Transaction Processing Facility

Program Development Support Reference

Version 4 Release 1

SH31-0164-13

Transaction Processing Facility

Program Development Support Reference

Version 4 Release 1

SH31-0164-13

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

Fourteenth Edition (June 2002)

This is a major revision of, and obsoletes, SH31-0164-12 and all associated technical newsletters.

This edition applies to Version 4 Release 1 Modification Level 0 of IBM Transaction Processing Facility, program number 5748-T14, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

IBM welcomes your comments. Address your comments to:

IBM Corporation
TPF Systems Information Development
Mail Station P923
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

Contents

Figures vii

Tables ix

Notices xi
Trademarks xi

About This Book xiii
Who Should Read This Book xiii
How This Book is Organized xiii
Conventions Used in the TPF Library xiii
Related Information xiv
IBM Transaction Processing Facility (TPF) 4.1 Books xiv
IBM ESA/390 Books xiv
Online Information xiv
How to Send Your Comments xv

Introduction to Programming Development Support 1

Program Test Vehicle (PTV) Online Utility 3
Testing Philosophy 3
Program Test Vehicle (PTV) Modes 4
Program and Database Security 4
Trace Input 5
Test Unit Tape (TUT) 5
Pilot Tape 6
Online Disk Packs 6
PTV Control Records 7
Input Messages 11
Required Input Sequence 11
Sample Job Control Language (JCL) for the Test Unit Tape (TUT) 11
Tape Requirements 12
Test Activation Procedures 13
Package Tests 13
Error Recovery 13
System Test 14
Live Test 14
PTV Control Blocks 14
Test Output 14
Input Messages 14
Messages 15
Processing Overview 15

System Test Compiler (STC) Offline Tape Generation Utility 17
DRIL Record Layout 18
Example of a DRIL Record 19
Required Formats for STC Input 21
Disk Allocation Records 21
Disk Type Records 21
RUNID 21
Terminal Simulation Records 22
Dump Option Records 22
Data Generation 23

DATA and MSG Records 23
GSTAR and GEND Records 24
Data Generation Detail Records 25
Data Generation Operations 27
Data Record Prefix Entries 29
Group Records 31
Examples of Data and Message Record Generation 34
Pilot Tape Generation 37
Standard Data Message File Update (SDMU) Program 37
SDMF Generation 38
SDMF Maintenance 39
SDMF Regeneration 41
SDMF Dump Requests 42
Required Input Sequence 43
Creating One Pilot System 43
Creating One Test Unit 43
Using SDMU Functions 43
File Input 44
Component Programs 44
File Output 44
Messages 44
Processing Overview 45

Real-Time Trace (RTT) System Utility 47
Trace Input 48
Trace Output 48
Example of RTT Output 48
Description of Alpha Pointers in the RTT Output 49
Processing Overview 51

Selective File Dump and Trace (SFDT) Debugging Tools 53
Selective File Trace (SFT) Debugging Tool 53
Selective File Dump (SFD) Debugging Tool 53
Trace Output 54
Selective File Dump (SFD) Output Example 54
Selective File Trace (SFT) Output Example 54
Optional Log Function Example 55
Processing Overview 56

Diagnostic Output Formatter (DOF) Online and Offline Utility 61
Program Event Recording (PER) Facility 62
Trace Tables 62
Collated Macro Trace Table 62
Collated Input/Output (I/O) Trace Table 63
Systems Network Architecture Input/Output (I/O) Trace Table 63
Path Information Unit (PIU) Trace Table 63
Program Test Vehicle (PTV) Utility 64
DUMP Options 65
Terminal Simulation 65
Selective File Dump (SFD) Debugging Tool 66
Selective File Trace (SFT) Debugging Tool 66
Description of a Main Storage Dump 66
Cover Page 67
Dump Label Index 69
I-Stream Status Display 70
Area of Program Error 74

Areas Referenced by General Registers . . . 75
C/C++ Language Register Environments . . . 76
Collated Macro Trace Table . . . 81
Collated Input/Output (I/O) Trace Table . . . 88
Prefix Page . . . 91
Branch Trace Table . . . 92
ECB Virtual Memory (EVM) . . . 93
Additional Blocks Mapped . . . 100
System Storage . . . 102
Dumps of ECB Working Storage in the System Virtual Memory (SVM) 103
ECB Working Storage Dumps in the System Virtual Memory (SVM) . . . 103
Link Map Data for C Load Modules . . . 110
Main Storage Dump Labels . . . 112
Dump Labels for the Entry Control Block (ECB) . . . 126
Dump Labels for the ISO-C Control Structures . . . 129
Dump Labels for the ISO-C Stack Area . . . 129
Processing Overview . . . 130

C Function Trace . . . 133
Summary of Command and Macro Options for C Function Trace . . . 134
Controlling Traces . . . 134
Selecting Programs to Be Traced at Compile Time . . . 134
Setting System Wide Trace Options . . . 134
Controlling Specific ECBs . . . 135
Activating Loadsets with the E-Type Loader . . . 136
System Error Processing . . . 136
Selecting ECBs for Formatting and Dumping C Function Trace Entries 137
Other Breakpoints . . . 137
Format and Description of C Function Trace . . . 138
Trace Output without Stack and Static Data . . . 138
Trace Output with Stack and Static Data . . . 142
Trace Output with Stack and without Static Data . . . 145
Trace Output with Static and without Stack Data . . . 147
Trace Output with Other Breakpoint Entries . . . 149
Question Marks (????????) in the CALLING/RETURN PARAMETERS Field 150
Customizing Trace Information . . . 151
User Exits . . . 151
C Function Trace Use of the SETOC Macro . . . 152

TPF MQSeries Local Queue Manager Support . . . 153
TPF MQSeries Trace and Postprocessing . . . 153
Communications Trace . . . 153
Function Trace . . . 153
Using TPF MQSeries Trace . . . 154
Postprocessing the Trace Data . . . 154
Communications Trace Example . . . 154
Function Trace Example . . . 156

Index . . . 161

Figures

1. Program Segment Name Example . . . 11
2. Sample JCL for the TUT for Phase 3 (Package Test) . . . 12
3. PTV Input/Output Message Stream . . . 15
4. Program Test Vehicle Processing Overview . . . 16
5. Data Generation Operations – Enter (ENT V.) Example . . . 27
6. Data Generation Operations – Enter Item Example . . . 28
7. Data Generation Operations – Repeat Example . . . 28
8. Data Generation Operations – Repeat Item Example . . . 28
9. Data Generation Operations – Subtract Example . . . 28
10. Data Generation Operations – Subtract Item Example . . . 29
11. Data Generation Operations – Add Example . . . 29
12. Data Generation Operations – Add Item Example . . . 29
13. Fixed File Load Address Example . . . 30
14. New Record Length Example . . . 31
15. Group Records Example . . . 32
16. Group Record Example . . . 32
17. Example of Records that Generate a Group Record . . . 33
18. GROUP Record . . . 33
19. Manual Generation of Data and Message Records . . . 35
20. Manual Generation of Data and Message Records . . . 35
21. Data Record Creation with DRIL . . . 36
22. Mixed SDMF, DRIL, and Manual Generation . . . 36
23. 3270 Input Message with Numeric Displacements . . . 37
24. 3270 Input Message using AM0SG Format . . . 37
25. Sample SDMU CREAT Run . . . 39
26. Sample SDMU UPDAT Run . . . 41
27. Sample SDMU REGEN Run . . . 42
28. System Test Compiler Processing Overview . . . 45
29. RTT Output Example . . . 49
30. Trace Count (ZCNTM) Output Example . . . 51
31. Real-Time Trace Processing Overview . . . 52
32. Selective File Dump (SFD) Output Example . . . 54
33. Selective File Trace (SFT) Output Example . . . 55
34. Optional Log Function Example . . . 56
35. Selective File Dump Processing Overview . . . 57
36. Dump Cover Page . . . 67
37. Dump Label Index . . . 70
38. I-Stream Status Display . . . 71
39. Dump Cover Page . . . 74
40. Areas Referenced by General Registers . . . 75
41. Collated Macro Trace Table . . . 82
42. Collated I/O Trace . . . 89
43. Prefix Page . . . 92
44. Branch Trace Table . . . 93
45. ECB Virtual Memory (EVM) . . . 94
46. Additional Blocks Mapped . . . 100
47. System Storage . . . 102
48. Dump of ECB Working Storage in the System Virtual Memory (SVM) . . . 104
49. Formatted Link Map Data in a Dump . . . 111
50. Diagnostic Output Formatter Processing Overview . . . 130
51. Trace Output without Static or Stack Data . . . 139
52. Trace Output with Stack and Static Data . . . 143
53. Trace Output with Stack and without Static Data . . . 146

