td>Dispatcher call diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>RETURN</td>
<td>Dispatcher call return diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>SVC</td>
<td>Supervisor call diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>SVCR</td>
<td>Supervisor call return diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>VTAM error diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>VSAM</td>
<td>VSAM error diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>PSR</td>
<td>Presentation services diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>RTR</td>
<td>Router diagnostic trap</td>
<td></td>
</tr>
<tr>
<td>LOGIC</td>
<td>Logic error diagnostic trap</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Display Only</th>
<th>The action taken when the conditions of the trap are met:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDUMP</td>
<td>Display Only</td>
<td>Issues an SDUMP request for a dump of virtual storage.</td>
</tr>
<tr>
<td>SNAP</td>
<td>Issues a SNAP request for the NPM address space.</td>
<td></td>
</tr>
<tr>
<td>SUSPEND</td>
<td>Suspend the update of the NPM internal trace table. The checking for other traps continues. See “Displaying Trap Values” on page 106. This panel permits you to display the NPM internal trace table.</td>
<td></td>
</tr>
<tr>
<td>NOOP</td>
<td>Takes no action other than writing message FNM990L.</td>
<td></td>
</tr>
<tr>
<td>ABEND</td>
<td>Abends with user code 3999.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Count</th>
<th>Display Only</th>
<th>A value from 1 to 255. The number of times the conditions for the trap were met.</th>
</tr>
</thead>
</table>

| Maximum    | Display Only | The maximum number of times the condition for the trap can be met before the trap is disabled. |
Displaying Trap Values

To display the values associated with a trap:

1. Select option 7 from the Problem Determination Facilities panel (FNM09DBM) to display the Display/Modify Trap Status panel (FNM09TRS).

2. Type / in the option field on the Display/Modify Trap Status panel and press ENTER. A panel is displayed showing information about the trap. For example, the panel shown in Figure 21 displays information about a message trap.

There are nine other displays, one for each type of trap. The fields on each trap display are the same as the ones used to add a trap. To update the trap fields, move the cursor to the Status, Action, or Count field, change the field, and press ENTER.

Table 13 describes the fields that are displayed for a diagnostic trap message.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trap ID</td>
<td>Display</td>
<td>The identifier assigned to the diagnostic trap.</td>
</tr>
<tr>
<td>Status</td>
<td>Display</td>
<td>The current status of the diagnostic trap:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACTIVE: The trap is active.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>INACTIVE: The trap is inactive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DISABLED: The trap is disabled.</td>
</tr>
<tr>
<td>Action</td>
<td>Display</td>
<td>The action taken when the conditions of the trap are met:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDUMP: Issues an SDUMP request for a dump of virtual storage.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SNAP: Issues a SNAP request for the NPM address space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUSPEND: Suspends the update of the NPM internal trace table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The checking for other traps continues. See Displaying Trap Values. This panel permits you to display the NPM internal trace table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOOP: Takes no action other than writing message FNM990I.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABEND: Abends with user code 3999.</td>
</tr>
<tr>
<td>Count</td>
<td>Display</td>
<td>A value from 1 to 255. The number of times the conditions for the trap are met.</td>
</tr>
<tr>
<td>Message ID</td>
<td>Display</td>
<td>The message number.</td>
</tr>
</tbody>
</table>
Taking an NPM Storage Dump

To take an NPM storage dump:

1. Select option 8 from the Problem Determination Facilities panel (FNM09DBM) to display the Take An NPM Storage Dump panel (FNM09DMP), shown in Figure 22.

2. Use Table 14 to complete the NPM Storage Dump panel.

3. Press ENTER to take a storage dump of NPM.

You can cancel any changes to the dump options using the END function.

Table 14. Fields in the NPM Storage Dump Panel (FNM09DMP)

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Entry/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option</td>
<td>Required</td>
<td>SNAP Takes a formatted snap dump of the NPM address space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SDUMP Takes an unformatted dump of the NPM address space. The dump is sent to the operat</td>
</tr>
</tbody>
</table>
Ending NPM Abnormally

To end NPM abnormally:

1. Select option 9 from the Problem Determination Facilities panel (FNM09DBM) to display the Abend NPM panel (FNM09ABN), shown in Figure 23.

2. Enter a decimal number from 4000 to 4095 that is used as the abend code and press ENTER.

   You can cancel any changes to the abend options using the END function.

Figure 23. Abend NPM Panel (FNM09ABN)
The NPM DEBUG and TRAP initialization statements are read from a partitioned data set (PDS) member during NPM startup. These statements control the types of traces and traps you set up to help in problem determination. The problem determination statements are coded in the FNMPDET member of the FNMPARM partitioned data set.

There is no default member name for problem determination statements. You must specify a member name to change the default options and to define traps.

The name is provided in an EXEC statement in the NPM startup procedure with the keyword PD, as in the following example:

```
// EXEC PGM=FNMMAIN,PARM='INIT=FNMINIT,PD=FNMPDET'
```

The DEBUG and TRAP initialization statements have the same syntax as other initialization statements. For information about these syntax requirements, see the *NetView Performance Monitor Installation and Customization* book.
## DEBUG Statement

The DEBUG initialization statement defines options and values used by NPM to collect data for problem determination. For information about reading the traces produced by this statement, see "Using the NPM Internal Trace" on page 127.

The following list describes the parameters you can code in the DEBUG initialization statement:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>BUFSCAN</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>BUFSCAN</td>
<td>nnnnn</td>
<td></td>
</tr>
<tr>
<td>CBS</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>CBS</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>DSP</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>ESTAE</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>ESTAE</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>GTF</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>GTF</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>IOB</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>IOB</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>LAN</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>LAN</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>MSG</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>MSG</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>MSGMOD</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>MSGMOD</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>MTRACE</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>MTRACE</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>MTRCSIZE</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>MTRCSIZE</td>
<td>nn</td>
<td></td>
</tr>
<tr>
<td>PNLTEST</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>PNLTEST</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>PRINTMBR</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>PRINTMBR</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>PRINTMSG</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>PRINTMSG</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>RCYDUMP</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>RCYDUMP</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>RTN</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>RTR</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>RTR</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SAVECB</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>SAVECB</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SDUMP</td>
<td>FNMSDUMP</td>
<td></td>
</tr>
<tr>
<td>SDUMP</td>
<td>ddname</td>
<td></td>
</tr>
<tr>
<td>SNAP</td>
<td>FNMSNAP</td>
<td></td>
</tr>
<tr>
<td>SNAP</td>
<td>ddname</td>
<td></td>
</tr>
<tr>
<td>SPO</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>SPO</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SVC</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>SVC</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>SYSDUMP</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>SYSOUT</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>SYSOUT</td>
<td>sysout_class</td>
<td></td>
</tr>
<tr>
<td>TCPIPO</td>
<td>SYSPRINT</td>
<td></td>
</tr>
<tr>
<td>TCPIPO</td>
<td>SYSLOG</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>TRAP</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>TRAP</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>TRAPCNT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>TRAPCNT</td>
<td>nnn</td>
<td></td>
</tr>
<tr>
<td>TRCSIZE</td>
<td>512</td>
<td></td>
</tr>
<tr>
<td>TRCSIZE</td>
<td>nnnnn</td>
<td></td>
</tr>
<tr>
<td>UXT</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>UXT</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>VSM</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>VSM</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>VTAMIM</td>
<td>FNMXIVTO</td>
<td></td>
</tr>
<tr>
<td>VTAMIM</td>
<td>module_name</td>
<td></td>
</tr>
</tbody>
</table>

The following list describes the parameters you can code in the DEBUG initialization statement:
Specifies whether VTAM errors are traced.

**Note:** This option also traces the VTAM exit calls.

Specifies how often the NPM buffer pool is checked for overlaid buffers. When a buffer that is not valid is detected, NPM abends with U0006. For more information, see the *NetView Performance Monitor Messages and Codes* book.

You can code BUFSCAN as any number from 0 to 99999. For example, BUFSCAN=1 indicates that every time the buffer manager is called, the buffer pool is checked; BUFSCAN=2 indicates every other time; and BUFSCAN=10 indicates every tenth time the buffer manager is called. The default of zero indicates that the buffer manager never checks the buffer pool.

**Note:** When BUFSCAN is nonzero, the virtual storage requirements of NPM increase by four times the BUFSEGS value on the BUFFERS initialization statement. This also increases CPU usage.

Specifies whether NPM traces the control blocks to GTF. To trace the data areas, the GTF option must also be YES.

Specifies whether NPM traces the requests to the NPM online dispatcher. The trace is placed in the NPM internal trace table. This option controls the most important diagnostic option in NPM because the dispatcher controls the flow processing on the entire online system. DSP=YES is required for CALL type traps.
Indicates whether ESTAE routines are disabled while NPM is being tested. If ESTAE=NO, there is no recovery from any error encountered during processing, including installation-wide exit processing.

Specifies whether NPM passes NPM internal trace entries and control blocks to the generalized trace facility (GTF). You must specify GTF=YES to trace control blocks to the generalized trace facility.

Specifies whether NPM passes I/O buffers from VTAM, VSAM, and sequential access methods to the NPM internal trace facility. If GTF=YES, the I/O buffer is written to GTF.

Specifies whether NPM traces the text of LAN requests made to the NetView program. The default is NO.

Specifies whether NPM traces the text of messages written to terminals and the SYSPRINT log. You must specify MSG=YES to update the NPM internal trace table and to activate any message traps.
Specifies whether the last five characters of the module that issued a message should be placed in the text of the message after the message ID. The module name that appears in the message is the name of the routine that caused the message to be displayed but might not be the module that actually generated the message.

Adding the module name to a message increases the length of the message. Some messages might be truncated.

Specifies whether NPM adds messages to the NPM message trace table. The MTRCSIZE parameter specifies the number of messages to be retained in memory. If MTRACE=NO, messages are not placed in the trace table until MTRACE is changed to YES.

**Note:** Specify MSG=YES on the DEBUG statement for the message text to be placed in the message trace table.

Specifies the number of message trace entries to be allocated in NPM storage. Each entry is 96 bytes long.

You can code MTRCSIZE to any number from 0 to 9999. If you use a value of zero, there is no message trace table. The actual number used by NPM might be rounded up to a greater value to allocate storage to the next control block.

Specifies the setting of the initial panel test option when an operator logs on to NPM. When the panel test option is in effect, the panel name is displayed in the upper left corner of the panel. The last line of the panel displays information about the current panel.

You can change the option for panel testing at any time by using the PNLTEST command. You enter this command at the terminal on the command line.

**Note:** If PNLTEST=YES, panels are not read at initialization.
Specifies whether NPM copies the statements read from the FNMSCMDS data set to the SYSPRINT data set. You can use this option to determine the cause of syntax errors.

Specifies whether NPM writes all messages to the SYSPRINT data set. This option includes all messages written to operator terminals that normally would not be logged in the SYSPRINT data set.

Specifies whether a dump is taken when a recoverable error occurs. For example, recoverable errors can occur in the execution of a user exit routine.

Specifies whether NPM traces the return of control through the dispatcher that resulted from the dispatcher call request. This trace is recorded in the NPM internal trace table.

Specifies whether NPM traces requests through the NPM-to-NPM router. The router controls the flow between application programs (screen programs) and transaction programs. Security checking is also a function of the router. Remote requests from the originating NPM are traced as local requests at the target NPM. This trace is recorded in the NPM internal trace table.
Specifies whether NPM releases storage for control blocks during termination processing. This option is also used to retain the storage for initialization control blocks.

\[
\begin{align*}
&\text{SDUMP=FNMSDUMP} \\
&\text{SDUMP=ddname}
\end{align*}
\]

Specifies the ddname of a data set to contain a machine-readable copy of a system dump. This option is used for TRAP dumps and problem determination.

\[
\begin{align*}
&\text{SNAP=FNMSNAP} \\
&\text{SNAP=ddname}
\end{align*}
\]

Specifies the ddname of a SYSOUT data set for a SNAP dump initiated from the console or from the problem determination panels. If the data set is not defined in the JCL, a dynamically allocated SYSOUT data set is defined.

\[
\begin{align*}
&\text{SPO=NO} \\
&\text{SPO=YES}
\end{align*}
\]

Specifies whether NPM traces the text of the command and message buffers from the SENDCMD and RCVCMD processors. These entries can be used to check the text of buffers used to start and stop the VTAM buffer trace. This trace is placed in the NPM internal trace table.

\[
\begin{align*}
&\text{SVC=NO} \\
&\text{SVC=YES}
\end{align*}
\]

Specifies whether NPM traces the returns and calls to operating system interface routines. These routines include the initialization routines after the trace table is defined and the options are set. Also, routines in the FNMFST00, FNMCCT00, and FNMSDC00 subtasks use this interface to call subroutines. This trace is recorded in the NPM internal trace table.

**Note:** Some calls to the operating system interface routines do not result in a trace entry because they do not request a save area.

\[
\begin{align*}
&\text{SYSDUMP=NO} \\
&\text{SYSDUMP=YES}
\end{align*}
\]

Specifies whether a system dump is taken during abend processing. If SYSDUMP=YES, dumps are written to one of the SYS1.DUMP data sets. Setting SYSDUMP=YES allows you to remove the SYSMDUMP data set in the NPM startup procedure to allow DASD to be
used for other tasks. If multiple NPM dumps are taken, each dump is saved in the SYS1.DUMP data sets. If SYSDUMP=NO, only the last dump is saved in the SYSMDUMP data set. This option is valid only for ABEND type dumps.

```
,SYSOUT=A
,SYSOUT=sysout_class
```

Specifies the SYSOUT class for dynamically allocated data sets.

```
,TCP1PT=0
,TCP1PT=nnn
```

Specifies the trace level for the FNMSNM00, FNMSNM06, FNMTLN00, and FNMTGUI tasks.

The following settings are supported for FNMSNM00 and FNMSNM06 tasks:

- **0**  No debug/trace.
- **1**  FNMSN general component trace.
- **2**  FNMSN CB component trace.
- **4**  WinSnmp trace.
- **8**  SNMP in/out packet trace.
- **10** SNMP in/out buffer trace.
- **20** Print contents of free CB chains.
- **40** Print SNMP collection summary information.
- **80** Print IP to LU Map Table.
- **100** WinSnmp storage trace.
- **200** Trace CBs queued to Network Collection.
- **767** Select all traces.

The following settings are supported for FNMTLN00:

- **0**  No trace.
- **1**  General trace.
- **2**  General and control block trace.

The following settings are supported for FNMTGUI:

- **0**  No trace.
- **1**  General trace.

**Note:** For trace information to be written, DD statements for the required task must be added to the NPM procedure:

```
//FNMTNL00 DD SYSOUT=*  
//FNMSNM00 DD SYSOUT=*  
//FNMSNM06 DD SYSOUT=*  
//FNMTGUI DD SYSOUT=*  
```
Specifies whether the output for the TCP/IP functions is the SYSPRINT or the SYSLOG.

If SYSLOG is selected, the SYSLOGD and INETD daemons must be started before TCP/IP. For more information about SYSLOGD and INETD daemons refer to the *OS/390 TCP/IP Open Edition: Configuration Guide*.

Specifies whether NPM is tested. This option is used to simulate functions in NPM that require system resources, such as common storage area. It is useful to test PTFs that update and use session collection.

**Note:** This option has the effect of HOSTCOLL=NO, except that control blocks for session collection are initialized from private storage, rather than the common storage area, and VTAM intercepts are not actually set.

Specifies whether NPM enables traps.

Specifies the number of extra trap areas reserved for adding traps. This number does not include the number of traps defined by the TRAP statement.

You can code TR APCNT to any number from 0 to 99. The actual number used by NPM might be rounded up to a greater value to allocate storage the next control block.

Specifies the number of 64-byte entries in the NPM internal trace table. You can code TRC SIZE to any number from 0 to 9999. If you use a value of 0, there is no internal trace table or trap table.
Note: If you code a number larger than 512 on the TRCSIZE parameter, you must also increase the corresponding REGION value in the NPM JCL procedure.

Specifies whether NPM traces the calls to the NPM installation-wide exit routine. This trace is recorded in the NPM internal trace table.

Specifies whether NPM traces VSAM I/O errors. Data from the VSAM request parameter list (RPL) is copied to the trace entry.

Specifies whether NPM traces all control block exchanges through intertask pipe data traffic (FNMPIPE). FNMPIPE activates the following kinds of trace for communication between VTAM statistics processors and the collector probes:

- NPM generates internal trace entries for command exchanges between VTAM statistics and the collector probes through the intertask pipe.
- NPM writes all the control block exchanges between the VTAM statistics and the collector probes. These traces are logged to FNMVLOGx data sets.

Specifies the name of the VTAM intercept module to use in place of the one NPM uses. You can use this module to test changes to the NPM intercept routines. The specified module can be in the STEPLIB or SYS1.LPALIB. If the module is loaded from one of the STEPLIB data sets, the module is placed in the common storage area.
TRAP Statement

The TRAP initialization statement defines one trap to be checked during the processing of the NPM online system.

**Note:** You can choose option 7 on the Problem Determination panel (FNM09DBM) to change the status and parameters of a trap.
The following describes the parameters you can code in the TRAP statement:

- **CALL**: Dispatcher call
- **LOGIC**: Logic error
- **MSG**: Message
- **PSR**: Presentation services request
- **RETURN**: Dispatcher call return
- **RTR**: Router
- **SVC**: Supervisor call
- **SVCR**: Supervisor call return
- **VSAM**: VSAM error
- **VTAM**: VTAM error
Specifies one of the following actions NPM is to take each time the conditions of a trap are met:

**SDUMP**
NPM issues an SDUMP request for a dump of virtual storage. Ensure that the FNMSDUMP data set points to a valid dump data set before choosing this action. Check message FNM991I to verify that the dump was taken successfully.

**SNAP**
NPM issues a SNAP request for the NPM address space. Ensure that the FNMSDUMP data set points to a valid dump data set before choosing this action. Check message FNM991I to verify that the dump was taken successfully.

**SUSPEND**
The update of the NPM internal trace table is suspended. The checking for other traps is continued. See “Displaying Diagnostic Trace Options” on page 96 for more information about how to enable the update of the NPM internal trace table.

**NOOP**
Message FNM990I is issued.

**ABEND**
NPM abends with U3999.

Specifies the module name of the called routine for the CALL, RETURN, SVC, and SVCR type traps. This name must be the full name of the module. If the module name is not specified, it is not checked.

Specifies the logic error code in a logic trap. You can set CODE to any number from 1 to 4095. If CODE=0, all logic errors are processed by this trap.

Specifies the maximum number of times the conditions of the trap can be met before the trap is disabled. Each time conditions are met for a trap, a counter is incremented by one.
When this counter reaches the value of the count parameter, the status of the trap changes to disabled. The trap status is not checked while the trap is inactive or disabled.

You can use option 7 on the Problem Determination panel (FNM09DBM) to change the status of a trap.

You can code COUNT to any number from 1 to 32767.

\[ , \text{FDBK2}=\text{FFFFFFFFFFF} \]
\[ , \text{FDBK2}=\text{sense_codes} \]

Specifies four hexadecimal bytes that are compared to the feedback word 2 field in the VTAM RPL. Each byte in the code field specified as FF is not checked in the RPL. For example, the default of FFFFFFFF results in the feedback word not being checked.

\[ , \text{FDBWD}=\text{FFFFFFFFFFF} \]
\[ , \text{FDBWD}=\text{feedback_code} \]

Specifies four hexadecimal bytes that are compared to the feedback code field in the VTAM or VSAM RPL. Each byte in the code field specified as FF is not checked in the RPL. For example, the default of FFFFFFFF results in the feedback word not being checked.

\[ , \text{ID}=\text{trap_identifier} \]

Specifies an identifier associated with the trap. The identifier is four characters displayed with the trap. If an identifier is not defined, an identifier is generated for the trap.

\[ , \text{ISTATUS}=\text{ACTIVE} \]
\[ , \text{ISTATUS}=\text{INACTIVE} \]

Specifies the initial status of the diagnostic trap. You can modify the status of the trap using the Problem Determination panel (FNM09DBM).

\[ , \text{MODID}=\text{module_name} \]

Specifies the module name of the calling routine for CALL, RETURN, SVC, and SVCR type traps. This name must be the full name of the module. If the module name is not specified, it is not checked.
Specifies the module name called in a router request or the module called for a logic error. If the module name is not specified, it is not checked.

Specifies the message ID for a MSG type trap. The message must contain the full message identifier. The message trap is also checked when the RCVCMD requests are completed. A VTAM message identifier can also be checked.

Attention: Do not specify message identifiers FNM990I, FNM991I, and FNM992I. These messages are used by the diagnostic trap routines and if specified, cause a loop in NPM.

Specifies a message number in a router response. If a message number is not specified, it is not checked.

Specifies the name of a VTAM terminal or the name of a VSAM file. The specified name must be eight hexadecimal characters.

Specifies four-hexadecimal bytes that are compared to the presentation control flags. The PFLGS value indicates the flag bits that are not checked.

The default of 00000000 results in the flags not being checked.

Specifies a hexadecimal mask that NPM logically adds to the presentation control flags before comparing the flags to the PFLGS value. The PMASK value indicates the flag bits to be checked.
The default of 00000000 results in the flags not being checked.

\[ , \text{PNLID} = \text{panel\_identifier} \]

Specifies the last four characters of the panel name in the PSR trap.

\[ , \text{RC} = \text{return\_code} \]

Specifies the return code passed in RETURN and SVCR type traps. This code is in decimal. If the return code is not specified, it is not checked.

\[ , \text{REQ} = \text{FF} \]
\[ , \text{REQ} = \text{request\_code} \]

Specifies the RPL request code in the VTAM and VSAM traps. It is a one-byte hexadecimal code. The default of FF results in the request code not being checked.

\[ , \text{USERID} = \text{user\_identifier} \]

Specifies the operator ID to be checked in presentation services or router traps. The specified identifier must be eight-hexadecimal characters. If not specified, the operator ID is not checked.
Determining the Applicable Parameters for Your Trap

Only certain parameters of the TRAP statement are used for a specific type of trap. Table 15 shows which parameters are used for a particular type of trap.

<table>
<thead>
<tr>
<th>Trap Type</th>
<th>CALL</th>
<th>LOGIC</th>
<th>MSG</th>
<th>PSR</th>
<th>RETURN</th>
<th>RTR</th>
<th>SVC</th>
<th>SVCR</th>
<th>VSAM</th>
<th>VTAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CODE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDBK2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDBWD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODULE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGNR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NODE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PFLGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMASK</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNLID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>USERID</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Activating NPM Internal Trace Classes for Traps

For a particular trap to be active, a corresponding NPM internal trace must be active. Table 16 lists the necessary NPM internal trace class for each trap.

Table 16. NPM Internal Trace Classes for Traps

<table>
<thead>
<tr>
<th>Trap</th>
<th>NPM Internal Trace Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG</td>
<td>MSG (for NPM messages); SPO (for VTAM messages)</td>
</tr>
<tr>
<td>CALL</td>
<td>DSP</td>
</tr>
<tr>
<td>RETURN</td>
<td>RTN</td>
</tr>
<tr>
<td>SVC</td>
<td>SVC</td>
</tr>
<tr>
<td>SVCR</td>
<td>SVC</td>
</tr>
<tr>
<td>VTAM</td>
<td>API</td>
</tr>
<tr>
<td>VSAM</td>
<td>VSM</td>
</tr>
<tr>
<td>PSR</td>
<td>Presentation services requests are traced by using option 4 on the Problem Determination panel (FNM09DBM).</td>
</tr>
<tr>
<td>RTR</td>
<td>RTR</td>
</tr>
<tr>
<td>LOGIC</td>
<td>None (this class is always active)</td>
</tr>
</tbody>
</table>
Using the NPM Internal Trace

The NPM internal trace is a block of main virtual storage reserved for 64-byte trace entries. Each entry represents one event that occurred during NPM online processing.

Finding the Internal Trace Table

To locate the internal trace table, find Register 11 in the dump. Register 11 points to the common address table (CAT) or you can locate the CAT as follows:

1. PSA control block is at location 0
2. PSA at X'10' points to CVT
3. CVT + X'148' points to CVT2
4. CVT2 + X'14' points to MAT
5. MAT + X'B4' points to CAT

Use the worksheet in "Problem Worksheets" on page 155 to record this information.

From the CAT, offset X'20' is the address of the diagnostic trace header (DTH). In the DTH, the offset X'8' is the address where the next trace entry is written. Scan the previous entries to locate the events that happened at the time the error occurred. Figure 24 and the hexadecimal core dump (see Figure 25 on page 128) illustrate the internal trace in storage.

In Figure 25 on page 128, the following pointers locate trace table entries:
- Pointer to oldest trace table entry (DTHDTRC@) at X'00034800'
- Pointer to beginning trace table entry (DTHDTRB@) at X'00030200'
- Pointer to ending trace table entry (DTHDTRE@) at X'00038200'.
### Trace Table Entries

<table>
<thead>
<tr>
<th>Address of DTH</th>
<th>Beginning trace table entry</th>
<th>Ending trace table entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000001</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000010</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000011</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000002</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000003</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000004</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000005</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000006</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
<tr>
<td>00000007</td>
<td>00000000000000000000000000</td>
<td>00000000000000000000000000</td>
</tr>
</tbody>
</table>

**Figure 25. Example of Locating NPM Internal Trace Table**
Understanding Trace Table Entries

Table 17 describes the entry types and classes that can be found in the internal trace.

Most of the trace options are enabled or disabled using the DEBUG initialization statement (see Table 19 on page 110) or by selecting option 2 on the Problem Determination panel (FNM09DBM). The following entries are the exceptions:

- Types LG, KS, KE, and KA are always active if tracing is active.
- Type PS is started from the Problem Determination panel (FNM09DBM), using option 4.
- Type SA is started from the Problem Determination panel (FNM09DBM), using option 5.

The format of the trace types, except for CB and SH, are described in Table 17. Types CB and SH contain NPM internal control blocks. This information can be helpful to the Support Center personnel.

In the table, the event identifier (EID) is a header in the generalized trace facility (GTF) log that identifies the type of record.

### Table 17. Trace Types and Class

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Class</th>
<th>EID (hex)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>APPCIOE</td>
<td>API</td>
<td>E416</td>
<td>VTAM set a non-zero return code for an APPC command. Fields from the VTAM RPL and RPL Extension are placed in the trace entry. See “APPCIOE Trace Entry (AE)” on page 147</td>
</tr>
<tr>
<td>CA</td>
<td>CALL</td>
<td>DSP</td>
<td>E402</td>
<td>The calling routine requests the NPM dispatcher to call the routine and return control. See Table 19 on page 133</td>
</tr>
<tr>
<td>CB</td>
<td>CBS</td>
<td>CBS</td>
<td>E428</td>
<td>Dump of control blocks is requested. The control blocks associated with dispatcher requests are traced to the generalized trace facility (GTF).</td>
</tr>
<tr>
<td>DI</td>
<td>DISP</td>
<td>DSP</td>
<td>E408</td>
<td>The NPM dispatcher starts a new process. See Table 19 on page 133</td>
</tr>
<tr>
<td>EX</td>
<td>EXIT</td>
<td>DSP</td>
<td>E403</td>
<td>A process ends and the dispatcher regains control. See Table 19 on page 133</td>
</tr>
<tr>
<td>IO</td>
<td>IOBUF</td>
<td>IOB</td>
<td>E41D</td>
<td>The first 48 bytes from an I/O buffer are traced when a VTAM (including LU 6.2), VSAM, or BSAM request is finished. See “Input/Output Buffer (IOB) Trace Entry (IO)” on page 143</td>
</tr>
<tr>
<td>IT</td>
<td>VTAMSTAT</td>
<td>VSI</td>
<td>E41B</td>
<td>The internal trace diagnostic record for Intertask Pipe data traffic. Intertask Pipe is used as a communication vehicle between VTAM statistics processors and collector probes. See “Intertask Pipe Data Trace Entry (IT)” on page 144</td>
</tr>
<tr>
<td>IX</td>
<td>USEREXIT</td>
<td>UXT</td>
<td>E41A</td>
<td>The parameter list of an installation-wide exit routine is traced. See “Installation-Wide Exit Trace Entry (IX)” on page 144</td>
</tr>
<tr>
<td>KA</td>
<td>TASKABND</td>
<td></td>
<td>E419</td>
<td>FNMSCEST is entered to process an abend in one of the NPM subtasks. See “TASK Abend Trace Entry (KA)” on page 143</td>
</tr>
<tr>
<td>KE</td>
<td>TASKEND</td>
<td></td>
<td>E418</td>
<td>FNMMMAIN detected the end of a subtask. See “TASK Start/TASK End Trace Entry (KS, KE)” on page 143</td>
</tr>
<tr>
<td>KS</td>
<td>TASKSTR</td>
<td></td>
<td>E417</td>
<td>FNMMMAIN successfully started a subtask. See “TASK Start/TASK End Trace Entry (KS, KE)” on page 143</td>
</tr>
</tbody>
</table>
### Table 17. Trace Types and Class (continued)