54. Trace Output with Static and without Stack Data 148
55. Trace Output with Other Breakpoint Entries 150
56. Calling or Return Parameters That Are not Addressable 151
57. TPF MQSeries Communications Trace Output Example 155
58. TPF MQSeries Function Trace Output Example 157

Tables

1. Column Layout of a DRIL Record . . . 18
2. RUNID Format for Package Tests . . . 21
3. Terminal Simulation Record Format . . . 22
4. Dump Option Record Format . . . 22
5. DATA Format . . . 24
6. MSG Record Format . . . 24
7. GSTAR Format . . . 24
8. GEND Format . . . 25
9. Data Generation Detail Record Format . . . 25
10. Data Generation Detail Records . . . 26
11. ORD Format . . . 30
12. SIZ Format . . . 31
13. GROUP Record Format . . . 34
14. SDMU CREAT Format . . . 38
15. SDMU ENTER Format . . . 38
16. SDMU END Format . . . 39
17. SDMU UPDAT Format . . . 39
18. SDMU ALTER Format . . . 40
19. SDMU DELET Format . . . 40
20. SDMU REGEN Format . . . 41
21. SDMU DUMPD Format . . . 42
22. SDMU DUMPR . . . 42
23. SDMU DUMPT . . . 43
24. Headings in the Main Storage Dump for the Cover Page Section . . . 67
25. Headings in the Dump Label Index . . . 70
26. Headings in the I-Stream Status Display . . . 71
27. Headings in the Dump Cover Page . . . 75
28. Headings in the Areas Referenced by General Registers . . . 76
29. Register Conventions for C/C++ Language . . . 76
30. Headings in the Collated Macro Trace Table . . . 87
31. Headings in the Collated Input/Output (I/O) Trace Table . . . 90
32. Headings in the Prefix Page . . . 92
33. Headings in the Branch Trace Table . . . 93
34. Headings in the ECB Virtual Memory (EVM) . . . 98
35. Headings in the ECB Virtual Memory (EVM) . . . 99
36. Headings in the Additional Blocks Mapped . . . 101
37. Headings in the Additional Blocks Mapped . . . 101
38. Headings in the System Storage . . . 102
39. Headings in the Dump of ECB Working Storage in the System Virtual Memory (SVM) . . . 109
40. Labels Found in Main Storage Dumps . . . 112
41. Dump Labels for the Entry Control Block (ECB) . . . 126
42. Dump Labels for ISO-C Control Structures . . . 129
43. Dump Labels for the ISO-C Stack Area . . . 129
44. Manipulation of C Function Trace and Trace Suboptions . . . 134
45. Descriptions of Alpha Pointers . . . 140

Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 830A
Mail Drop P131
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this book to non-IBM Web sites are provided for convenience only and do not in any way serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this book or accessed through an IBM Web site that is mentioned in this book.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking

APPN

CICS

C/370

IBM

MQSeries.

Other company, product, and service names may be trademarks or service marks of others.

About This Book

This book describes the tools provided by the TPF 4.1 system to assist in testing application programs. These tools help you to control the test environment and progressively test each application program under various conditions until you achieve a high level of confidence.

In this book, abbreviations are often used instead of spelled-out terms. Every term is spelled out at first mention followed by the all-caps abbreviation enclosed in parentheses; for example, Systems Network Architecture (SNA). Abbreviations are defined again at various intervals throughout the book. In addition, the majority of abbreviations and their definitions are listed in the master glossary in the *TPF Library Guide*.

Who Should Read This Book

This book is intended for application programmers responsible for testing application programs designed to run on the TPF 4.1 system.

How This Book is Organized

This book is organized by chapter. After the introduction, each chapter describes a different tool that you can use to test your application program.

Conventions Used in the TPF Library

The TPF library uses the following conventions:

Conventions	Examples of Usage
italic	Used for important words and phrases. For example: A *database* is a collection of data. Used to represent variable information. For example: Enter **ZFRST STATUS MODULE** *mod*, where *mod* is the module for which you want status.
bold	Used to represent text that you type. For example: Enter **ZNALS HELP** to obtain help information for the ZNALS command. Used to represent variable information in C language. For example: **level**

`monospaced`	Used for messages and information that displays on a screen. For example:
	`PROCESSING COMPLETED`
	Used for C language functions. For example:
	`maskc`
	Used for examples. For example:
	`maskc(MASKC_ENABLE, MASKC_IO);`

bold italic	Used for emphasis. For example: You ***must*** type this command exactly as shown.
Bold underscore	Used to indicate the default in a list of options. For example: **Keyword=OPTION1** | **DEFAULT**

Conventions	Examples of Usage
Vertical bar |	Used to separate options in a list. (Also referred to as the OR symbol.) For example: **Keyword=Option1** | **Option2** **Note:** Sometimes the vertical bar is used as a *pipe* (which allows you to pass the output of one process as input to another process). The library information will clearly explain whenever the vertical bar is used for this reason.
CAPital LETters	Used to indicate valid abbreviations for keywords. For example: KEYWord=*option*

Scale	Used to indicate the column location of input. The scale begins at column position 1. The plus sign (+) represents increments of 5 and the numerals represent increments of 10 on the scale. The first plus sign (+) represents column position 5; numeral 1 shows column position 10; numeral 2 shows column position 20 and so on. The following example shows the required text and column position for the image clear card.
	`|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...`
	`LOADER  IMAGE CLEAR`
	Notes:
	1. The word LOADER must begin in column 1.
	2. The word IMAGE must begin in column 10.
	3. The word CLEAR must begin in column 16.

Related Information

A list of related information follows. For information on how to order or access any of this information, call your IBM representative.

IBM Transaction Processing Facility (TPF) 4.1 Books

- *TPF ACF/SNA Data Communications Reference*, SH31-0168
- *TPF General Macros*, SH31-0152
- *TPF Main Supervisor Reference*, SH31-0159
- *TPF Non-SNA Data Communications Reference*, SH31-0161
- *TPF Operations*, SH31-0162
- *TPF System Installation Support Reference*, SH31-0149
- *TPF System Macros*, SH31-0151.

IBM ESA/390 Books

- *ESA/370 Principles of Operation*, SA22-7200
- *ESA/390 Principles of Operation*, SA22-7201.

Online Information

- *Messages (Online)*
- *Messages (System Error and Offline)*.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other TPF information, use one of the methods that follow. Make sure you include the title and number of the book, the version of your product and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

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

- If you prefer to send your comments electronically, do either of the following:
  - Go to http://www.ibm.com/tpf/pubs/tpfpubs.htm.

    There you will find a link to a feedback page where you can enter and submit comments.
  - Send your comments by e-mail to tpfid@us.ibm.com
- If you prefer to send your comments by mail, address your comments to:

  IBM Corporation
  TPF Systems Information Development
  Mail Station P923
  2455 South Road
  Poughkeepsie, NY 12601-5400
  USA
- If you prefer to send your comments by FAX, use this number:
  - United States and Canada: 1 + 845 + 432 + 9788
  - Other countries: (international code) + 845 + 432 +9788

Introduction to Programming Development Support

You should extensively test every application program before it is integrated with the online system. Each application program must:

- Perform its function effectively without disrupting existing program relationships
- Be able to interface with the TPF 4.1 system and other application programs.

Program development support programs enable you to:

- Check for logic violations
- Specify or change data for each test
- Interrupt, modify, resume, or end testing
- Start online debugging tools
- Batch test runs
- Customize data for each test
- Maintain a uniform database
- Restore the database between tests
- Produce various diagnostic reports.

There are inherent problems involved when testing application programs for a multiple program system. The programming effort is large. Each time a new function is added to the system, logic complexity is affected. Clearly, all conditions cannot be tested. Symptoms of some errors can be propagated through the system before the error is detected. When the symptoms do appear, it can be difficult to determine the condition that caused the error or to duplicate the condition.

The following utilities were developed to help you control the test environment and progressively test each application program under various conditions until a high level of confidence is established.

Utility Name	Program Description
Program Test Vehicle (PTV)	PTV runs in a user-defined test environment. When used with offline and online test tools, PTV enables you to create and control the environment, activity, and reporting of each test unit.
System Test Compiler (STC)	STC is an offline utility that generates the data records for program test vehicle (PTV) input and system installation.
Real-Time Trace (RTT)	RTT monitors programs according to parameters specified with the activating command. RTT counts or traces the use of selected macros and related data (data blocks, blocks, registers, and ECBs associated with the macro execution) according to options that you select.
Selective File Dump and Trace (SFDT)	SFDT are utilities that dump either the content of specified files or the records that were updated during the trace period to a real-time tape. You specify the trace and dump options with the activating commands.

Diagnostic Output Formatter (DOF)
: The DOF formats and prints diagnostic data. The type of output produced depends on the system activity (test, trace, error dump) that recorded the data.

This book describes each of these programs in the chapters that follow. Emphasis is given to the function performed rather than the programming that produces the result.

Program Test Vehicle (PTV) Online Utility

Program test vehicle (PTV) is an online utility (referred to as the PTV utility in the remainder of this book) that enables you to test application programs in progressive levels of system involvement. Applications are tested individually, with related programs, within a transaction, and within a full TPF system.

Note: The PTV utility is allowed only in a uniprocessor environment.

There are 3 distinct test activities:

- Preparation of a test unit tape (TUT)

  A test unit is defined by you. User-defined input is recorded on the test unit tape by the system test compiler (STC).

  You provide control record information that defines each test unit. These records specify:

  – The test environment
  – Whether pilot tapes or pool directories will be loaded
  – Whether message input will be in burst or asynchronous mode
  – Whether there will be live input
  – The amount of data to be traced and under what conditions
  – When the database should be restored
  – What devices, if any, will be simulated
  – When dumps will be taken.
- Preparation of an operational system

  An operational system includes the system segments, application programs, and data stored on disk packs called online modules. These are created by the system loader. See the *TPF System Installation Support Reference* for more information about the system loader.
- Processing of applications programs.

  During this phase, the PTV utility performs tests and provides information required to verify program reliability.

Data from the test run is written to the real-time log tape. PTV output is formatted and printed by the offline diagnostic output formatter (DOF).

Testing Philosophy

A testing philosophy has been maintained in the PTV utility that recognizes the requirement for systematic program testing. The following terms are used to describe the gradients of program testing.

Term	Definition
Package Unit	Several application programs to be tested together to verify performance of specified functions.
Transaction Unit	A complete transaction to be tested in a single thread environment. If all application programs are available, the driver is not required. It can be used to monitor the test.

Full Scale	Multithread testing with simulation of the full environment in which the application programs operate.

Program Test Vehicle (PTV) Modes

You can run the PTV utility in the following modes (also known as *testing levels*). To decide which mode to use you must determine the status of program development at your installation and the degree of testing complexity required. A description of each mode follows here.

- **Phase 3 (Package Test)**

  In this mode, more than 1 test unit can be included on a test unit tape (TUT). A test unit can also span more than 1 TUT.

  The PTV utility starts automatically when running in this mode. During each test unit and after each test unit is complete, you can enter live input if you specified live input on the RUNID card.

  No driver is permitted in this mode. Also, the application program interfaces for processing messages in the test unit must be established. After the PTV utility has created the environment, PTV initiates the test by passing your first message to the control program, simulating an input message arriving on a communication line.

  Messages are handled in single or multithread environments, and in burst or asynchronous mode:

  – In *burst mode*, the user-defined number of messages is added to the input list but all messages are allowed to end before a new burst is entered. In a system with the multiple database function (MDBF), messages are processed by the basic subsystem (BSS) or by the subsystem user (SSU). The SSU is determined by the line address or by the prefix on the input message.

  – In *asynchronous mode*, a user-defined number of messages is kept in process until all messages in the test unit have been input. A new message is added to the input list as soon as an *in process* message exits.

- **System Test Vehicle (STV)**

  In this mode, only 1 test unit can be included on a TUT. However, this test unit can span more than 1 TUT. You start and control the PTV utility in this mode by entering the ZSTVS commands. You can enter live input during the unit test.

  In this mode, a full scale system is tested. Messages are handled in a multithread environment. Data and programs must have been loaded with standard system facilities. Data and programs are not allowed on the TUT. In this mode, messages are handled as in the phase 3 (package test) mode, except that the testing mode is specified through a PTV command.

- **Live**

  This mode stops the PTV utility and returns control to the TPF system.

See *TPF Operations* for more information about PTV commands and for procedures you can use to run the PTV utility.

Program and Database Security

At the conclusion of every test run, the PTV utility restores all application programs and data records altered by the test run to their condition before the test. This is not a user option; it is controlled automatically by the PTV utility. However, you have the option to restore or not to restore between test units in a run.

If several test units are run without a database restore, the updates to the database are accumulated. The next test unit to request database restore causes the database to be restored to its original state on completion of the test unit. This allows several test units to be cascaded so that each depends on the results of the preceding test unit. After the final test unit in a package, known as the *phase 3* test run, the PTV utility restores the database automatically as part of deactivation procedures.

Trace Input

The following section discusses the various types of input you can use when running the PTV utility.

Test Unit Tape (TUT)

The preliminary input to the PTV utility is the test unit tape generated by the system test compiler (STC). A *test unit tape (TUT)* is a tape file consisting of 1 or more test units used to test application programs. This tape is created by the STC from the RUNID test units that STC processed as input. Each test unit contains a RUNID to identify the test unit, data records to be loaded, and input messages to be processed.

The primary purpose of this tape is to supply input to the program test vehicle (PTV) utility. The PTV utility uses the TUT solely as input. The TUT is not modified after it is created.

Test Unit Tape (TUT) Contents

The RUNID card is written to the TUT in card-image format to identify the test unit. After the RUNID card is written to the tape, there may be dump options and terminal simulation records written in card-image format. Finally, the existing data records are written to the tape. The records are either called from the standard data/message file or created by the data generation portion of the STC.

The input messages are last in a test unit. The messages are created by the data generation portion of the STC.

Phase 3 (Package Test) Mode: To run the PTV utility in the Phase 3 (package test) mode, provide the following input. See “Program Test Vehicle (PTV) Modes” on page 4 for more information about the Phase 3 (package test) mode.

1. *RUNID record*

   This is an 80-character card-image record.
2. *Dump option* and terminal simulation records, if there are any

   These are 80-character card-image records.
3. *9s record*

   This 20-byte record indicates the end of the Dump option and Terminal Simulation records in this test unit.
4. *Data records*, if there are any

   These are usually 941-, 1615-, or 4655-byte records. The first 560 bytes of the record are used as a prefix to the actual data record. This prefix contains bit settings to indicate record type and ordinal number for file address calculation in the record. The format of this prefix is the same as the format of the SDMF data record prefix. See the SDFPF data macro.
5. *9s record*

   This 20-byte record indicates the end of the data in this test unit.

6. *Message records*, if there are any

   The messages are blocked in variable size blocks. The maximum record size depends on the maximum 3270 message length as defined in the system initialization program (SIP) deck (NETWK statement M3270 parameters).
7. *Two 9s records*

   These two 20-byte records indicate the end of the blocked messages in this test unit.

   The previous steps are repeated for each test unit.
8. *End STC record*

   This 80-byte card-image record signifies the end of the TUT.

System Test Vehicle (STV) Mode: To run the PTV utility in STV mode, provide the following input. See "Program Test Vehicle (PTV) Modes" on page 4 for more information about the STV mode.

1. *RUNID record*

   This is an 80-character card-image record.
2. *Two 9s records*

   These two 20-byte records indicate that the tape does not contain any data.
3. *Message records*

   The messages are blocked in variable size blocks. The maximum record size depends on the maximum 3270 message length as defined in the system initialization program (SIP) deck (NETWK statement m3270 parameter).
4. *Two 9s records*

   These two 20-byte records indicate the end of the blocked messages in this tape.
5. *End STC record*

   This 80-byte card image record indicates the end of the TUT.

Pilot Tape

A *pilot system tape* is similar in format to a test unit tape (TUT) but can contain only data records; no programs or messages are on pilot tapes. A pilot system tape is generated by the STC and can contain more than one pilot system or set of data records. Each pilot system on the pilot tape is preceded by a RUNID record that contains PILOT in position 8–12 and the identification number of the pilot system.

You can have a pilot system loaded for a unit or package test. STC records the request on the TUT and the PTV utility will load the pilot according to the information specified on the RUNID record. See "Pilot Tape Generation" on page 37 for more information about STC generation of pilot tapes.