<table>
<thead>
<tr>
<th>LG</th>
<th>LOGIC</th>
<th>E400</th>
<th>A logic error is detected. See Table 19 on page 132.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LN</td>
<td>LAN</td>
<td>E415</td>
<td>The NPM LAN Service was requested to trace the LAN request block. See Table 19 on page 132.</td>
</tr>
<tr>
<td>LQ</td>
<td>LREQ</td>
<td>E420</td>
<td>The NPM router is processing a local request. The text of the request buffer is placed in the trace entry. See Table 19 on page 140.</td>
</tr>
<tr>
<td>LR</td>
<td>LRSP</td>
<td>E421</td>
<td>The NPM router completed processing a local request. The text of the response buffer is placed in the trace entry. See Table 19 on page 140.</td>
</tr>
<tr>
<td>MG</td>
<td>MSG</td>
<td>E410</td>
<td>The message and its text are traced in the message trace table. See Table 19 on page 132.</td>
</tr>
<tr>
<td>PO</td>
<td>POST</td>
<td>E409</td>
<td>The NPM dispatcher posted one of the event control blocks (ECB) to start a new process. See Table 19 on page 132.</td>
</tr>
<tr>
<td>PS</td>
<td></td>
<td>E40E</td>
<td>The presentation services trace is turned on for an operator. Each call to FNMPs000 is traced. See Table 19 on page 132.</td>
</tr>
<tr>
<td>QU</td>
<td>QUE</td>
<td>E401</td>
<td>A new process begins when the calling routine requests the NPM dispatcher to queue a new routine. See Table 19 on page 132.</td>
</tr>
<tr>
<td>RC</td>
<td>RCVCMD</td>
<td>E413</td>
<td>The text of an RCVCMD function is traced. See Table 19 on page 132.</td>
</tr>
<tr>
<td>RE</td>
<td>REQUE</td>
<td>E407</td>
<td>The calling routine requests activation of a previously suspended process. See Table 19 on page 132.</td>
</tr>
<tr>
<td>RQ</td>
<td>RREQ</td>
<td>E422</td>
<td>The NPM router is processing a remote request. The text of the request buffer is placed in the trace entry. See Table 19 on page 140.</td>
</tr>
<tr>
<td>RR</td>
<td>RRSP</td>
<td>E423</td>
<td>The NPM router completed processing a remote request. The text of the response buffer is placed in the trace entry. See Table 19 on page 140.</td>
</tr>
<tr>
<td>RT</td>
<td>RTCALL</td>
<td>E40D</td>
<td>The NPM dispatcher returns control to the calling routine in response to a CALL request. See Table 19 on page 132.</td>
</tr>
<tr>
<td>SA</td>
<td>SESSION</td>
<td>E424</td>
<td>The session analysis trace for a logical unit has been requested. The first 48 bytes of the session statistics block are placed in the trace entry. See Table 19 on page 141.</td>
</tr>
<tr>
<td>SC</td>
<td>SENDCMD</td>
<td>E412</td>
<td>The text of a SENDCMD function is traced. See Table 19 on page 132.</td>
</tr>
<tr>
<td>SH</td>
<td></td>
<td>E429</td>
<td>The text of a session data header is traced to GTF from the buffer trace intercept routine. This record type is active if GTF is active.</td>
</tr>
<tr>
<td>SR</td>
<td>SVCRTN</td>
<td>E41F</td>
<td>An initialization routine or an operating system interface routine returned control to its caller. See Table 19 on page 132.</td>
</tr>
<tr>
<td>SV</td>
<td>SVC</td>
<td>E41E</td>
<td>An initialization routine or an operating system interface routine is called. See Table 19 on page 132.</td>
</tr>
</tbody>
</table>
### Table 17. Trace Types and Class (continued)

| TE | TERM | DSP | E406 | The calling routine (FNMTRM00) requests the dispatcher to end after all outstanding processes are complete. See [Table 19 on page 133](#).
|----|------|-----|------|---|
| TI | TIME | DSP | E404 | The calling routine requests the NPM dispatcher to queue the new process after an interval of time expires. See [Table 19 on page 133](#).
| VE | VTAMIO | API | E411 | VTAM set a nonzero return code for a function. Fields from the VTAM RPL (including the feedback and sense codes) are placed in the trace entry. See [“VSAM and VTAM Request Parameter List (RPL) Trace Entry (VE, VS)” on page 133](#).
| VS | VSAM | VSM | E41C | One of the VSAM processors encounters a VSAM error. Fields from the VSAM RPL are placed in the trace entry. See [“VSAM and VTAM Request Parameter List (RPL) Trace Entry (VE, VS)” on page 133](#).
| VX | VTAMEXT | API | E414 | One of the VTAM exit routines is entered. The type of exit routine and the parameter list are placed in the trace entry. If the VTAM exit is associated with a logical unit, the name of the logical unit is also traced (LOGON and LOSTERM). See [Figure 28 on page 136](#).
| WA | SYN | DSP | E405 | The calling routine requests the NPM dispatcher to suspend the current process and queue a new process. The FNMTPX and FNMSVSX services use this function to process requests. See [Table 19 on page 133](#).
Dispatcher Trace Entry (Types QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, SR)

Table 18 shows the record format of dispatcher trace entries (types QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, and SR):

Table 18. Dispatcher Trace Entry (Basic Format of Each Trace Entry)

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0'</td>
<td>X'2'</td>
<td>Type</td>
<td>The type of trace entry (QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, or SR)</td>
</tr>
<tr>
<td>X'2'</td>
<td>X'3'</td>
<td>SEQ</td>
<td>A sequence number to identify the process</td>
</tr>
<tr>
<td>X'3'</td>
<td></td>
<td>PTY</td>
<td>The priority of the process:</td>
</tr>
<tr>
<td>X'4'</td>
<td></td>
<td>Offset</td>
<td>The offset in the calling module</td>
</tr>
<tr>
<td>X'8'</td>
<td>X'10'</td>
<td>CALID</td>
<td>The name of the calling module</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td>MODID</td>
<td>The name of the called module</td>
</tr>
<tr>
<td>X'18'</td>
<td></td>
<td>Register 1</td>
<td>Register 12</td>
</tr>
<tr>
<td>X'20'</td>
<td>X'24'</td>
<td>Register 13</td>
<td>Register 13</td>
</tr>
<tr>
<td>X'24'</td>
<td></td>
<td>Register 14</td>
<td>Register 14</td>
</tr>
<tr>
<td>X'28'</td>
<td></td>
<td>Register 15</td>
<td>Register 15</td>
</tr>
<tr>
<td>X'30'</td>
<td>X'34'</td>
<td>Time Stamp</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Table 19 describes the fields in a dispatcher trace entry.

Table 19. Dispatcher Trace Entry Fields

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, or SR)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>SEQ</td>
<td>A sequence number to identify the process</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>PTY</td>
<td>The priority of the process:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Offset</td>
<td>The offset in the calling module</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Delay</td>
<td>The first word of the store-clock (STCK) time when the process is dispatched</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>CALID</td>
<td>The name of the calling module</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>MODID</td>
<td>The name of the called module</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>Register 12</td>
<td>Register 12</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>Register 13</td>
<td>Register 13</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>Register 14</td>
<td>Register 14</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>Register 15</td>
<td>Register 15</td>
</tr>
<tr>
<td>2C</td>
<td>4</td>
<td>Register 0</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>Register 1</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
Message Trace Entry (MG)

The message entry record is placed into the NPM internal trace table each time a message is issued by the online subtask of NPM. You must set the MSG option. The message is also placed into the message trace table.

Table 20 illustrates the record format of message trace entries (type MG):

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0'</td>
<td>X'4'</td>
<td>Type</td>
<td>The type of trace record (MG)</td>
</tr>
<tr>
<td>X'10'</td>
<td>X'8'</td>
<td>Message Text</td>
<td>The first 48 bytes of message text</td>
</tr>
<tr>
<td>X'20'</td>
<td>X'38'</td>
<td>Time</td>
<td>The time from the time-of-day clock</td>
</tr>
<tr>
<td>X'30'</td>
<td></td>
<td>Message Text</td>
<td>STCK format</td>
</tr>
</tbody>
</table>

Table 21 describes the fields in a message trace entry:

Table 21. Message Trace Entry Fields

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace record (MG)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>(parm list)</td>
<td>The address of the message parameter</td>
</tr>
<tr>
<td>8</td>
<td>48</td>
<td>Message text</td>
<td>The first 48 bytes of message text</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>STCK format</td>
</tr>
</tbody>
</table>
Command Trace Entry (SC, RC)

A command trace entry record is created when the SPO trace option is active. It contains the text of the commands sent to VTAM using the SENDCMD request and the messages received using the RCVCMD request.

The following figure illustrates the record format of command trace entries (types RC and SC).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0'</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (SC or RC)</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td>Command Text (continued)</td>
<td></td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td>Command Text (continued)</td>
<td></td>
</tr>
<tr>
<td>X'30'</td>
<td></td>
<td>Command Text (continued)</td>
<td></td>
</tr>
<tr>
<td>X'38'</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in store-clock (STCK) format</td>
</tr>
</tbody>
</table>

Table 22 describes the fields in a command trace entry.

Table 22. Command Trace Record Fields
VSAM and VTAM Request Parameter List (RPL) Trace Entry (VE, VS)

When a VTAM- or VSAM RPL-based error occurs, a trace entry is built showing the fields from the MVS request parameter list.

The following figure illustrates the record format of VSAM and VTAM RPL trace entries (types VE and VS).

```
<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0'</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (VE or VS)</td>
</tr>
<tr>
<td>X'10'</td>
<td>4</td>
<td>RPLREQ</td>
<td>The first four bytes of the RPL</td>
</tr>
<tr>
<td>X'14'</td>
<td>4</td>
<td>RPLFDBWD</td>
<td>The RPL feedback word</td>
</tr>
<tr>
<td>X'18'</td>
<td>4</td>
<td>RPLFDBK2</td>
<td>The VTAM RPL sense feedback word</td>
</tr>
<tr>
<td>X'1A'</td>
<td>4</td>
<td>A(RPLAREA)</td>
<td>The address of the data buffer</td>
</tr>
<tr>
<td>X'1E'</td>
<td>4</td>
<td>RPLBUFL</td>
<td>The length of the data buffer</td>
</tr>
<tr>
<td>X'22'</td>
<td>4</td>
<td>RPLRLEN</td>
<td>The size of the record area in the data buffer</td>
</tr>
<tr>
<td>X'26'</td>
<td>4</td>
<td>RPLUSFLD</td>
<td>The contents of the VTAM user field</td>
</tr>
<tr>
<td>X'2A'</td>
<td>4</td>
<td>RPLRH3</td>
<td>The contents of the VTAM RPL request header fields</td>
</tr>
<tr>
<td>X'2E'</td>
<td>4</td>
<td>RPOPTC2</td>
<td>The contents of various VTAM RPL option fields</td>
</tr>
<tr>
<td>X'30'</td>
<td>8</td>
<td>Name</td>
<td>The address of the RPL that met the error</td>
</tr>
<tr>
<td>X'38'</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
```

Figure 27. VTAM/VSAM Trace RPL Entry (Type: VE or VS)

Table 23 describes the fields in a VTAM/VSAM trace entry.

**Table 23. VTAM/VSAM Trace Entry Fields**

Offset (hex) | Length (dec) | Field         | Description                                      |
-------------|--------------|---------------|--------------------------------------------------|
0            | 2            | Type          | The type of trace entry (VE or VS)               |
2            | 2            | Reserved      |                                                  |
4            | 4            | RPLREQ        | The first four bytes of the RPL                  |
8            | 4            | RPLFDBWD      | The RPL feedback word                            |
C            | 4            | RPLFDBK2      | The VTAM RPL sense feedback word                 |
10           | 4            | A(RPLAREA)    | The address of the data buffer                    |
14           | 4            | RPLBUFL       | The length of the data buffer                     |
18           | 4            | RPLRLEN       | The size of the record area in the data buffer    |
1C           | 4            | RPLUSFLD      | The contents of the VTAM user field               |
20           | 4            | RPLRH3        | The contents of the VTAM RPL request header fields|
24           | 4            | RPOPTC2       | The contents of various VTAM RPL option fields    |
28           | 4            | Reserved      |                                                  |
2C           | 4            | A(RPL)        | The address of the RPL that met the error         |
30           | 8            | Name          | The name of the terminal or file that had the error|
38           | 8            | Time          | The time from the time-of-day clock in STCK format|
VTAM Exit Trace Entry (VX)

The VTAM exit trace record is created when VTAM drives one of the asynchronous exit routines; LOGON, LOSTERM, TPEND, or SCIP.

The following figure illustrates the record format of VTAM exit trace entries (type VX).

Table 24 describes the fields in a exit trace entry.

### Table 24. VTAM Exit Trace Record Fields

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace record (VX)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>XID</td>
<td>Identifies the kind of VTAM exit routine:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>APND APFC TPEND exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ATTN Attention exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LGON Logon exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LTRM LOSTERM exit routine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SCIP SCIP exit routine</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>Parameter list</td>
<td>The parameter list passed to the exit routine in Register 1</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>Name</td>
<td>The name of the logical unit if passed by VTAM in the parameter list</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Figure 28. VTAM Exit Trace Entry (Type: VX)

Table 24 describes the fields in a exit trace entry.
Logic Trace Entry (LG)

When NPM detects an internal logic error, a trace record is built to contain the information about where the error occurred.

The following figure illustrates the record format of logic trace entries (type LG).

```
<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (LG)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>LGDSP</td>
<td>The displacement into the module that requested a logic error entry be built</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>LGCDE</td>
<td>A code that describes the type of logic error</td>
</tr>
<tr>
<td>C</td>
<td>36</td>
<td>Data</td>
<td>The contents of registers 2 through 10 at the time of the error</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>LGMOD</td>
<td>The name of the module that generated the logic error</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
```

Figure 29. Logic Error Trace Entry (Type: LG)

Table 25 describes the fields in a logic error trace entry.

Table 25. Logic Error Trace Record Fields

Table 25

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (LG)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>LGDSP</td>
<td>The displacement into the module that requested a logic error entry be built</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>LGCDE</td>
<td>A code that describes the type of logic error</td>
</tr>
<tr>
<td>C</td>
<td>36</td>
<td>Data</td>
<td>The contents of registers 2 through 10 at the time of the error</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>LGMOD</td>
<td>The name of the module that generated the logic error</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
LAN Service Trace Entry (LN)

The LAN trace entry is placed in the NPM internal trace table each time NPM makes a request to the NetView program for data from a LAN segment, LAN bridge, or NetWare resources data.