Online Disk Packs

The *online disk packs* are created by the system loader. They contain the control program, IPL record and bootstrap, keypoints, restart segments, required application programs, and predefined data records.

The PTV utility must be loaded to the online file modules before running PTV. Data records can be obtained from a pilot system, if specified; application programs are obtained from the object library. These packs are formatted according to the structure created by the system allocator and the system configuration (SYSCON) macro.

Test units can make use of any data in the online system or can overlay the data with the TUT input using the online, allocated locations. In either case, the PTV utility always restores the originals at the end of the run.

PTV Control Records

The following section discusses:

- Test unit identification record (RUNID)
- Terminal simulation options
- Dump options (DUMP).

Test Unit Identification Record (RUNID)

The *RUNID record* defines the start of a test unit, the programs to be tested, testing conditions, and any test options. See Table 2 on page 21 for the required RUNID format. The following information is specified on the RUNID record.

Test Unit Identifier: Five characters in position 8–12 that identify the test unit. If the RUNID record identifies a pilot system, the characters PILOT are entered in position 8–12.

Database Restore Indicator: A blank (do not restore) or a D (restore database at beginning of test unit) in position 13. The option is whether to restore the database at the start of the test unit. Database restore is always active for all test units on the TUT.

Pilot System Identification: A nonblank character is required in position 14 if a pilot tape is to be loaded. If position 14 is not coded, a pilot tape is not loaded. If a previous test unit loads a pilot tape, the pilot tape retains all modifications made to it unless the database restore option is active.

User of Pilot Tape: A blank in position 15 indicates the pilot tape for the subsystem user (SSU) pilot tape is loaded. If data is entered in position 15, the basic subsystem (BSS) pilot tape is loaded.

Global Storage Load: Position 16 is a single digit (1–9), which specifies the mode of global storage load.

Screen Reset Indicator: When output messages for a 3270 are postprocessed by the offline diagnostic output formatter (DOF), an output message may lock the keyboard. When this occurs, the DOF expects an additional message to unlock the keyboard. For more information about keyboard lock and how to code position 17 in the RUNID, see “Keyboard Lock” on page 66.

Message Input Mode: Position 25 is either A for asynchronous mode, or B for burst mode.

Number of Messages in System: Position 26 is the number (1–9) of messages to be maintained in the system if position 25 specified A, or supplied in each burst if position 25 is B.

Online Terminal Use: Position 27 specifies L for live test or is left blank. L permits the use of online terminals during unit test and while testing in live or phase 3 (package test) modes.

Running State: Position 28 is either blank for NORM state, 1 for 1052 state, 2 for CRAS, or 4 for MESW.

User Comments: User comments are entered in position 35–80. It is possible to have multiple RUNID records for each input tape. In system test vehicle (STV) mode, the RUNID record is the first record of the tape and serves as a tape identification record. You can only have 1 RUNID record for each input tape while using STV mode. The characters, RUNID (position 2–6), are the minimal required information for the RUNID record.

Position 17 can be used to specify the screen reset indicator. Positions 35–80 are reserved for comments. All other positions are reserved for future use by the PTV utility. All other options are specified through commands. See *TPF Operations* for more information about the PTV commands. The RUNID record is immediately followed by message records.

Terminal Simulation Options

If you require the input and output for certain terminals to be printed in simulated terminal format in addition to the normal output printed by the DOF, terminal simulation records must be included in the test unit.

When running the PTV utility in system test vehicle (STV) mode, the terminal tables assembled in the BMP0 (STPP) segment are used to control simulation.

See “Diagnostic Output Formatter (DOF) Online and Offline Utility” on page 61 for more information about the diagnostic output formatter (DOF). See Table 3 on page 22 for the required format.

The following section explains the types of information entered on the terminal simulation record.

Note: You must provide a separate terminal simulation record for each type of device in a given test unit.

Device Type: There are 11 device types that can be specified in position 2–7.

Device Type	Description
CRTs	Display device (2915 or 4505)
1977	1977 hardcopy
1980	1980-21
1984	1980-24 hardcopy
3277-1	3277 Model 1 (480 character) display device
3277-2	3277 Model 2 (1920 character) display device plus 3284-3 printer
328X-1	3284 Model 1 (480 character) or 3286 Model 1 (1920 character) print, if attached
328X-2	3284 Model 2 (1920 character) or 3286 Model 2 (1920 character) printer, if attached.

Device Address: The address of the device to be simulated is specified in position 16–80. The address can be specified as follows.

Address	Description
LNIATA	The identify of a display device or 1977, 1980-021, or 1980-024 terminals.

Each field contains the line number (LN), interchange address (IA), and terminal address (TA), each expressed as 2 hexadecimal digits. Each 6-column field, except the last, is followed by a comma. A maximum of 8 terminals can be specified for each device type.

LNIATA(CD) The identity of a 3277 Model 1 or 2, a 3284 Model 1 or 2, or a 3286 Model 1 or 2.

Each 10-column field contains the:

- Line number (LN)
- Interchange address (IA)
- Terminal address (TA)
- Control unit (C) of the specified terminals
- Device address of the specified terminals.

The control unit and device address are separated from the LNIATA by the delimiters shown; for example, (and).

Each address is expressed as 2 hexadecimal digits except the control unit and device address, which are each 1 hexadecimal digit. A maximum of 6 terminals can be specified for each type of device.

Note: These LNIATA(CD) terminal addresses are only used for non-SDLC terminals.

LEID The logical endpoint identifier is the address of a terminal attached through the network extension facility (NEF).

The LEID is 3 bytes long and represents the endpoint, which is either the terminal or the function associated with that terminal, which remains with the terminal or the function regardless of a change in the network physical addressing structure.

Dump Options (DUMP)

One or more DUMP records can be included when you require additional output to supplement RTT output. A test unit can contain multiple dump option records with the following restrictions:

- A macro group cannot appear on more than 1 dump option record.
- If a dump option record specifies macro group identifier GRP15, only those macro groups not previously specified in dump option records are dumped.

The contents of DUMP records are analyzed by the PTV utility at the beginning of each test unit. The PTV utility contains 2 tables, one with an entry for each system macro, and the other with space for as many as 50 application program segment 4-byte names. As each DUMP record is read, an indication of the option code is entered in the locations in the macro table corresponding to the macro or macro group in the record. A pointer to the first available field in the program name table is also placed in each macro table entry. The program segment names are then copied from the DUMP record image into successive program name table fields.

Once the DUMP record information has been stored during PTV initialization, it remains dormant until the RTT utility gets control at each macro time during test processing. On each such occasion, the RTT utility determines from its indicators in the ECB whether such output is required for the macro just issued. If both

conditions are satisfied, the requested RTT output is generated, and the PTV macro and program tables are examined to determine whether additional output is required for the particular macro and program.

The dump option code, macro identifier, and program segment name are coded by you, copied to the test unit tape by the STC, and processed by the PTV utility during the test. See “Dump Options (DUMP)” on page 9 for the required format for DUMP input.

The following section explains the information you can enter in the DUMP record.

- Dump Option Code

  The 4 possible dump option codes follow.

Code	Description
CDMP	Activate system error dump after all entries have been serviced. **Note:** The CDMP option code can only be specified once in a given test unit.
DATD	Dump the attached data blocks.
EBDW	Dump the entire ECB and all attached data blocks.
ECBD	Dump the entire ECB.

  When all input messages have been processed and all entries have exited, the PTV utility passes control to the system error routines to produce a dump according to the option entered in position 2–5.

- Macro or Macro Group Identifier

  This field contains a 5-character macro or identifies a macro group with the format, GRP*xx*, where *xx* is 2 digits that signify one of the following macro groups.

Macro Group Identifier	Description
01	BACKC and all ENTER-type macros.
02	DEFRC, DLAYC, and all CREATE-type macros.
03	EXITC, LMONC, MONTC, CIOSC, and WAITC.
04	UNFRC and all FILE-type macros.
05	All FIND-type macros.
06	GETCC and RELCC.
07	RELFC and all Get File Storage-type macros.
08	CRASC, ROUTC, and all SEND-type macros.
09	Real-time tape macros.
10	General tape macros.
11	Miscellaneous and unit record macros.
13	All SON-type macros.
15	Remaining macros.

- Program Segment Name

The 4-character program segment for which the DUMP record is valid. If this field is blank, the DUMP record is valid for all programs. If specific segments are named, each 4-column name, except the last, must be followed by a comma.

Figure 1 shows an example of the program segment name.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 EBDW    GRP04 ABCD,BCDE
 DATD    GRP02 BBCC,CCDD
 EBDW    GRP15 DEFG,EFGH
```

Figure 1. Program Segment Name Example

The GRP04 and GRP02 macro groups are marked for the dump option in records 1 and 2, respectively. When record 3 is processed, all macro groups are checked for dump options because the GRP04 and GRP02 macro groups were marked by records 1 and 2, they are not modified by record 3. Only the remaining macro groups are marked for the dump option in record 3.

If record 3 preceded 1 and 2, all macro groups would be marked for the dump option. Subsequent records 1 and 2 would be ignored.

Input Messages

The STC data generation facility is used to generate the input messages that are activated for the test. They are identified in the input stream by the MSG record, followed by the data generation detail records; this group must be the last input to the STC for a test unit. The PTV utility stores the messages and activates them in sequence using the procedure required for the type of test.

Required Input Sequence

The PTV utility expects to receive the following user-defined records input in this sequence for each type of test:

- Phase 3 (Package Test):
  - RUNID
  - Terminal simulation (optional)
  - Dump option (optional)
  - Data generation (optional)
  - Message generation (optional).
- System Test Vehicle (STV)
  - RUNID
  - MSG.

Sample Job Control Language (JCL) for the Test Unit Tape (TUT)

A sample of job control language (JCL) that can be used to create a test unit tape (TUT) for the STV or phase 3 (package test) mode of testing is detailed in Figure 2 on page 12.

```
//STC  EXEC   PGM=STC,REGION=400K,PARM='SALVER=24,SALSIZ=125000'
//STEPLIB   DD  DSN=LINKLIB,DISP=SHR
//STCDD     DD  DSN=OBJECTLIB.OB,DISP=SHR
//SALTB     DD  DSN=SALTABLE,DISP=SHR
//SYSPRINT  DD  SYSOUT=A
//SYSUDUMP  DD  SYSOUT=A
//SYSABEND  DD  SYSOUT=A
//ISMDD     DD  SYSOUT=A
//LIST      DD  SYSOUT=A
//TAPE0     DD  DUMMY
//TAPE1     DD  DSN=TUT,LABEL=(1,SL),UNIT=(TAPE,,DEFER),
//              VOL=SER=xxxxxx,DISP=(NEW,KEEP),
                DCB=(RECFM=U,BLKSIZE=4656,DEN=4)
//TAPE2     DD  DUMMY
//TAPE3     DD  DUMMY
//DISK01    DD  DSN=PARSRCDS,DISP=OLD
//SNAPDDXX  DD  DUMMY
//CARDIN    DD  *
 LOAD ADDRESSES 1601 TO 4F0B,N,381             20,000 SMALL RCDS
 LOAD ADDRESSES 5000 TO 680B,N,1055      1     20,000 LARGE RCDS
 STC     LOAD  DISKFILE             3380
 RUNID CRS01         A1L1                STV or PHASE 3
 MSG
*1
AM0SG    GSTAR 1.
AM0LIT   ENT   X'010000'.              <------ Terminal I.D.
AM0NP1   ENT   X'6C'.                  <------ ALC NOP Character
AM0NP2   ENT   X'6C'.                  <------ ALC NOP Character
23       ENT   X'7D5AD6115AD1'.        <------ Simulation of Enter key
29       ENT   ZDSYST.                 <------ Message text
         GEND
*2
AM0SG    GSTAR 1.
AM0LIT   ENT   X'010000'.              <------ Terminal I.D.
AM0NP1   ENT   X'6C'.                  <------ ALC NOP Character
AM0NP2   ENT   X'6C'.                  <------ ALC NOP Character
23       ENT   X'7D5AD6115AD1'.        <------ Simulation of Enter key
29       ENT   ZDTIMT.                 <------ Message text
         GEND
*3
AM0SG    GSTAR 1.
AM0LIT   ENT   X'010000'.
AM0NP1   ENT   X'6C'.
AM0NP2   ENT   X'6C'.
AM0TXT   ENT   ZDSYST.
         GEND
*4
AM0SG    GSTAR 1.
AM0LIT   ENT   X'010000'.
AM0NP1   ENT   X'6C'.
AM0NP2   ENT   X'6C'.
AM0TXT   ENT   ZDTIMT.
         GEND
```

Figure 2. Sample JCL for the TUT for Phase 3 (Package Test)

Tape Requirements

The PTV utility requires the following tapes.

Tape	Description
DBR	Database restore (used for package test only)

PDG	Pool directory generation (if required)
RTL/RTA	Real-time log tape (always required)
SDF	Pilot tape (if required)
TUT	Test unit tape (always required)

Test Activation Procedures

Following creation of the online modules, the system is IPLed from the prime module. The PTV utility tests for the number of I-streams active in the TPF system. If more than one I-stream is found to be active, the following message is displayed on the terminal: `NOT IN UNIPROCESSOR MODE - PTV IGNORED`

Processing continues with PTV inactive. You must redefine the number of I-streams active (by entering the ZCNIS command) and reactivate the PTV utility by entering **ZSTVS TEST RESET**. When the system is IPLed the initializer and restart sequence are initiated, and control is passed to the PTV utility. If the PTV utility is operative in normal mode (CPMOPM bit 4=0), the following message is output:

```
IDENTIFY TEST RUN.  PH3, ZSTVS TEST YES
                    STV  ----- ZSTVS TEST STV
                    LIVE ----- ZSTVS TEST LIVE
```

Then, enter one of the following responses:

- For a package test, enter **ZSTVS TEST YES**
- For a system test, enter **ZSTVS TEST STV**.
- For a live test with the PTV utility, enter **ZSTVS TEST LIVE**.

If the PTV utility is operating in STV only mode (CPMOPM bit 4=1), the previous procedure is bypassed and the PTV utility automatically sets itself up in STV mode.

Package Tests

The PTV utility verifies that the TPF system is in test mode. If live output is specified, the PTV utility sets up the required intercepts. If a pilot system is required, the SDF tape is searched for the required PILOT and loaded. Data from the test unit is then loaded and the input messages are stored. The dump option tables required for real-time trace are established. In a package test, control is passed to application programs through commands. At the completion of a test unit, if live was specified, the system continues to operate until a command is received indicating the end of live message input. See *TPF Operations* for more information about the PTV utility. When this message is received, or if live was not specified, IPL simulation is initiated. When control returns to the PTV utility, the operational system is restored. Output from the test goes to the console or to tape (RTL or RTA).

Error Recovery

Individual test units are treated as separate entries under the PTV utility. In the event of a catastrophic error in one test unit, the control program (CPSE) attempts to IPL the system. If this fails, a manual IPL of the prime module of the online system is required. This activates the PTV utility, which processes the next test unit (including database restore if requested and program restore).

System Test

After the TPF system is cycled as many as the desired system state, a command can be entered to initiate STV processing. System overlays are made to perform test functions and simulate output lines. The test unit tape, which only contains messages, is positioned. The messages used to control system tests are found in *TPF Operations*.

Live Test

After the system is cycled as many as the desired operating state, input from the live terminals can be entered. See *TPF Operations* for more information about PTV commands and how to change the PTV state from live to system test. No input from simulated terminals is allowed while running in a live state.

PTV Control Blocks

PTVA is the *file-resident PTV keypoint record*. See the TVDSA data macro for more information. It is modified during all testing phases to indicate type of test unit and options. Fields within PTVA indicate the status of the test unit tape, the status of the pilot tape, names of programs, requested test options, and test phases.

Test Output

Test output is written to the real-time log tape for offline processing by the diagnostic output formatter (DOF). DOF formats PTV information and prints a listing designed to facilitate analysis. Output can include the input/output (I/O) message stream, traced data, terminal simulation output, dumps, and the warning or control messages sent during the test. See “Diagnostic Output Formatter (DOF) Online and Offline Utility” on page 61 for detailed information on how PTV output is formatted and printed.

Figure 3 on page 15 is the input/output message stream generated during a package test run. The direct one-to-one sequence of input and matching output messages indicates messages were processed in single thread mode.

The messages are numbered directly to the right of INPUT MSG NR and OUTPUT MSG NR label. Printable characters are shown on the line with the message number. The hexadecimal representation of nonprinting characters is distributed between the line labelled hex zone and the message line. The LNIATA, 010000, is shown first, distributed as

```
000
100
```

followed by the control characters 6C6C, distributed as

```
66
CC
```

relevant to the 1052.

Input Messages

1. The first input message requests that the ZDSYS command is started to display certain, system information. (The output from the function is shown on the OUTPUT MSG line.)
2. The second input message is an ANSWER-BACK message generated by the TPF system for a hard terminal.

3. The third input message requests that the ZDTIM command is entered.

The remaining input messages result in error messages, along with invocations of the ZDSYS and ZDTIM commands. These messages result from control characters, 7D5AD6115AD1 relevant to the 3270. These erroneous messages can help you evaluate parts of new or modified programs through testing.