The following figure illustrates the record format of LAN service trace entries (type LN).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace record (LN)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>52</td>
<td>Request</td>
<td>The first 52 bytes of the LAN request block.</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
Presentation Service Trace Entry (PS)

The presentation service trace entry is created when the presentation service trace is requested from the Problem Determination panel (FNM09DBM), using option 4. With this option, you can trace active requests to display screens, the addresses of various fields, and the contents of control flags.

The following figure illustrates the record format of presentation services trace entries (type PS).

Table 27 describes the fields in a presentation service trace entry.

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (PS)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A(OCB)</td>
<td>The address of the operator control block</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>A(NCB)</td>
<td>The address of the node control block</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>A(PCT)</td>
<td>The address of the presentation correlation table</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>A(PCB)</td>
<td>The address of the parameter control block</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>A(PSRB)</td>
<td>The address of the presentation service request block</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>OCBPFLGS</td>
<td>The contents of OCBPFLGS</td>
</tr>
<tr>
<td>1C</td>
<td>4</td>
<td>PCTHFLGS</td>
<td>The contents of PCTHFLGS</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>PSRBWINX</td>
<td>The starting window index</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>Panel ID</td>
<td>The last four characters of the panel name</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>Terminal ID</td>
<td>The terminal ID that requested the display</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>Operator ID</td>
<td>The operator ID that requested the display</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Figure 31. Presentation Service Trace Entry (Type: PS)

Table 27 describes the fields in a presentation service trace entry.
NPM-to-NPM Trace Entry (LQ, LR, RQ, RR)

The NPM-to-NPM trace entry is written when the RTR option is set. Each time you request data from an NPM, a transaction program is executed to collect the data for the display program. There are four types of trace records:

- **LQ** Local request
- **LR** Local response
- **RQ** Remote request
- **RR** Remote response

If the requests collect data from the local NPM, only the LQ or LR entries are recorded.

NPM-to-NPM requests observe the following sequence:
1. **RQ** Request traced on the originating NPM.
2. **LQ** Request traced on the target NPM.
3. **LR** Response traced on the target NPM.
4. **RR** Response traced on the originating NPM.

The following figure illustrates the record format of NPM-to-NPM trace entries (types LQ, LR, RQ, and RR).

Table 28 describes the fields in an NPM-to-NPM trace entry.

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (LQ, LR, RQ, or RR)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A(RRB)</td>
<td>The address of the router request block</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>TXDAT</td>
<td>The text of the request or response buffer</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>TXMDN</td>
<td>The name of the transaction routine</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>TXUID</td>
<td>The operator ID that requested the transaction</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Figure 32. NPM to NPM Trace Entry (Type: LQ, LR, RQ, RR)
Session Trace Entry (SA)

Session trace entries are recorded in the NPM internal trace table if you select option 5 on the Problem Determination panel (FNM09DBM). Information placed in the entry is used to calculate transit times. There is one entry for each RU.

Note: When using option 5, a trace entry is also written to the GTF data set from the VTAM buffer trace intercept.

The following figure illustrates the record format of session trace entries (type SA).

```
<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Value</th>
<th>Field Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Type</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>A(SASSB)</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>SASDH</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>SATH4</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>SAHTM</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>SANTM</td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>SAPLU</td>
</tr>
<tr>
<td>22</td>
<td>28</td>
<td>SASLU</td>
</tr>
<tr>
<td>26</td>
<td>32</td>
<td>Time</td>
</tr>
</tbody>
</table>
```

Figure 33. Session Trace Entry (Type: SA)

Table 29 describes the fields in a session trace entry.

**Table 29. Session Trace Entry Fields**

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (SA)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A(SASSB)</td>
<td>Address of the session statistic block</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>SASDH</td>
<td>Flags from the session data header</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>SATH4</td>
<td>Fields from the transmission header and request unit:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SASNF, Two-byte sequence number field</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SADCF, Two-byte data count field</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SARH, Three-byte request header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SARU, The first five bytes of the request unit</td>
</tr>
<tr>
<td>1C</td>
<td>4</td>
<td>SAFLG</td>
<td>Flags from the session statistic block</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>SAHTM</td>
<td>The host transit time value in hundredths of seconds</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>SANTM</td>
<td>The network transit time value in hundredths of seconds</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>SAPLU</td>
<td>The name of the primary logical unit</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>SASLU</td>
<td>The name of the secondary logical unit</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
**TASK Start/TASK End Trace Entry (KS, KE)**

The following figure illustrates the record format of start and end TASK trace entries (types KS and KE).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'0'</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (KS or KE)</td>
</tr>
<tr>
<td>X'10'</td>
<td></td>
<td>TCA (continued)</td>
<td></td>
</tr>
<tr>
<td>X'20'</td>
<td></td>
<td>TCA (continued)</td>
<td></td>
</tr>
<tr>
<td>X'30'</td>
<td></td>
<td>TCA (continued)</td>
<td></td>
</tr>
<tr>
<td>X'38'</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

**Table 30. TASK Start/TASK End Trace Entry Fields**

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (KS or KE)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A(TCA)</td>
<td>The address of the task control area</td>
</tr>
<tr>
<td>8</td>
<td>48</td>
<td>TCA</td>
<td>Task control area</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
**TASK Abend Trace Entry (KA)**

The following figure illustrates the record format of TASK abend trace entries (type KA).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (KA)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A(SDWA)(^1)</td>
<td>Address of system diagnostic work area</td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td>Register</td>
<td>Contents of registers 0–9</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>PSW</td>
<td>PSW after abend</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

\(^1\) The system diagnostic work area (SDWA) has all registers. See the diagnosis data areas.
Installation-Wide Exit Trace Entry (IX)

The following figure illustrates the record format of installation-wide exit trace entries (type IX). See *NetView Performance Monitor Installation and Customization* for more information about the NPM installation-wide exit.

<table>
<thead>
<tr>
<th>X'0' Type</th>
<th>X'2' Fcn</th>
<th>X'4' Offset</th>
<th>X'8' A(Parm List)</th>
<th>X'C' Return Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'10' Parameter list</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X'20' Parameter list (continued)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X'30' Caller Name</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X'38' Time Stamp (STCK format)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 36. Installation-Wide Exit Trace Entry (Type: IX)*

Table 32 describes the fields in an installation-wide exit trace entry.

**Table 32. Installation-Wide Exit Trace Entry**

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (IX)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Fcn</td>
<td>Function code</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Offset</td>
<td>The offset into the module that called the installation-wide exit</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>A(parm List)</td>
<td>Address of the parameter list</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>Return Code</td>
<td>Return code passed from the exit routine</td>
</tr>
<tr>
<td>10</td>
<td>32</td>
<td>Parameter List</td>
<td>The parameter list passed to the exit routine in Register 1</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>Caller Name</td>
<td>Name of the calling program</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
Input/Output Buffer (IOB) Trace Entry (IO)

The following figure illustrates the record format of input and output buffer (IOB) trace entries (type IO).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (IO)</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>A(NCB)</td>
<td>Address of the node control block</td>
</tr>
<tr>
<td>C</td>
<td>36</td>
<td>I/O Buffer</td>
<td>First 36 bytes in I/O Buffer</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>NCB Name</td>
<td>ID of the NCB</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Table 33 describes the fields in an IOB trace entry.

Figure 37. IOB Trace Entry (Type: IO)

Table 33. IOB Trace Entry
Intertask Pipe Data Trace Entry (IT)

The following figure illustrates the record format of intertask pipe data trace entries (type IT).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (IT)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>FNMWEB</td>
<td>First 28 bytes of the work element block (WEB)</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>FNMVRH</td>
<td>First 24 bytes of data of the VTAM request/response block header for VTAM statistics data</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Figure 38. Intertask Pipe Data Trace Entry (Type: IT)

Table 34 describes the fields in an IOB trace entry.

Table 34. Intertask Pipe Data Trace Entry

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (IT)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>FNMWEB</td>
<td>First 28 bytes of the work element block (WEB)</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>FNMVRH</td>
<td>First 24 bytes of data of the VTAM request/response block header for VTAM statistics data</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
APPCIOE Trace Entry (AE)

The following figure illustrates the record format of APPCIOE data trace entries (type AE).

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Type</td>
</tr>
<tr>
<td>2</td>
<td>RPL6REQ</td>
</tr>
<tr>
<td>4</td>
<td>RPL6QUAL</td>
</tr>
<tr>
<td>5</td>
<td>RPLVTFL1</td>
</tr>
<tr>
<td>7</td>
<td>RPLVTFL2</td>
</tr>
<tr>
<td>10</td>
<td>RPLAREA</td>
</tr>
<tr>
<td>14</td>
<td>RPLBUFL</td>
</tr>
<tr>
<td>16</td>
<td>RPLRLEN</td>
</tr>
<tr>
<td>18</td>
<td>RPL6USR</td>
</tr>
<tr>
<td>1C</td>
<td>RPLOPTC2</td>
</tr>
<tr>
<td>20</td>
<td>RPL6NSO</td>
</tr>
<tr>
<td>24</td>
<td>RPL6NSI</td>
</tr>
<tr>
<td>28</td>
<td>RPL6RC</td>
</tr>
<tr>
<td>2C</td>
<td>RPL6FLGS</td>
</tr>
<tr>
<td>30</td>
<td>Name</td>
</tr>
<tr>
<td>38</td>
<td>Time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offset (dec)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Control value of APPCCMD</td>
</tr>
<tr>
<td>1</td>
<td>Qualify value of APPCCMD</td>
</tr>
<tr>
<td>1</td>
<td>The VTAM RPL flags</td>
</tr>
<tr>
<td>2</td>
<td>The VTAM RPL post/respond flags</td>
</tr>
<tr>
<td>4</td>
<td>The VTAM RPL feedback word</td>
</tr>
<tr>
<td>4</td>
<td>The VTAM RPL sense feedback word</td>
</tr>
<tr>
<td>4</td>
<td>The address of the data buffer</td>
</tr>
<tr>
<td>2</td>
<td>The length of the data buffer (in 2 bytes)</td>
</tr>
<tr>
<td>2</td>
<td>The size of the record area in the data buffer (in 2 bytes)</td>
</tr>
<tr>
<td>4</td>
<td>The VTAM RPL extension user field</td>
</tr>
<tr>
<td>4</td>
<td>The contents of various VTAM RPL option fields</td>
</tr>
<tr>
<td>4</td>
<td>Sense data specified on APPCCMD</td>
</tr>
<tr>
<td>4</td>
<td>Sense data returned by APPCCMD</td>
</tr>
<tr>
<td>4</td>
<td>The VTAM RPL extension return codes</td>
</tr>
<tr>
<td>4</td>
<td>All indicators specific to the VTAM APPCCMD macro</td>
</tr>
<tr>
<td>8</td>
<td>The name of the LU that had the error</td>
</tr>
<tr>
<td>8</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>

Figure 39. APPCIOE Trace Entry (Type: AE)

Table 35. APPCIOE Data Trace Entry

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Type</td>
<td>The type of trace entry (AE)</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>RPL6REQ</td>
<td>Reserved</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>RPL6QUAL</td>
<td>Control value of APPCCMD</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>RPL6QUAL</td>
<td>Qualify value of APPCCMD</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>RPLVTFL1</td>
<td>The VTAM RPL flags</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>RPLVTFL2</td>
<td>The VTAM RPL post/respond flags</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>RPLFDBWD</td>
<td>The VTAM RPL feedback word</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>RPLFDBK2</td>
<td>The VTAM RPL sense feedback word</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>RPLAREA</td>
<td>The address of the data buffer</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>RPLBUFL</td>
<td>The length of the data buffer (in 2 bytes)</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>RPLRLEN</td>
<td>The size of the record area in the data buffer (in 2 bytes)</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>RPL6USR</td>
<td>The VTAM RPL extension user field</td>
</tr>
<tr>
<td>1C</td>
<td>4</td>
<td>RPLOPTC2</td>
<td>The contents of various VTAM RPL option fields</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>RPL6NSO</td>
<td>Sense data specified on APPCCMD</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>RPL6NSI</td>
<td>Sense data returned by APPCCMD</td>
</tr>
<tr>
<td>28</td>
<td>4</td>
<td>RPL6RC</td>
<td>The VTAM RPL extension return codes</td>
</tr>
<tr>
<td>2C</td>
<td>4</td>
<td>RPL6FLGS</td>
<td>All indicators specific to the VTAM APPCCMD macro</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>Name</td>
<td>The name of the LU that had the error</td>
</tr>
<tr>
<td>38</td>
<td>8</td>
<td>Time</td>
<td>The time from the time-of-day clock in STCK format</td>
</tr>
</tbody>
</table>
Using the NPM Message Trace

The NPM message trace is a set of 96-byte trace entries for messages that have been issued by any of the NPM online subtasks. The NPM message trace can be used to reconstruct messages issued to NPM operators or messages issued to the system console operator.

Finding a Message Trace

To locate the last message placed in the message trace table, find register 11. Register 11 points to the common address table (CAT), or you can locate the CAT as follows:

1. PSA control block is at location 0
2. PSA at X'10' points to CVT
3. CVT + X'148' points to CVT2
4. CVT2 + X'14' points to MAT
5. MAT + X'B4' points to CAT

From the CAT, offset X'20' is the address of the diagnostic trace header (DTH). In the DTH, the offset X'14' is the address where the next message entry is written. Scan the previous entries to locate the events that happened at the time the error occurred. Figure 40 and the hexadecimal core dump (see Figure 41 on page 150) illustrate the internal trace in storage.

![Figure 40. Message Trace Diagram](image)

In Figure 41 on page 150 the following pointers help locate trace table entries:

- Pointer to the next message entry (DTHDTMC@) X'0382C0'
- Pointer to the beginning message entry (DTHDTMB@) X'038200'
- Pointer to the ending message entry (DTHDTME@) X'038500'
Figure 41. Example of Locating Message Trace Table Entries

<table>
<thead>
<tr>
<th>Start line</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX Code</td>
</tr>
<tr>
<td>00080000</td>
</tr>
<tr>
<td>00080020</td>
</tr>
<tr>
<td>000300100</td>
</tr>
<tr>
<td>00080040</td>
</tr>
<tr>
<td>00080060</td>
</tr>
<tr>
<td>00080080</td>
</tr>
<tr>
<td>000800A0</td>
</tr>
<tr>
<td>000800C0</td>
</tr>
<tr>
<td>000800E0</td>
</tr>
<tr>
<td>00080100</td>
</tr>
<tr>
<td>00080120</td>
</tr>
<tr>
<td>00080140</td>
</tr>
<tr>
<td>00080160</td>
</tr>
<tr>
<td>00080180</td>
</tr>
<tr>
<td>000801A0</td>
</tr>
<tr>
<td>000801C0</td>
</tr>
<tr>
<td>000801E0</td>
</tr>
<tr>
<td>00080200</td>
</tr>
<tr>
<td>00080220</td>
</tr>
<tr>
<td>00080240</td>
</tr>
<tr>
<td>00080260</td>
</tr>
<tr>
<td>00080280</td>
</tr>
<tr>
<td>000802A0</td>
</tr>
<tr>
<td>000802C0</td>
</tr>
<tr>
<td>000802E0</td>
</tr>
<tr>
<td>00080300</td>
</tr>
<tr>
<td>00080320</td>
</tr>
<tr>
<td>00080340</td>
</tr>
<tr>
<td>00080360</td>
</tr>
<tr>
<td>00080380</td>
</tr>
<tr>
<td>000803A0</td>
</tr>
<tr>
<td>000803C0</td>
</tr>
<tr>
<td>000803E0</td>
</tr>
<tr>
<td>00080400</td>
</tr>
<tr>
<td>00080420</td>
</tr>
<tr>
<td>00080440</td>
</tr>
<tr>
<td>00080460</td>
</tr>
<tr>
<td>00080480</td>
</tr>
<tr>
<td>000804A0</td>
</tr>
<tr>
<td>000804C0</td>
</tr>
<tr>
<td>000804E0</td>
</tr>
<tr>
<td>00080500</td>
</tr>
<tr>
<td>00080520</td>
</tr>
<tr>
<td>00080540</td>
</tr>
<tr>
<td>00080560</td>
</tr>
<tr>
<td>00080580</td>
</tr>
<tr>
<td>000805A0</td>
</tr>
<tr>
<td>000805C0</td>
</tr>
<tr>
<td>000805E0</td>
</tr>
<tr>
<td>00080600</td>
</tr>
<tr>
<td>00080620</td>
</tr>
<tr>
<td>00080640</td>
</tr>
<tr>
<td>00080660</td>
</tr>
<tr>
<td>00080680</td>
</tr>
<tr>
<td>000806A0</td>
</tr>
<tr>
<td>000806C0</td>
</tr>
<tr>
<td>000806E0</td>
</tr>
<tr>
<td>00080700</td>
</tr>
<tr>
<td>00080720</td>
</tr>
<tr>
<td>00080740</td>
</tr>
<tr>
<td>00080760</td>
</tr>
<tr>
<td>00080780</td>
</tr>
<tr>
<td>000807A0</td>
</tr>
<tr>
<td>000807C0</td>
</tr>
<tr>
<td>000807E0</td>
</tr>
<tr>
<td>00080800</td>
</tr>
<tr>
<td>00080820</td>
</tr>
<tr>
<td>00080840</td>
</tr>
<tr>
<td>00080860</td>
</tr>
<tr>
<td>00080880</td>
</tr>
<tr>
<td>000808A0</td>
</tr>
<tr>
<td>000808C0</td>
</tr>
<tr>
<td>000808E0</td>
</tr>
<tr>
<td>00080900</td>
</tr>
<tr>
<td>00080920</td>
</tr>
<tr>
<td>00080940</td>
</tr>
<tr>
<td>00080960</td>
</tr>
<tr>
<td>00080980</td>
</tr>
<tr>
<td>000809A0</td>
</tr>
<tr>
<td>000809C0</td>
</tr>
<tr>
<td>000809E0</td>
</tr>
<tr>
<td>00080A00</td>
</tr>
<tr>
<td>00080A20</td>
</tr>
<tr>
<td>00080A40</td>
</tr>
<tr>
<td>00080A60</td>
</tr>
<tr>
<td>00080A80</td>
</tr>
<tr>
<td>00080AA0</td>
</tr>
<tr>
<td>00080AC0</td>
</tr>
<tr>
<td>00080AE0</td>
</tr>
<tr>
<td>00080B00</td>
</tr>
<tr>
<td>00080B20</td>
</tr>
<tr>
<td>00080B40</td>
</tr>
<tr>
<td>00080B60</td>
</tr>
<tr>
<td>00080B80</td>
</tr>
<tr>
<td>00080BA0</td>
</tr>
<tr>
<td>00080BC0</td>
</tr>
<tr>
<td>00080BE0</td>
</tr>
<tr>
<td>00080C00</td>
</tr>
<tr>
<td>00080C20</td>
</tr>
<tr>
<td>00080C40</td>
</tr>
<tr>
<td>00080C60</td>
</tr>
<tr>
<td>00080C80</td>
</tr>
<tr>
<td>00080CA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
<tr>
<td>00080D60</td>
</tr>
<tr>
<td>00080D80</td>
</tr>
<tr>
<td>00080DA0</td>
</tr>
<tr>
<td>00080CE0</td>
</tr>
<tr>
<td>00080D00</td>
</tr>
<tr>
<td>00080D20</td>
</tr>
<tr>
<td>00080D40</td>
</tr>
</tbody>
</table>
The following figure shows the format of a message trace entry.

![Message Trace Entry](image)

**Table 36. Message Trace Entry Fields**

<table>
<thead>
<tr>
<th>Offset (hex)</th>
<th>Length (dec)</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>Date</td>
<td>Date message was written (0YYMMDD)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Time</td>
<td>Time message was written (0HHMMSSF)</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>User Identification</td>
<td>User identification associated with the message</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>A(OCB)</td>
<td>Address of the operator control block</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>A(MWA)</td>
<td>Address of the message work area</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>Flags</td>
<td>Routing and option flags from the message work area</td>
</tr>
<tr>
<td>1C</td>
<td>68</td>
<td>Message Text</td>
<td>The first 68 bytes of message text</td>
</tr>
</tbody>
</table>

*Figure 42. Message Trace Entry*

*Table 36* describes the fields of a message trace entry.
Part III. Appendixes
Problem Worksheets

The following worksheets are for your convenience when you are collecting information about problems. You can copy these worksheets for repeated use.

- NPM Internal Trace
- NPM Major Control Blocks
- NPM Abend Dump
- NPM Save Area Layout
- Problem Determination Information Control Block
- RTM2 Work Area Layout
- VM GCS Abends and Program Checks
- VM GCS State Blocks
NPM Internal Trace

Use this worksheet to help you locate the NPM internal trace table. See “Using the NPM Internal Trace” on page 127 for information.

CAT - ____________________ (ADDRESS OF CAT (REGISTER 11 IN DUMP))

DTH - ____________________ (ADDRESS OF DIAGNOSTIC TRACE HEADER)

(CAT + X'20')

(DTH + X'08') (ADDRESS OF NEXT TRACE ENTRY TO BE WRITTEN TO)

(DTH + X'0C') (ADDRESS OF FIRST TRACE ENTRY TO BE WRITTEN TO)

(DTH + X'10') (ADDRESS OF LAST TRACE ENTRY TO BE WRITTEN TO)
NPM Major Control Blocks

Use this worksheet to help you locate the PDI control block. See "Using PDI, FNMILOG, SYSPRINT, and PANELID" on page 73 for information.

<table>
<thead>
<tr>
<th>MAJOR CONTROL BLOCKS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSA</td>
<td>(LOCATION X'0')</td>
</tr>
<tr>
<td>CVT</td>
<td>(PSA + X'10')</td>
</tr>
<tr>
<td>CVT2</td>
<td>(CVT + X'14')</td>
</tr>
<tr>
<td>MAT</td>
<td>(CVT2 + X'14')</td>
</tr>
<tr>
<td>CAT</td>
<td>(MAT + X'B4')</td>
</tr>
</tbody>
</table>
NPM Abend Dump

Use this worksheet to collect information about NPM abends when you are documenting ABEND problems. See “Documenting Problems” on page 35 for information.

COMPLETION CODE: ________
ADDRESS OF RTM2WA: ________
ABEND PROGRAM NAME: ________
ABEND PROGRAM ADDRESS: ________

EC PSW AT TIME OF ERROR: ________ ________ ________ ________

REGISTERS AT TIME OF ERROR

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG 15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### NPM Save Area Layout

Use this worksheet to find the module flow and to see the general purpose registers. One save area would show the content of the general purpose registers.

<table>
<thead>
<tr>
<th>LOC.</th>
<th>DESC.</th>
<th>CONTENTS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'04'</td>
<td>BACK PTR.</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'08'</td>
<td>FWRD PTR.</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'0C'</td>
<td>REG E</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'10'</td>
<td>REG F</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'14'</td>
<td>REG 0</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'18'</td>
<td>REG 1</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'1C'</td>
<td>REG 2</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'20'</td>
<td>REG 3</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'24'</td>
<td>REG 4</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'28'</td>
<td>REG 5</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'2C'</td>
<td>REG 6</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'30'</td>
<td>REG 7</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'34'</td>
<td>REG 8</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'38'</td>
<td>REG 9</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'3C'</td>
<td>REG 10</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'40'</td>
<td>REG 11</td>
<td>_________</td>
<td>______________</td>
</tr>
<tr>
<td>X'44'</td>
<td>REG 12</td>
<td>_________</td>
<td>______________</td>
</tr>
</tbody>
</table>
**Problem Determination Information Control Block**

Use this worksheet to help you locate the PDI control block. See "Using PDI, FNMILOG, SYSPRINT, and PANELID" on page 73 for information.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00'</td>
<td>FNMCAT AT X'24' (POINTER TO NEXT FNMPDI)</td>
<td></td>
</tr>
<tr>
<td>X'00'</td>
<td>D7C4C940 (CONTROL BLOCK IDENTIFIER 'PDI')</td>
<td></td>
</tr>
<tr>
<td>X'04'</td>
<td>ADDRESS OF NEXT PDI</td>
<td></td>
</tr>
<tr>
<td>X'08'</td>
<td>TIME OF ABEND</td>
<td></td>
</tr>
<tr>
<td>X'10'</td>
<td>ABEND CODE</td>
<td></td>
</tr>
<tr>
<td>X'14'</td>
<td>ADDRESS OF SDWA</td>
<td></td>
</tr>
<tr>
<td>X'18'</td>
<td>ADDRESS OF TCA</td>
<td></td>
</tr>
<tr>
<td>X'1C'</td>
<td>ADDRESS OF EPL</td>
<td></td>
</tr>
<tr>
<td>X'20'</td>
<td>LOAD MODULE NAME</td>
<td></td>
</tr>
<tr>
<td>X'28'</td>
<td>CSECT NAME</td>
<td></td>
</tr>
<tr>
<td>X'30'</td>
<td>CSECT ASMB. DATE</td>
<td></td>
</tr>
<tr>
<td>X'38'</td>
<td>PTF/FMID VALUE</td>
<td></td>
</tr>
<tr>
<td>X'40'</td>
<td>REGISTERS AT ABEND</td>
<td>REG 0 REG 1 REG 2 REG 3</td>
</tr>
<tr>
<td>X'50'</td>
<td>REG 0 REG 1 REG 2 REG 3</td>
<td>REG 4 REG 5 REG 6 REG 7</td>
</tr>
<tr>
<td>X'60'</td>
<td>REG 4 REG 5 REG 6 REG 7</td>
<td>REG 8 REG 9 REG 10 REG 11</td>
</tr>
<tr>
<td>X'70'</td>
<td>REG 8 REG 9 REG 10 REG 11</td>
<td>REG 12 REG 13 REG 14 REG 15</td>
</tr>
<tr>
<td>X'80'</td>
<td>EC PSW AT ABEND</td>
<td></td>
</tr>
<tr>
<td>X'88'</td>
<td>ILC CC EXCEPTION ADDRESS</td>
<td></td>
</tr>
<tr>
<td>X'90'</td>
<td>OFFSET IN MODULE</td>
<td></td>
</tr>
</tbody>
</table>
RTM2 Work Area Layout

Use this worksheet when you cannot find the PDI control block.

**PSW AT TIME OF ERROR (OFFSET X'7C')**: 

**ILC AND PIC AT TIME OF ERROR (OFFSET X'84')**: 

**REGISTERS AT TIME OF ERROR (OFFSET X'3C')**

<table>
<thead>
<tr>
<th>REGISTER</th>
<th>CONTENTS</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'3C'</td>
<td>REG 0</td>
<td></td>
</tr>
<tr>
<td>X'40'</td>
<td>REG 1</td>
<td></td>
</tr>
<tr>
<td>X'44'</td>
<td>REG 2</td>
<td></td>
</tr>
<tr>
<td>X'48'</td>
<td>REG 3</td>
<td></td>
</tr>
<tr>
<td>X'4C'</td>
<td>REG 4</td>
<td></td>
</tr>
<tr>
<td>X'50'</td>
<td>REG 5</td>
<td></td>
</tr>
<tr>
<td>X'54'</td>
<td>REG 6</td>
<td></td>
</tr>
<tr>
<td>X'58'</td>
<td>REG 7</td>
<td></td>
</tr>
<tr>
<td>X'5C'</td>
<td>REG 8</td>
<td></td>
</tr>
<tr>
<td>X'60'</td>
<td>REG 9</td>
<td></td>
</tr>
<tr>
<td>X'64'</td>
<td>REG 10</td>
<td></td>
</tr>
<tr>
<td>X'68'</td>
<td>REG 11</td>
<td></td>
</tr>
<tr>
<td>X'6C'</td>
<td>REG 12</td>
<td></td>
</tr>
<tr>
<td>X'70'</td>
<td>REG 13</td>
<td></td>
</tr>
<tr>
<td>X'74'</td>
<td>REG 14</td>
<td></td>
</tr>
<tr>
<td>X'78'</td>
<td>REG 15</td>
<td></td>
</tr>
</tbody>
</table>
VM GCS Abends and Program Checks

Use this worksheet to collect information about NPM abends when you are documenting ABEND problems. See "Documenting Problems" on page 35 for information.