```
     HEX ZONE        00066
 INPUT MSG NR.     1 100CCZDSYS+
     HEX ZONE        00061                                          16
 OUTPUT MSG NR     1 100C5CSMP97I 12.09.33 CPU-B SS-BSS  SSU-HPN 5DSYS01I
12.09.33 THE SYSTEM IS IN NORM STATED
     HEX ZONE        000610
 INPUT MSG NR.     2 100C5D
     HEX ZONE        00066
 INPUT MSG NR.     3 100CCZDTIM+
     HEX ZONE        00061                                          1
 OUTPUT MSG NR     2 100C5CSMP97I 12.09.34 CPU-B SS-BSS  SSU-HPN 5DTIM01I
12.09.34 SUBSYSTEM BSS  LOCAL STANDARD TIME+
     HEX ZONE        00066 5 15
 INPUT MSG NR.     4 100CC'AO1AJZDSYS+
     HEX ZONE        00061                                          16
 OUTPUT MSG NR     3 100C5CSMP97I 12.09.34 CPU-B SS-BSS  SSU-HPN 5CVAD20E
12.09.34 ILLEGAL FUNCTIOND
     HEX ZONE        000610
 INPUT MSG NR.     5 100C5D
     HEX ZONE        00066 5 15
 INPUT MSG NR.     6 100CC'AO1AJZDTIM+
     HEX ZONE        00061                                          16
 OUTPUT MSG NR     4 100C5CSMP97I 12.09.34 CPU-B SS-BSS  SSU-HPN 5CVAD20E
12.09.34 ILLEGAL FUNCTIOND
     HEX ZONE        000610
 INPUT MSG NR.     7 100C5D
   END OF TEST UNIT
                                          *** END OF PTV SNA SIM ***
 END OF PTV POST PROCESSOR MESSAGE SIMULATION
 ********************************************************************************
  SS NAME,  SS USER,  DATE AND TIME FROM TOD BSS     HPN   06/14/85  12:11:22
 ********************************************************************************
```

Figure 3. PTV Input/Output Message Stream

Messages

See *Messages (System Error and Offline)* and *Messages (Online)* for more information about the following messages:

- Messages written to the real-time tape during PTV processing. These messages are printed by the diagnostic output formatter (DOF).
- Error messages from processing PTV commands.
- System error messages from PTV processing.

Processing Overview

Figure 4 on page 16 shows an overview of the PTV processing.

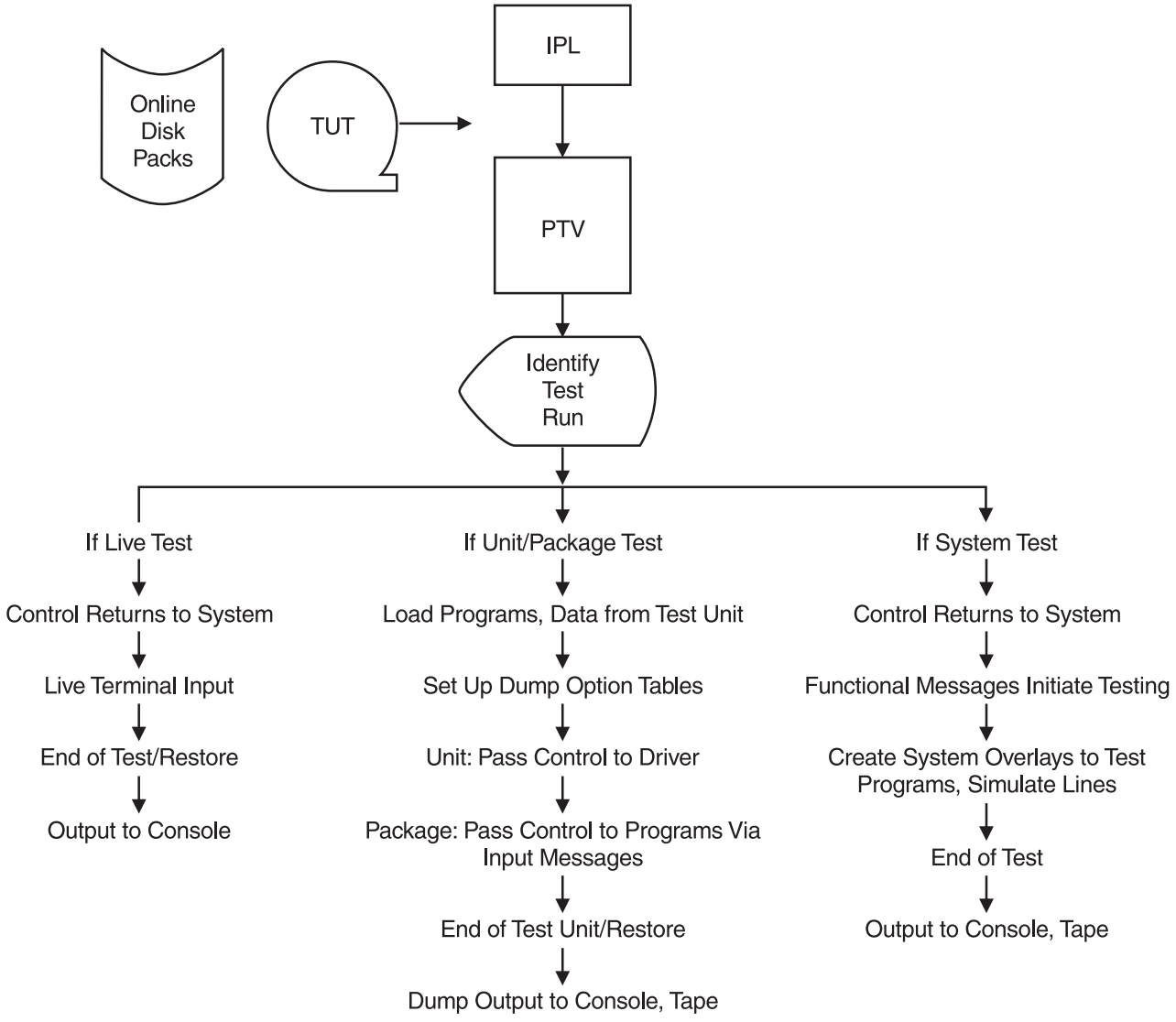

Figure 4. Program Test Vehicle Processing Overview

System Test Compiler (STC) Offline Tape Generation Utility

The system test compiler (STC) is an offline tape generation utility (referred to as the STC utility in the remainder of this book) that:

- Enables you to generate and modify data records and message records
- Generates test unit tapes, pilot tapes, and standard data/message file tapes.

The *data record information library (DRIL)* is a tape file of the record types and data macros selected by you. You can enter or modify data in any field or subfield, within any logical record of a multivolume file that is defined in DRIL. The STC utility accesses DRIL to create or change the records specified by you. If the specified record format is available in DRIL, then you can use symbolic field names instead of absolute displacements. The STC utility determines the displacements and attributes from the symbolic field name. This reduces the possibility of errors because you code fewer statements. To create records for DRIL you must be familiar with the format of a DRIL record, especially columns 59 through 72. Also, column 19 determines the size of the record. The layout of these columns is described in detail in “DRIL Record Layout” on page 18.

Note: A DRIL file is distributed with each release of the TPF system. It is your responsibility to select the DRIL records required for installation and to add new DRIL records. The DRIL file must be created and may be maintained through the IBM MVS IEBUPDT(E) utility program.

Test unit tapes (TUTs) contain one or more test units. Each test unit is identified by a RUNID record and defined by user-coded input to the STC utility. The STC utility writes the data records, messages, and testing options onto the TUT. If data generation is specified, the STC utility will construct data records and/or messages, and record them on the TUT. The TUT becomes the primary input to the program test vehicle (PTV) utility. The PTV utility directs testing according to the information the STC utility recorded on the TUT.

Pilot tapes (SDF) are used to load data records. A pilot tape may contain one or more pilot systems; each pilot system contains a different set of data records, which are defined by the user to satisfy online or testing requirements. Each set of data records can represent a different mode of operation, a different application program, or a different point in time. The pilot tape will be loaded onto the online file modules during system initialization or loaded by PTV to supplement the data used during testing.

The *standard data/message file (SDMF)* is a tape file of predefined data records and input message records. The STC utility uses SDMF to obtain records for pilot and test unit tapes and to modify SDMF itself. Individual records or groups of records can be called from SDMF.

The STC utility also enables you to create, modify, or delete SDMF records. This STC function is known as *standard data message update (SDMU)*. See “Standard Data Message File Update (SDMU) Program” on page 37 for more information about the SDMU function.

DRIL Record Layout

When you create a DRIL record, the DS statement following the record name determines the record size according to the number in column 19. The statement will have this format:

```
MACRO
WAOAA        record name (must be 5 bytes)
DS        OCLx
```

where *x* (column 19) is one of the following.

1 indicates a 381-byte record

2 indicates a 1055-byte record

3 indicates a 1000-byte record

4 indicates a 4095-byte record.

Columns 59 through 72 play an important role in defining records for the data record information library (DRIL). Table 1 defines the meaning of these columns.

Table 1. Column Layout of a DRIL Record

Column	Types & Bytes	Description
59	CL1	• The maximum number of consecutive file addresses that exist starting at the displacement expressed in column 60. • If this column is blank, a zero is assumed.
60	CL1	• The displacement of the first file address. • Use the letters A–Z to represent the decimal numbers 1–26. • If this column is blank, a zero is assumed.
61	CL1	• The maximum number of consecutive file addresses that exist starting at the displacement expressed in column 62. • If this column is blank, a zero is assumed.
62	CL1	• The displacement of the first file address. • Use the letters A–Z to represent the decimal numbers 1–26. • If this column is blank, a zero is assumed.
63	CL1	• The control type, represented by one of the following. **Character** — **Meaning** (blank) — Normal field 1 — Subfield 2 — Slot field • A *normal field* is a field that can only be addressed by using its label. • A *subfield* is a field contained within a larger field. • A *slot field* is a field that contains subfields. Therefore, the entire field can be addressed by using the label of the slot field or only part of the field can be addressed by using one of the subfield labels.
64–65	CL2	• The repetition interval or slot length, that is, the interval at which this item repeats itself. • If 08 is in columns 64–65, then this field is repeated every 8 bytes.

Table 1. Column Layout of a DRIL Record (continued)

Column	Types & Bytes	Description
66–68	CL3	• The maximum number of slots for each physical record. This is related to the repetition interval. The number in this column designates how many times this field can be repeated.
69	CL1	• The field type, represented by one of the following. **Character** — **Meaning** (blank) — No special checking a — Alphabetic field n — Spare field c — Numeric field h — Pointer H field i — Pointer I field j — Pointer J field k — Pointer K field l — Pointer L field m — Pointer M field • The character in this column identifies the field type and causes a check to be made to ensure that only the specified type is used in this field. For example, a C in this column indicates that only numeric characters should be used in this field.
70	CL1	• This field must be blank.
71	CL1	• The field address indicator, denoting where the referenced file address is located. **Character** — **Meaning** (blank) — Not in a file address field. 1 — In a normal file address field. 2 — In a slot or subfield file.

Example of a DRIL Record

The following code is an example of a DRIL record used in the TPF 4.1 system. This representative macro illustrates how columns 59–72 are used. A particular line is referenced by the number in parenthesis to the left of that line. A scale was placed in the code to assist in viewing the card columns.

```
            MACRO                          VERSION=21
            BL0RP
            DS    0CL2    1055 BYTE REC
(1) BL0BID  DS    H       RECORD ID = BL                                A
    BL0RST  DS    H       # OF RESTARTS
    BL0SQC  DS    F       SEQUENCE # OF THIS BLOCK
    BL0CTN  DS    H       # OF ITEMS IN THIS BLOCK
    *                     (BYTE COUNT)
    BL0AGP  DS    H        ACTIVE GROUP FOR RESTART
    BL0ORD  DS    F        LOWEST COMPLETED ORDINAL NUMBER
    *                      FOR RESTART
    BL0RIO  DS    X        ACTIVE DESCRIPTOR ORDINAL NUMBER
    *                      FOR RESTART
    BL0SPA  DS    CL7      SPARE
```