VM GCS ABENDS

X'298' : ____________ (PTR TO ABEND WORK AREA)

WORK AREA

X'00' ________ ________ ________ ________
REG 0 REG 1 REG 2 REG 3
X'10' ________ ________ ________ ________
REG 4 REG 5 REG 6 REG 7
X'20' ________ ________ ________ ________
REG 8 REG 9 REG 10 REG 11
X'30' ________ ________ ________ ________
REG 12 REG 13 REG 14 REG 15
X'40' ________ ________ ________ ________
PSW

VM GCS PROGRAM CHECKS

X'A68' : ____________ (PTR TO PROGRAM CHECK WORK AREA)

WORK AREA

X'00' __________________________________________
PSW
X'08' ________ ________ ________ ________
REG 0 REG 1 REG 2 REG 3
X'18' ________ ________ ________ ________
REG 4 REG 5 REG 6 REG 7
X'28' ________ ________ ________ ________
REG 8 REG 9 REG 10 REG 11
X'38' ________ ________ ________ ________
REG 12 REG 13 REG 14 REG 15
VM GCS State Blocks

Use this worksheet to collect information about NPM abends when you are documenting ABEND problems. See "Documenting Problems" on page 35 for information.

VM GCS ABENDS

X’210’ : __________________________________________________ (MACHINE ID)
X’212’ : __________________________________________________ (CURRENT TASK)
X’214’ : __________________________________________________ (PTR TO CURRENT TASK BLOCK)

TASK BLOCK

X’10’ : __________________________________________________ (ACTIVE STATE BLOCK ADDRESS)
X’18’ : __________________________________________________ (PSW)
X’C4’ : __________________________________________________ (ABEND CODE)

STATE BLOCK

X’00’ : __________________________________________________ (PROGRAM NAME)
X’08’ : __________________________________________________ (PSW OF CALLER)
X’10’ : __________________________________________________ (ADDRESS OF NEXT STATE BLOCK ON CHAIN)
X’14’ : __________________________________________________ (PREVIOUS STATE BLOCK ON CHAIN)
X’18’ : __________________________________________________ (ADDRESS OF TASK BLOCK FOR STATE BLOCK)
X’20’ : __________________________________________________ (ENTRY POINT OF PROGRAM OR SVC)
X’24’ : __________________________________________________ (FLAGS 80 LINK, 40 SVC, 20 AEB)
X’26’ : __________________________________________________ (WAIT COUNT)

X’30’ REGISTERS:

<table>
<thead>
<tr>
<th>REG 0</th>
<th>REG 1</th>
<th>REG 2</th>
<th>REG 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG 4</td>
<td>REG 5</td>
<td>REG 6</td>
<td>REG 7</td>
</tr>
<tr>
<td>REG 8</td>
<td>REG 9</td>
<td>REG 10</td>
<td>REG 11</td>
</tr>
<tr>
<td>REG 12</td>
<td>REG 13</td>
<td>REG 14</td>
<td>REG 15</td>
</tr>
</tbody>
</table>
Checkpoint Processing for NetWare Resources Collections

When you start a NetWare resources collection, the NPM NetWare Agent sends collection data to NPM on an unsolicited basis. The NPM NetWare Agent collects and forwards the data by interval instead of NPM polling for NetWare data. The data continues to be sent until the collection is completed or the collection is stopped by the user issuing an LWGSTOP command.

If NPM is terminated without all active collections being stopped or if NPM ABENDS, the NPM NetWare Agent continues to collect and forward data. NPM checkpoint processing takes a snapshot of active NetWare collections when NPM terminates or ABENDS. NPM begins the checkpointing process by writing a record in file FNMLCOLL for each active NetWare collection. When you start NPM again, NPM uses the information in file FNMLCOLL to try to restore the NetWare collections to the status they were in before termination. See NetView Performance Monitor Installation and Customization for more information on starting NPM.

Defining NetWare Resources

When you start NPM after an ABEND or termination, NPM uses the information held in the FNMSRT file that defines NetWare resources. NPM has no knowledge of any NetWare resources defined by an LWGRES command before NPM ABENDED or was terminated. To ensure that you can collect data for these resources, either include these resources in the FNMSRT file before restarting NPM or use LWGRES commands to define them immediately after restarting NPM.
Restarting Active NetWare Collections

NPM uses the data from the checkpoint file FNMLCOLL to restart as many of the collections that were active at termination as possible. As part of the checkpoint processing, NPM checks each record read from FNMLCOLL as follows:

- **Logging** collections with STOP time not expired and resource definitions found will be restarted.

- **Logging** collections with STOP time not expired, no resource definitions found, and DYNAMIC=YES will have an LWGCOLL OPTION=START command put on the NPM dynamic queue. These collections will only be restarted when the resource is redefined to NPM through LWGRES commands or EXECs.

- **Logging** collections with STOP time expired will be stopped. The NPM NetWare Agent will be sent a command to stop the collection.

- **Logging** collections where no resource definition is found and DYNAMIC=NO will be stopped. The NPM NetWare Agent will be sent a command to stop the collection.

- **Data forwarding** collections will be stopped. The NPM NetWare Agent will be sent a command to stop the collection.

Any data sent to NPM by the NPM NetWare Agent while NPM is down is discarded.

Understanding Checkpoint Messages

As part of the checkpoint and restart processing, NPM issues messages. The most likely reasons for these messages are as follows:

- **FNM007E**
  - NPM is trying to restart a collection for a NetWare resource but the resource is not defined in the FNMSRT file.

- **FNM277W**
  - NPM is trying to restart a collection for a NetWare resource but the command failed.

- **FNM1170I**
  - NPM is terminating and has found active NetWare collections.

- **FNM1177I**
  - NPM has been restarted and is carrying out checkpoint processing. You might obtain this message more than once until NPM completes checkpoint processing. No action is required as this message is for information only.

- **FNM1179I**
  - NPM has been restarted and during checkpoint processing discovered that the sequence number for a NetWare collection just received from the NPM NetWare Agent was not the number that NPM was expecting. Data sent by the NPM NetWare Agent has been lost while NPM was down.
Glossary

A

ACB
(1) In VTAM programs, access method control block. (2) In NCP, adapter control block. (3) Application control block.

access method control block (ACB)
(1) A control block that links an application program to VSAM or VTAM. (2) In VTAM statistics collection, the control block passed to VTAM when an application is establishing itself as able to send and receive data.

access method services (AMS)
The facility used to define and reproduce VSAM key-sequenced data sets (KSDS).

active collection
In NPM, the process of collecting data at the current time about a resource.

ACTPU
Activate physical unit. In SNA, a command used to start a session on a physical unit.

adapter control block (ACB)
In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

address space
A set of addresses used to uniquely identify network accessible units, sessions, adjacent link stations, and links in a node for each network in which the node participates. A type 2.1 node has one address space for intranode routing and one for each transmission group on which it can send message units.

ADP
Automatic data processing.

advanced peer-to-peer networking (APPN) end node
A type 2.1 node that provides full SNA end-user services and supports sessions between its local control point (CP) and the CP in an adjacent network node, to dynamically register its resources with the adjacent CP (its network node server), to send and receive directory search requests, and to obtain management services; it can also attach to a subarea network as a peripheral node.

advanced peer-to-peer networking (APPN) network
An SNA network consisting of APPN network nodes, APPN end nodes, and, optionally, LEN end nodes.

advanced peer-to-peer networking (APPN) network node
A type 2.1 node that, besides offering full SNA end-user services, provides intermediate routing services within a type 2.1 network and network services to its local LUs and attached type 2.1 end nodes in its domain; it can also attach to a subarea network as a peripheral node.

advanced peer-to-peer networking (APPN) node
An APPN network node or an APPN end node.

advanced peer-to-peer networking (APPN)
An extension to SNA featuring (a) greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure, (b) dynamic exchange of network topology information to foster ease of connection and reconfiguration, adaptive route selections, and simplified network definition, and (c) automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for end-user services to network control; APPN also uses LU 6.2 protocols on its own control point sessions that provide the network control.

advanced program-to-program communication (APPC)
An implementation of the SNA/SDLC LU6.2 protocol that allows interconnected systems to communicate and share the processing of programs.
agent
Reports to the managing process, such as a network control station, on the status of managed network elements and performs actions on these elements as directed by the managing process. Examples of network elements are devices such as hosts, routers, and terminal servers.

alert
(1) A message sent to a management services focal point in a network to identify a problem or an impending problem. (2) In the NetView program, a high priority event that warrants immediate attention. A database record is generated for certain event types that are defined by user-constructed filters.

alert condition
A problem or impending problem for which some or all of problem determination, diagnosis, or resolution is expected at a collection point or a system problem determination focal point.

alias address
An address used by a gateway NCP and a gateway system services control point (SSCP) in one network to represent a logical unit (LU) or SSCP in another network.

alias name
A name unique within one of two or more interconnected networks that is assigned by a gateway SSCP and is used in that subnetwork to represent an NAU that resides in another subnetwork. Alias names must be predefined within a gateway SSCP; if an alias is not provided, it is assumed to be the same as the real name.

APF
(1) Authorized program facility. (2) Application processing function.

APPC
See Advanced program-to-program communication.

APPL
Application program.

application identifier (APPLID)
The name of an application as defined to VTAM.

APPLID
Application ID.

APPN
Advanced peer-to-peer networking.

APPN network
See advanced peer-to-peer networking network.

APPN network node
See advanced peer-to-peer networking (APPN) network node.

APPN node
See advanced peer-to-peer networking (APPN) node.

B

base collection interval
The primary unit, in seconds or minutes, which NPM uses to calculate the span of time in which data is collected.

bracket protocol
In SNA, a data flow control protocol in which exchanges between two session partners are achieved through the use of brackets, with one partner designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules.
buffer pool
(1) An area of storage in which all buffers of a program are kept. (2) In ACF/TCAM, a group of buffers having the same size. A buffer pool is established at initialization time in the message control program. The buffers are built in extents chained together.

cache
(1) An optional part of the directory database in network nodes where frequently used directory information may be stored to speed directory searches. (2) To place, hide, or store in a cache.

collection interval
In NPM, a user-specified value that controls the time at which performance data is transmitted.

command list (CLIST)
A list of commands and statements designed to perform a specific function for the user.

command procedure (PROC)
In the NetView program, either a command list or a command processor.

Communications storage manager (CMS).
A component of VTAM that enables host applications to share data with VTAM and other CSM users without having to physically copy the data.

CSM
Communications storage manager.

dsn
Data set name.

dump
(1) To record, at a particular instant, the contents of all or part of one storage device in another storage device. Dumping is usually for the purpose of debugging. (2) Data that has been dumped. (3) To copy data in a readable format from main or auxiliary storage onto an external medium such as tape, diskette, or printer. (4) To copy the contents of all or part of virtual storage for the purpose of collecting error information.

EXEC
(1) In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches. (2) In NPM, a user-written command file that contains NPM commands that can be run at NPM startup, online, or through the console. Also called NPM EXEC.

fast path
In SAA Basic Common User Access® architecture, a method of doing something more directly and quickly than the usual way; for example, pressing a function key is faster than typing a command.

flow control
In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also adaptive session-level pacing, pacing, and session-level pacing.
frame
(1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

FSB
File service buffer.

FST
File service task.

G

gateway
A functional unit that interconnects two computer networks with different network architectures.

generation
The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

GTF
Generalized trace facility.

H

half-duplex (HD, HDX)
In data communication, pertaining to transmission in only one direction at a time. Contrast with duplex. See also half-duplex operation and half-duplex transmission.

historical data
Indicates that the data displayed in the HISTORICAL DataView window was requested within a specified time span in the past from the NPM host VSAM data files. The performance data collection is not currently active. No data forwarding is required.

I

IPX
Internetwork package exchange

ISTATUS
In VTAM and NCP, a definition specification method for indicating the initial status of resources. See also indirect activation.

J

JCL
Job control language.

JES
Job entry subsystem (MVS).

job control language (JCL)
A control language used to identify a job to an operating system and to describe the job’s requirements.

K

keyword
(1) A name or symbol that identifies a parameter. (2) The part of a command operand that consists of a specific character string (such as DSNAME=). See also definition statement and keyword operand. Contrast with positional operand.
KSDS
Key-sequenced data set.

L

line
The portion of a data circuit, external to data circuit-terminating equipment (DCE), that connects the DCE to a data switching exchange (DSE), that connects a DCE to one or more other DCEs, or that connects a DSE to another DSE. Synonymous with channel and circuit. See also telecommunication line.

line speed
The number of binary digits that can be sent over a telecommunication line in one second, expressed in bits per second (bps).

link
In SNA, the combination of the link connection and the link stations joining network nodes, for example a System/370™ channel and its associated protocols or a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with data link.

load module
All or part of a computer program in a form suitable for loading into main storage for execution. A load module is usually the output of a linkage editor.

logical unit (LU)
(1) A type of network accessible unit that enables end users to gain access to network resources and communicate with each other. (2) In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions and one with another LU and may be capable of supporting many sessions with other LUs. See also secondary logical unit (SLU).

logical unit (LU) 6.2
A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

LPA
Link pack area.

LRECL
Logical record length.

LU
Logical unit.

LU group
In the NetView Performance Monitor (NPM), a file containing a list of related or unrelated logical units. The LU group is used to help simplify data collection and analysis.

LU type
The classification of an LU in terms of the specific subset of SNA protocols and options it supports for a given session, namely:
- The mandatory and optional values allowed in the session activation request
- The usage of data stream controls, function management headers (FMHs), request unit parameters, and sense data values
- Presentation services protocols such as those associated with FMH usage
LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.
LU type 6.2 (LU 6.2)
A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (a) a peer relationship between session partners, (b) efficient utilization of a session for multiple transactions, (c) comprehensive end-to-end error processing, and (d) a generic application program interface consisting of structured verbs that are mapped into a product implementation.

LU 6.2
Logical unit 6.2.

LU 6.2 session
A session that is initiated by VTAM on behalf of a logical unit (LU) 6.2 application program, or a session initiated by a remote LU in which the application program specifies that VTAM is to control the session by using the APPCCMD macroinstruction.

LU 6.2 verb
A syntactical unit in the LU 6.2 application program interface representing an operation.

M

MIB
Management information base.

module
A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

monitor
In the IBM Token-Ring Network, the function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

monitor event message
A message created in response to a value that has exceeded a user-defined threshold.

monitor resolution message
A message created when a value that has exceeded a user-defined threshold falls into normal range.

MVS
(1) Multiple virtual storage. Implies MVS/370, the MVS/XA™ product, and the MVS/ESA™ product. (2) Multiple Virtual Storage, consisting of MVS/System Product Version 1 and the MVS/370 Data Facility Product operating on a System/370 processor. See also MVS/XA product.

MVS/ESA product
Multiple Virtual Storage/Enterprise Systems Architecture.

MVS/XA product
Multiple Virtual Storage/Extended Architecture product, consisting of MVS/System Product Version 2 and the MVS/XA Data Facility Product, operating on a System/370 processor in the System/370 extended architecture mode. See also MVS.

MWA
Message work area.

N

NCP
Network Control Program.

NCP/token ring interconnect (NTRI)
An NCP function that allows a communication controller to attach to the IBM Token-Ring Network and that provides both subarea and peripheral node DLC services in the SNA network.
NCP/EP definition facility (NDF)
A program that is part of System Support Programs (SSP) and is used to generate a partitioned emulation program (PEP) load module or a load module for a Network Control Program (NCP) or for an Emulation Program (EP).

NCP/Token Ring interconnect (NTRI)
An NCP function that allows a communication controller to attach to the IBM Token-Ring Network and that provides both subarea and peripheral node data link control (DLC) services in the SNA network.

NEO
Network extension option.

NEOPU
Network extension option physical unit.

NetView Performance Monitor (NPM)
An IBM-licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

NetView Performance Monitor Desk/2
NPM Desk/2 is the interface between an OS/2 workstation and an NPM host. NPM Desk/2 provides functions to collect and display performance data using Configuration and DataView windows.

NetView Program
An IBM-licensed program used to monitor a network, manage it, and diagnose its problems.

Network Control Program (NCP)
An IBM-licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

network gateway accounting (NGA)
The NetView Performance Monitor (NPM) subsystem that receives traffic information from the gateway NCP for sessions that flow throughout a network.

network node (NN)
Synonym for APPN network node. A node that offers full SNA end-user services and that can provide the following to its local LUs and client end nodes:
- Distributed directory services
- Intermediate routing services within an APPN network
- Network services
The APPN network node cooperates with other network nodes to maintain a network topology database, which is used to select optimal routes for LU-LU sessions based on requested classes of service. An APPN network node can also attach to a subarea network as a peripheral node or to other end nodes.

network session accounting (NSA)
The NetView Performance Monitor (NPM) subsystem that receives session accounting information from the NCP for sessions that flow throughout a network.

network transit time
In the NetView Performance Monitor (NPM), the average time (in seconds) that all transactions spend in the network. See operator transit time and host transit time.

NGA
Network gateway accounting.

node type
A designation of a node according to the protocols it supports and the network accessible units that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Within a subarea network, type 1, type 2.0, and type 2.1 nodes are peripheral nodes, while type 4 and type 5 nodes are subarea nodes.

non-BC
Non-broadcast.
NPALU
Network performance analysis logical unit.

NPM
NetView Performance Monitor.

NPM Desk/2
NetView Performance Monitor Desk/2.

NPM log
In NPM, a data set in which NPM data can be stored.

NPM security profile
In NPM, a data set that can be created to restrict operator access to data and functions.

octet
(1) A byte that consists of 8 bits. (2) A byte composed of 8 binary elements. Synonymous with eight-bit byte.

ODAI
Origin destination assignment indicator.

ODLC
Outboard data link control.

operand
(1) An entity on which an operation is performed. (2) That which is operated upon. An operand is usually identified by an address part of an instruction. (3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor. (4) An expression to whose value an operator is applied. See also definition statement, keyword, keyword parameter, and parameter.

operator transit time
In the NetView Performance Monitor (NPM), the sum of host and network transit times. See host transit time and network transit time.

OS/390 operating system
An IBM licensed program that not only includes and integrates functions previously provided by many IBM software products (including the MVS operating system) but also (a) is an open, secure operating system for the IBM S/390 family of enterprise servers, (b) complies with industry standards, (c) is Year 2000 ready and enabled for network computing and e-business, and (d) supports technology advances in networking server capability, parallel processing, and object-oriented programming.

partitioned data set (PDS)
A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

path information unit (PIU)
A message unit consisting of a transmission header (TH) alone, or a TH followed by a basic information unit (BIU) or a BIU segment. See also transmission header.

PDS
Partitioned data set.

PDU
Protocol data unit.
pending active session
In VTAM, the state of an LU-LU session recorded by the system services control point (SSCP) when it finds both logical units (LUs) available and has sent a CINIT request to the primary logical unit (PLU) of the requested session.

PEP
Partitioned emulation programming.

physical unit (PU)
The component that manages and monitors the resources (such as attached links and adjacent link stations) associated with a node, as requested by an SSCP via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. This term applies to type 2.0, type 4, and type 5 nodes only. See also peripheral PU and subarea PU.

PIU
Path information unit.

PLU
Primary logical unit.

poll
(1) To determine whether any remote device on a telecommunication line is ready to transmit data. (2) To execute a polling sequence. See also positive poll and negative poll.

polling
(1) On a multipoint connection or a point-to-point connection, the process whereby data stations are invited, one at a time, to transmit. (2) Interrogation of devices for such purposes as to avoid contention, to determine operational status, or to determine readiness to send or receive data.

positive poll
A positive response to polling in a binary synchronous terminal.

positive response
In SNA, a response indicating that a request was received and processed. Contrast with negative response.

protocol data unit (PDU)
A unit of data in a network. For SDLC protocols, a PDU is a path information unit. For token-ring protocols, a PDU is an I-frame.

PST
(1) Process scheduling table. (2) Program scheduling table.

PU
Physical unit.

PU type
(1) Deprecated term for node type. (2) The type of physical unit in a node.

Q

QCB
Queue control block.

R

RACF
Resource Access Control Facility.

real-time data
Indicates an active data forwarding collection. The performance information is current, and the data is displayed online in a REAL-TIME DataView window. Real-time performance data is collected at regular intervals and the DataView is refreshed automatically, as soon as the data arrives.
refresh interval
The interval at which NPM displays the most recent data collected from VTAM. See sampling interval.

resolution message
A message created when a value that has exceeded a user-defined threshold falls into normal range.

Resource Access Control Facility (RACF)
An IBM-licensed program that provides for access control by identifying and verifying the users of the system, by authorizing access to protected resources, by logging the detected unauthorized attempts to enter the system, and by logging the detected accesses to protected resources.

resource table
In ACF/TCAM extended networking, a main-storage table that associates each resource identifier with an external logical unit (LU) or application program.

resource types
In the NetView program, a concept to describe the organization of panels. Resource types are defined as central processing unit, channel, control unit, and I/O device for one category; and communication controller, adapter, link, cluster controller, and terminal for another category. Resource types are combined with data types and display types to describe display organization. See also data types and display types.

response time monitor (RTM)
A feature available with certain hardware devices to allow measurement of response times, which may be collected and displayed by the NetView program.

RNR
Receive not ready.

router
(1) A computer that determines the path of network traffic flow. The path selection is made from several paths based on information obtained from specific protocols, algorithms that attempt to identify the shortest or best path, and other criteria such as metrics or protocol-specific destination addresses. (2) An attaching device that connects two LAN segments, which use similar or different architectures, at the reference model network layer. (3) In OSI terminology, a function that determines a path by which an entity can be reached.

routing
(1) The process of determining the path to be used for transmission of a message over a network. (2) The assignment of the path by which a message will reach its destination.

sampling interval
The interval at which NPM updates its own counters. This updated data is not displayed until the next refresh interval. If a sampling interval is set higher than a refresh interval, the sampling interval is automatically reset to the refresh interval. See refresh interval.

SDLC
Synchronous Data Link Control.

secondary logical unit (SLU)
In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with primary logical unit (PLU).

segment
(1) Synonym for BIU segment. (2) In the IBM Token-Ring Network, a section of cable between components or devices. A segment may consist of a single patch cable, several patch cables that are connected, or a combination of building cable and patch cables that are connected. (3) See link connection segment.

service point (SP)
An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not
under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

**session collection**
The NPM subsystem that collects, monitors, and displays data collected in the host for analysis.

**session data**
Data about sessions collected by NPM.

**session monitor**
The component of the NetView program that collects and correlates session-related data and provides online access to this information.

**session partner**
In SNA, one of the two network accessible units (NAUs) having an active session.

**session trace**
In the NetView program, the function that collects session trace data for sessions involving specified resource types or involving a specific resource.

**Simple Network Management Program (SNMP)**
In the Internet suite of protocols, a network management protocol that is used to monitor routers and attached networks. SNMP is an application layer protocol. Information on devices managed is defined and stored in the application’s Management Information Base (MIB).

**SIO**
Start I/O.

**SLU**
Secondary logical unit.

**SMF**
System management facility.

**SNA**
Systems Network Architecture.

**SNMP**

**soft error**
(1) An error that occurs sporadically and that may not appear on successive attempts to read data. Synonymous with *transient error*. (2) An intermittent error on a network that requires retransmission. Contrast with *hard error*.

**Note:** A soft error by itself does not affect overall reliability of a network, but reliability may be affected if the number of soft errors reaches the ring error limit.

**SSCP**
System services control point.

**subarea**
A portion of the SNA network consisting of a subarea node, attached peripheral nodes, and associated resources. Within a subarea node, all network accessible units (NAUs), links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

**Systems Network Architecture (SNA)**
The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.
teleprocessing request block (TPRB)
A function used by NPM to control input/output requests to or from terminals to files.

threshold
(1) In the NetView program, a percentage value, set for a resource and compared to a calculated error-to-traffic ratio. (2) In NPM, high or low values supplied by the user to monitor data and statistics being collected.

TP
(1) Transmission priority. (2) Transaction program.

TPR
Transaction processing routine.

TPRB
Teleprocessing request block.

transit time
In NPM, transit time is the same as response time. See *response time, host time, network time*, and *operator time*.

unbind
In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request*. Contrast with *BIND*.

Virtual Storage Access Method (VSAM)
An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Telecommunications Access Method (VTAM)
An IBM-licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VSAM
Virtual Storage Access Method.

VTAM
Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

write-to-operator with reply (WTOR)
An optional user-coded service whereby a message can be written to the system console operator informing the operator of errors and unusual system conditions that may need correcting.

WTOR
Write-to-operator with reply.
Index

A

abnormal end (ABEND)
  abnormal end problems
    choosing keywords 30
    documenting 38
    possible causes 30
    symptoms 30
  using PDI control blocks to diagnose 73
dump worksheet 158
locating codes 38
NPM panel (FNM09ABN) 108
  specifying codes when performing an abend 108
abnormally ending NPM 108
accounting problems, network 18, 34
ACTION parameter on TRAP statement 121
Add Diagnostic Trap panel (FNM09TRA) 102
address space
  NetView program address space 52
  NPM address space 37
  NSI address space 57
  SSI address space 52
alerts problems, performance 20, 61
allocation, virtual storage
  NPM 74
  private 75
AMASPZAP dump 62, 66
AMBLIST dump 62, 66
AMODE 99
APAR (authorized programming analysis report) 8
API parameter on DEBUG statement 110
archive data management problems
  documenting 48
  search argument 34
arguments, search 34
authorized program analysis report (APAR) 8

B

BACKWARD command 92
BAF (search argument) 34
Batch Archive Facility (BAF) problems, search argument 34
batch reporting, problems with
  checklist 10
  documenting 49
books
  feedback xvi
  online xvi
  ordering xvi
BUFSCAN parameter on DEBUG statement 111

C

CALID parameter on TRAP statement 121
CBS parameter on DEBUG statement 111
CHAR command 90, 92
checklists
  batch reporting utility 10
  dynamic definite response (DDR) 11
  dynamic network collection (DNC) 12
  LAN bridge collection 13
  LAN segment utilization data 14
  NCP definition 15
  NCP EXECs 19
  NetWare resources collection 16
  network accounting 18
  network collection 18
  NPM performance alerts 20
  ODLC LAN resource collection 21
  RTM collection 22
  session collection 23
  SMF collection 25
  system console support 24
  transit analysis program 26
  VTAM statistics collection 27
checkpoint messages for NetWare 166
checkpoint processing for NetWare 17, 165
choosing keywords 29
classifying problems 6
CMS environment message format 38
CODE parameter on TRAP statement 121
collecting information on problems 6
command trace entry (SC, RC) 134
common address table (CAT)  
  internal trace table 127
  locating using the NPM internal trace 127
  locating using the NPM message trace 149
  message trace 149
component ID number 36
collection report
  configuration hierarchy based on NCP RRT 77
  configuration summary from NCP RRT 7, 76
  Configuration Symbol Resource Table 78
console dump 65
log 65
support problems
  checklist 24
  documenting 65
control blocks
  names 93
  saving 98
  worksheet 157
  writing to GTF 111, 112
COUNT parameter on TRAP statement 121
Customer Support xvii
D
DEBUG statement
parameters:
- API 110
- BUFSSCAN 111
- CBS 111
- DSP 111
- ESTAE 112
- GTF 112
- IOB 112
- LAN 112
- MSG 112
- MSGMOD 113
- MTRACE 113
- MTRCSIZE 113
- PNLTEST 113
- PRINTMBR 114
- PRINTMSG 114
- RCYDUMP 114
- RTN 114
- RTR 114
- SA VECB 115
- SDUMP 115
- SNAP 115
- SPO 115
- SVC 115
- SYSDUMP 115
- SYSOUT 116
- TCPIPO 117
- TCPIPT 116
- TEST 117
- TRAP 117
- TRAPCNT 117
- TRCSIZE 117
- UXT 118
- VSM 118
- VTAM 118
- VTAMIM 118

trace options, enabling and disabling 129
diagnostic options, modifying 98
see checklists 9
trace header 127, 149
traps 102
worksheets 155
Diagnostic Trace Options panel, Display/Modify (FNM09DTF) 96
Diagnostic Trap - Message panel (FNM09DT1) 106
Diagnostic Trap panel, Add (FNM09TRA) 102
disabled loop 32, 43
dispatcher trace entry (QU, CA, EX, TI, WA, TE, RE, DI, PO, RT, SV, SR) 132
display
- commands 92
- storage 90, 92
- trace options 96
Display Main Storage panel (FNM09DMS) 90
Display/Modify Diagnostic Trace Options panel (FNM09DTF) 96
Display/Modify Operator Trace panel (FNM09OPR) 100
Display/Modify Options panel (FNM09OPT) 98
Display/Modify Session Trace panel (FNM09OPS) 101
Display/Modify Trap Status panel (FNM09TRS) 104
DNC collection problems 12
documentation problems
- choosing keywords 30
- documenting 41
- symptoms 30
documenting problems 35
- DR (search argument) 34
DSP parameter on DEBUG statement 112
DUMP command 90, 92
dumps
- AMASPZAP 62, 66
- AMBLIST 62, 66
- NetView program address space 52
- NPM, based on operating system 37
- NPM address space 37
- NPM common storage area (CSA) 37
- NSI address space 57
- SSI address space 52
dynamic definite response (DDR) function problems
checklist 11
documenting 50
dynamic reconfiguration, incorrect 34

E
e-mail contact xvii
enabled loop 32, 43
entries in internal trace table 129
ESTAE,
- entered 30, 73
parameter on DEBUG statement 112
event identifier (EID), on internal trace table 129
exit routines, tracing 96
exit trace entry, installation-wide 144

F
failing
- instruction 39
- module 39
types 6, 29
FDBK2 parameter on TRAP statement 122
FDBWD parameter on TRAP statement 122
feedback about publications xvii
FIL (search argument) 34
files, transfer to tape 37, 53
flowchart, diagnostic procedure 5
FNM09ABN (Abend NPM panel) 108
FNM09DBM (Problem Determination Facilities panel) 89
FNM09DMP (Take an NPM Storage Dump panel) 107
FNM09DMS (Display Main Storage panel) 90
FNM09DT1 (Diagnostic Trap - Message panel) 106
FNM09DTF (Display/Modify Diagnostic Trace Options panel) 96
FNM09OPR (Display/Modify Operator Trace panel) 100
FNMO9OPS (Display/Modify Session Trace panel) 101
FNMO9OPT (Display/Modify Options panel) 98
FNMO9TRA (Add Diagnostic Trap panel) 102
FNMO9TRS (Display/Modify Trap Status panel) 104
FNMArchv problems 48
FNMILOG data set
  example of initialization statements 7
FNMILOG dataset
  example of 79
  initialization statements 74
  message symptoms 32
  subpool abbreviations 75
FNMSRT file, used in diagnosing
  NCP definition problems 53
  network collection problems 56
  session collection problems 63
  start collection commands problems 84
FNMTAP problems 26, 68
FNMVARCH EXEC problems 48
FORWARD command 92
  function problems, documenting specific 47

G

GCS
  environment message format 38
generalized trace facility
  event identifier (EID), use of 129
  on Diagnostic Trace Options Panel 96
  parameter on DEBUG statement 112
  trace dataset 63

H

HEX command 90, 92
  hexadecimal core dump, example of 127, 149
  high processor utilization 32

I

I/O buffers, writing to GTF 97
ID number 36
ID parameter on TRAP statement 122
  identifying operating system 36
  incorrect documentation 30
  incorrect output problems
    choosing keywords 31
    documenting 42
    symptoms 32
Information/Access program 8
  information on problems, collecting 6
  initialization statements
    DEBUG 110
    TRAP 119

initialization statements (continued)
  using FNMILOG data set 7
installation problems 7
installation-wide exit trace
  calls 97
  entry (IX) 144
  instruction, failing 39
Interactive Problem Control System (IPCS) 38
  internal trace table
    finding 127
    understanding entries 129
    used in traps 6, 126
internal trace worksheet 156
IOB parameter on DEBUG statement 112
IOB trace entry (IO) 145
IPCS (Interactive Problem Control System) 38
  ISTATUS parameter on TRAP statement 122

K

keywords to describe problems
  choosing symptoms
    ABEND 30
    DOC 30
    INCORROUT 31
    LOOP 32
    MESSAGE 32
    PERFM 33
    WAIT 34
    definition 29
  using to search RETAIN database 7

L

lack of response to NPM 34
LAN bridge collection problems
  checklist 13
  documenting 51
LAN control block 93
LAN parameter on DEBUG statement 112
LAN segment utilization data collection problems
  checklist 14
  documenting 52
LAN services
  on Diagnostic Trace Options Panels 97
  trace entry (LN) 138
logic
  trace entry (LG) 137
LOOP problems
  choosing keywords 32
  documenting 43
  symptoms 32
LOP (search argument) 34
LSEGCOLL command 14
NCP definition problems
  checklist 15
  documenting 53
NCP resource resolution table (RRT)
  configuration hierarchy 77
  configuration summary 76
NET (search argument) 34
NetView Performance Monitor (NPM)
  abend dump worksheet 158
  dump, example of 39
  EXEC definition problems
    checklist 19
    documenting 35, 61
  initialization datasets
    configuration report, establishing 74
    contained in FNMILOG 74
  internal trace worksheet 156
  major control blocks worksheet 157
NetView Performance Monitor (NPM) (continued)
  save area layout worksheet 159
  storage, dumping 107, 121
  Storage Dump panel, Take an (FNMO9DMP) 107
  testing 98
NetView synergy interface (NSI) problems
  documenting 57
  search argument 34
NetWare resources collection problems
  checklist 16
  checkpointing 165
  documenting 54
  resource definition 165
  restarting active collections 166
network accounting problems
  checklist 18
  search argument 34
network collection problems
  checklist 18
  documenting 56
  search argument 34
network log 34
NGA (search argument) 34
no response problems 34
NODE parameter on TRAP statement 123
NPM-to-NPM
  router requests and responses, tracing 97
  trace entry (LQ, LR, RQ, RR) 140
NSA (search argument) 34
NSI (search argument) 34
O
ONL (search argument) 34
online problem determination facilities 89
online publications xvii
online system problems
  documenting 60, 87
  search argument 34
operating system
  identifying 36
  service requests, tracing 97
operator current status, determining 100
Operator Trace panel, Display/Modify (FNMO9OPR) 100
operator traces, displaying and modifying 100
order number, book 41
ordering publications xvii
out-of-storage problem 38
outboard data link control (ODLC) LAN resource data
collection problems
  checklist 21
  documenting 59
  output, incorrect 42
PANELID command
diagnosing with 87
using 7, 37
panels
identifying 7, 87
printing 87
testing 98
using SYSPRINT data set 84
using SYSPRINT dataset 88
PDI (problem determination information) control block
diagnosing with 40, 73
structure 73
worksheet 160, 161
performance alerts problems
checklist 20
documenting 61
search argument 34
performance measurement table
control block names 93, 94
NPM performance alerts 20
Performance Measurement Table Display panel, control
table 75
performance problems
choosing keywords 33
documenting 45
symptoms 33
PFLGS parameter on TRAP statement 123
PMASK parameter on TRAP statement 123
PNLID parameter on TRAP statement 124
PNLTEST parameter on DEBUG statement 113
presentation service trace entry (PS) 139
PRINT command
SYSPRINT data set 84, 87
using 7, 37
printing messages
from Problem Determination Facilities panel 98
on DEBUG statement 114
SYSPRINT data set 84
printing panels 87
PRINTMBR parameter on DEBUG statement 114
PRINTMSG parameter on DEBUG statement 114
problem determination facilities, online 89
Problem Determination Facilities panel (FNM09DBM) 89
problem determination information (PDI) control block 73
problem types 6, 29
problems
classifying 6
documenting
ABEND 38
archive data management 48
basic information needed for all 35
batch reporting utility 49
DOC 41
dynamic definite response (DDR) 50
general information 35
INCORROUT 42
LAN bridge collection 51
LAN segment utilization data 52
LOOP 43
MESSAGE 44
NCP definitions 53
problems (continued)
documenting (continued)
NetWare resources collection 54
network collection 56
NSI (NetView synergy interface) 57
ODLC LAN synergy interface 59
online system 60, 87
PERFM 45
performance alerts 61
RTM collection 62
session analysis 63
session collection 63
SMF collection 66
system console support 65
transit analysis program 68
WAIT 46
during installation 7, 12
general information on diagnosing 5, 7
reporting 7, 31
searching the RETAIN database for solutions 7
symptoms
ABEND 30
DOC 30
INCORROUT 31
LOOP 32
MESSAGE 32
PERFM 33
WAIT 34
using tools to diagnose 6, 73
processor utilization, high 32
program
number 36
program status word (PSW) 39
repeating 32
program temporary fix (PTF)
building 8, 39
recording 36
SYSMOD, recording 40
publications
feedback xvi
online xvi
ordering xvi
R
RC parameter on TRAP statement 124
RCYDUMP parameter on DEBUG statement 114
release number, operating system 36
reporting problems to the IBM Support Center 37
reporting problems to the Support Center 7
REQ parameter on TRAP statement 124
request errors, tracing 97
request parameter list trace entry (VE, VS) 135
response, poor 34
response time monitor (RTM) collection problems
checklist 22
documenting 62
search argument 34
worksheet 161
S
save area layout worksheet 159
SAVECB parameter on DEBUG statement 115
SDUMP parameter on DEBUG statement 115
search arguments 34
searching RETAIN for solutions 7
SES (search argument) 34
session collection and analysis problems
  checklist 23
  documenting 63
  search argument 34
session trace entry (SA) 141
Session Trace panel, Display/Modify (FNM09OPS) 101
session traces, displaying and modifying 101
slow response problems 33
SMF
  collection problems
    checklist 25
    documenting 66
    search argument 34
snap dump, performing 107
SNAP parameter on DEBUG statement 115
specific function problems, documenting
  archive data management 48
  batch reporting utility 49
  dynamic definite response (DDR) 50
  LAN bridge collection 51
  LAN segment utilization data 52
  NCP definitions 53
  NetWare resources collection 54
  network collection 56
  NSI (NetView synergy interface) 57
  ODLC LAN resource data collection 59
  online system 60
  performance alerts 61
  RTM collection 62
  session collection and analysis 63
  SMF collection 66
  system console support 65
  transit analysis program 68
SPO commands, tracing 97
SPO parameter on DEBUG statement 115
statements, initialization
  DEBUG 110
  TRAP 119
storage, displaying
  main 90
  virtual 91
storage, dumping 107
Storage Dump panel, Take an NPM (FNM09DMP) 107
storage problem 38
subpools, in FNMILOG, abbreviations of 75
subsystem interface (SSI) address space 51, 52
Support Center 7
SVC parameter on DEBUG statement 115
synonym chain 29, 78
Synonym Chain Distribution Table 78
syntax diagrams, how to use xix
SYSDUMP parameter on DEBUG statement 115
SYSOUT parameter on DEBUG statement 116
SYSPRINT data set
  and WAIT problems 46
  example of 84
SYSPRINT dataset
  and DEBUG statement 114
  and message symptoms 30, 32
  contents 7
  example of 84
  on Problem Determination Facilities panel 98
system console support
  log 37
T
Take an NPM Storage Dump panel (FNM09DMP) 107
task, incorrect output 34
TASK abend trace entry (KA) 143
TASK start/TASK end trace entry (KS, KE) 142
TCPiPO parameter on DEBUG statement 117
TCPiPT parameter on DEBUG statement 116
terminating NPM abnormally 108
TEST parameter on DEBUG statement 117
testing NPM 98
  testing panels 98
Tivoli Customer Support xvii
tools, diagnostic 6, 73
Totals by Resource Type Total Table 78
trace dispatcher events, tracing 96
trace options
  and DEBUG initialization statement 6
  displaying 96
  exceptions 129
trace status of specific resource, determining 101
trace table, internal
  APPCIOE trace entry 129, 147
  command trace entry 129, 134
  dispatcher trace entry 129, 132
  example of, locating in dump 127
  exit trace entry, installation-wide 129, 144
  general information 127
  installation-wide exit trace entry 129, 144
  intertask pipe trace entry 146
  IOB entry 129, 145
  LAN service trace entry (LN) 130, 138
  locating, in dump 127, 149
  logic trace entry 129, 137
  message trace entry 129, 133
  NPM-to-NPM trace entry 130, 140
<table>
<thead>
<tr>
<th>Trace Type</th>
<th>Internal Trace Entries</th>
<th>Request Parameter List Trace Entries</th>
<th>RPL Trace Entries</th>
<th>Session Trace Entries</th>
<th>TASK Abend Trace Entries</th>
<th>TASK Start/TASK End Trace Entries</th>
<th>TCP/IP Trace Entry</th>
<th>Traces, Information Shown in</th>
<th>VSAM and VTAM Request Parameter List Trace Entries</th>
<th>VTAM Exit Trace Entries</th>
<th>VTAM Request Parameter List Trace Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace Tables, Locating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace Types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traces, Displaying and Modifying</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traces, Performing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transfer Files to Tape</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transit Analysis Program Problems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transit Time Values, Problems with</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trap - Message Panel, Diagnostic (FNM09DT1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trap Dump, Performing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRAP Parameter on DEBUG Statement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRAP Statement Parameters</td>
<td>ACTION 121</td>
<td>CODE 121</td>
<td>FDBK2 122</td>
<td>FDBWD 122</td>
<td>ID 122</td>
<td>ISTATUS 122</td>
<td>MODID 122</td>
<td>MODULE 123</td>
<td>MSGID 123</td>
<td>MSGNR 123</td>
<td>NODE 123</td>
</tr>
<tr>
<td></td>
<td>COUNT 121</td>
<td>CODE 121</td>
<td>FDBK2 122</td>
<td>FDBWD 122</td>
<td>ID 122</td>
<td>ISTATUS 122</td>
<td>MODID 122</td>
<td>MESSAGE 124</td>
<td>PFLGS 123</td>
<td>PMASK 123</td>
<td>PNLID 124</td>
</tr>
<tr>
<td></td>
<td>FDBK2 122</td>
<td>FDBWD 122</td>
<td>ID 122</td>
<td>ISTATUS 122</td>
<td>ID 122</td>
<td>ISTATUS 122</td>
<td>MODID 122</td>
<td>MESSAGE 124</td>
<td>PFLGS 123</td>
<td>PMASK 123</td>
<td>PNLID 124</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>USERID 124</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trap Status Panel, Display/Modify (FNM09TRS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRAPCNT Parameter on DEBUG Statement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Traps (continued)**
- Internal Trace Classes 126
- Modifying 104
- Types of 102, 106
- TRCSIZE Parameter on DEBUG Statement 117
- TYPE Parameter on TRAP Statement 120

**U**
- User-abend Messages 38
- USERID Parameter on TRAP Statement 124
- Using Keywords to Search Database 7
- UXT Parameter on DEBUG Statement 118

**V**
- Virtual Storage Allocation of 74
- Displaying 92
- Virtual Storage Allocation by Subpool Report 75
- VSAM Request Errors, Tracing 97
- VSM Parameter on DEBUG Statement 118
- VTAM Exit Trace Entry (VX) 136
- VTAM Request Errors and Exit Routines, Tracing 96
- VTAM SPO Commands, Tracing 97
- VTAM Statistics Collection, Problems with
  - Checklist 27
  - Documenting 69
- VTAMIM Parameter on DEBUG Statement 118

**W**
- Wait Problems
  - Choosing Keywords 34
  - Documenting 46
  - Symptoms 34
- Worksheets
  - NPM Abend Dump 158
  - NPM Internal Trace 156
  - NPM Major Control Blocks 157
  - NPM Save Area Layout 159
  - Problem Determination Information Control Block 160
  - RTM2 Work Area Layout 161