```
    BL0SSN   DS    CL4      SYMBOLIC SUBSYSTEM NAME
    BL0SSU   DS    CL4      SYMBOLIC SUBSYSTEM USER NAME
             DS    0F
    |...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
(2) BL0ITM   DS    0CL20                                  1D1L220052  2
    *                       SEEK ITEM
(3) BL0SID   DS    H          SEEK ID                         120052A
(4) BL0SRC   DS    H          SEEK RCC                        120052
(5) BL0SFA   DS    F          SEEK ADDRESS                1   120052  2
    *                       REFERENCE-FROM ITEM
(6) BL0RID   DS    H          REFERENCE-FROM ID               120052A
(7) BL0RCC   DS    H          REFERENCE-FROM RCC              120052
(8) BL0RFA   DS    F          REFERENCE-FROM ADDRESS      1   120052  2
    *                       MISCELLANEOUS
(9) BL0TYP   DS    X          REF-FROM IS FIXED/POOL          120052
(10)BL0ERR   DS    X          ERROR INDICATORS                120052
(11)BL0SPR   DS    XL2        SPARE                           120052N
             DS    50XL20     50 MORE ITEMS
    BL0LST   DS    XL3        END OF RECORD INDICATOR               N
             ORG   BL0BID
    BL9ITM   DS    XL20
             ORG   BL9ITM
    BL9SID   DS    H
    BL9SRC   DS    H
    BL9SFA   DS    F
    BL9RID   DS    H
    BL9RCC   DS    H
    BL9RFA   DS    F
    BL9TYP   DS    X
    BL9ERR   DS    X
    BL9SPR   DS    XL2
             DS    50XL20
             MEND
```

Line (1)

The A in column 69 indicates that the field type is alphabetic. Though a character in this column is not necessary, it specifies that a check will occur to ensure that only alphabetic characters are used in this field. Therefore, if you were to code an E6 in the BL0BID field, an error would occur.

Line (2)

This line is examined column by column.

Number	Meaning
59	The 1 indicates that there is one file address located at the displacement given in column 60.
60	The D is the displacement of the file address. The file address is located four bytes from this field. (This is because D is the fourth letter of the alphabet.) Notice that BL0SID starts 0 bytes away from BL0ITM; BL0SRC starts 2 bytes away; and, BL0SFA starts 4 bytes away. Therefore, BL0SFA must contain an address.
61 and 62	These two columns function in the same way as columns 59 and 60. Columns 61 and 62 indicate that there is one file address with a displacement of 12. This is at label BL0RFA.
63	The 2 means that this is a slot field.
64 and 65	The 20 is the slot length or the number of bytes that this slot occupies.
66–68	The 052 signifies that this field can be repeated up to 52 times.
69	This field is blank; therefore, the field type will not be checked.

70 This field must be blank.

71 The 2 indicates that the referenced file addresses are within subfields.

Lines 3–11

Lines 3–11 each contain 120052 in columns 63–68. As previously detailed, the 20052 denotes that this field is repeated every 20 bytes for a maximum 52 times. The 1 means that this is a subfield (and is used, for example, in item type cards such as ENTIT). In line 3, the A in column 69 signifies that a check is performed to ensure that only alphabetic characters are used in this field. In line 5, the 1 in column 59 means that one file address is located at the displacement given in column 60. Since column 60 is blank, a default value of 0 is used. This means that the file address is located in this field. The 2 in column 71 means that this is a file address in a subfield.

Required Formats for STC Input

The following section provides the required format for the input the STC utility uses to create your tape output. The information regarding what options are selected for any given input is described in the appropriate function's chapter. For example, the information to code on the RUNID that pertains to dump processing (keylock condition) is described in "Diagnostic Output Formatter (DOF) Online and Offline Utility" on page 61. RUNID information, which pertains to system tests, is described in "Program Test Vehicle (PTV) Online Utility" on page 3.

Disk Allocation Records

Disk allocation records define the set of disk tracks that are formatted with 381 or 1055-byte records on the STC work disk. A maximum of 40 disk allocation cards may be used in a single STC run; 20 for large records and 20 for small records. See *TPF Operations* for more information about the required format.

Disk Type Records

Disk type records identify the type of disk pack available to the STC utility for use as a work disk. See *TPF Operations* for more information about the required format for the system test compiler (STC) utility.

RUNID

The RUNID record identifies the test unit or pilot system that the STC utility will generate. It must be the first record in the input message stream. The required content of the RUNID depends on the type of run:

- Package test
- System test
- Pilot.

For example, the only information required on the RUNID record for a system test is RUNID in positions 2–6. For package tests, the statements following each RUNID define the data to generate and any options for the particular test unit. Subsequent RUNID records mark the beginning of other test units. The last test unit to be recorded on the TUT must have an END STC statement as the last record.

Table 2. RUNID Format for Package Tests

Position	Content	Description
1	Blank	Not Applicable

Table 2. RUNID Format for Package Tests (continued)

Position	Content	Description
2–6	RUNID	Record identification.
7	Blank	Not Applicable
8–12	XXXXX	Test unit identifier or pilot system identifier.
13	X	Database restore indicator.
14	X	Pilot system ID number. If you specify an ID of N, the TPF system can be in any state when you use the ZSLDR command to load the pilot tape. If the ID of the pilot tape is not N, the TPF system must be in 1052 state when the pilot tape is loaded.
15	X	Subsystem pilot tape use.
16	X	Global storage load mode.
17	X	Diagnostic output formatter (DOF) screen reset indicator.
18 – 24	Blank	
25	X	Message input mode.
26	X	Number of messages.
27	X	Online terminal use.
28	X	Running state.
29	Blank	Not Applicable
35–80		User comments.

When loading a pilot tape, see “Pilot Tape Generation” on page 37 for more information about the RUNID record fields.

Terminal Simulation Records

Table 3. Terminal Simulation Record Format

Position	Content	Description
1	Blank	Not Applicable
2–7	X....X	Device type.
8–15	Blank	Not Applicable
16–80	X....X	Device addresses.

Dump Option Records

Table 4. Dump Option Record Format

Position	Content	Description
1	Blank	Not Applicable
2–5	XXXX	Dump option code.
6–9	Blank	Not Applicable
10–14	XXXXX	Macro or macro group identifier.
15	Blank	Not Applicable
16–72	XXXX	Program segment name field.

Data Generation

The STC utility provides a powerful data generation facility. A data or message record can be generated completely from control record input.

The data generation routine (DGR) creates or alters the records specified by the user-defined data generation input. The amount of working storage available to DGR determines how many records can be created with one set of input called *one GSTAR set*. DGR uses whatever working storage is available after all STC programs and required IBM MVS support routines have been loaded. Enough main storage is allocated as working storage to support 50 4K records. Therefore, using one set of input records, that is, one GSTAR record, users may create as many records (of any particular length) as will fit in a 50 times 4K space. If DGR cannot create all the records requested, it will create as many as possible and send a message indicating how many were created. The STC utility inserts delimiters on the output tape to identify the generated data for the loading program.

If a description of the format of the record to be generated is available in the DRIL file, then symbolic field names can be used. The STC utility determines the displacements and attributes from the DRIL record. By using DRIL records, you do not need to code as many statements and much of the required information is defaulted.

The DRIL file distributed to all users contains a variety of different record formats. You must select the necessary DRIL records from the distributed DRIL file that meet the user installation needs or add any user-defined records that are required.

The AMSG records (AM0SG, AM1SG, and AM2SG) should be retained in DRIL, especially if your system contains 3270 or SNA support. By using the AM0SG format to code 3270 and SNA messages, much of the header information is automatically placed in the header and the message length is calculated. If numeric displacements were used to code these messages, each displacement and message length would have to be calculated.

If samples of the desired record type are available on SDMF, then data generation can be simplified further. Each data record or message record sample in SDMF is identified by record type and sequence number in its header section. You can have any sample record from SDMF written to tape. If the SDMF sample does not completely meet your requirements, then the sample can be called from SDMF and modified.

SDMF also contains group records. Each SDMF GROUP record has a unique identification in its header. The GROUP record may contain a variable number of sample records of one or more record types. You can simply specify the GROUP identification instead of specifying the identification for each data record or message record individually.

DATA and MSG Records

Data records and message records are created by the STC utility with the same procedures described here. However, PTV requires that messages and data records be identified and loaded separately.

DATA Records

The DATA record shown in Table 5 must precede all data generation input.

Table 5. DATA Format

Position	Content	Description
1	Blank	Not Applicable
2–5	DATA	Record identification.
6–80	Blank	Not Applicable

MSG Records

The MSG record shown in Table 6 must precede all input for generation of messages. When both messages and data records are being generated, all data record generation input must precede all message record generation input.

If the message is to be generated with an address in the interchange/line address/terminal address (IALNTA) form rather than the line address, interchange, terminal (LNIATA), certain control characters must also be supplied. The first character of text must be a hexadecimal 7A or the last character of text must be a character B or hexadecimal C2. This represents a circle B character which is also used for control. When an input message is coded for SENSE data, it should be coded as SENSE=*xxxx* in position 5–14 of the MSG record.

Table 6. MSG Record Format

Position	Content	Description
1	Blank	Not Applicable
2–4	MSG	Record identification.
5–80	Blank	Not Applicable

GSTAR and GEND Records

A GSTAR and GEND record are used to generate:

- An individual message or data record
- A set of similar records.

One or more detail generation records must be included between each GSTAR and GEND record.

GSTAR Records

The GSTAR record identifies the records to be generated.

Table 7. GSTAR Format

Position	Content	Description
1–5	X...X Blank	Record length, record type, or blank.
6–9	Blank	Not Applicable
10–14	GSTAR	Record identification.
15	Blank	Not Applicable
16–n	X...X	The number of records, SMF record ID, or SDMF GROUP record ID.

Record Length: 1–4 decimal digits, no leading zeroes.

Record Type: Name of the record type if it is included in DRIL. The DRIL name of a record type is the same as the name of the data macro used to define the record in an application program.

Blank: Positions 1–5 are left blank when a GROUP record is called.

Position 16–n:

- Number of Records

  1–4 decimal digits, left-justified, specifying the number of records to be generated in the set, ***followed by a period***, or:
- SDMF record ID

  If a sample of the desired record is available on SDMF, enter SDMF 12-character identification and sequence number, ***followed by a comma***, and number of records (above), or:
- SDMF GROUP record ID

  To call a predefined group of records from SDMF, enter the 12-character identification of the GROUP record, ***followed by a period***.

  Note: When altering a GROUP record this field should contain only 1.

GEND Records

The GEND record marks the end of the data generation records that began with the preceding GSTAR record. A GEND record must precede a new GSTAR set of records.

Table 8. GEND Format

Position	Content	Description
1	Blank	Not Applicable
10–13	GEND	Record identification.
14	Blank	Not Applicable

Data Generation Detail Records

Data generation detail records cause the STC utility to generate fields of data either within the record or in the STC prefix. See “Data Generation Operations” on page 27 for detailed descriptions and examples of each data operand and its parameters.

Table 9. Data Generation Detail Record Format

Position	Content	Description
1–6	X...X	Field location.
7–9	Blank	Not Applicable
10–15	X...X	Data operation code.
16–71	X...X	Field set operands.
72	X	Continuation character.

Field Location:

- Displacement: 1–4 digits that define the displacement of the field (from the first byte of the record) to be generated by this detail record, or

- Symbolic Address: within the STC prefix or within the actual record (if the record is described in DRIL).

Data Operation Code: Table 10 summarizes the operation codes for each data generation operation.

Field Set Operands: Table 10 specifies the required format and explains the field set operands.

Continuation Character:

- Blank:

  If no continuation record follows, position 72 must be blank.

- Semicolon:

  If position 1–6 in the detail record contains a symbolic DRIL address, and if the named field is continuous (field set exceeds 55 characters or 52 hexadecimal digits), then position 72 must contain a semicolon (;). The field set continues from position 16 on the next detail record; position 1–15 must be blank. This is the only situation in which a field set can be divided between two data generation detail records.

  For a continuous hexadecimal field, each continuation record must begin with **X** in position 16–17 and end with an apostrophe (').

  A period (.) delimits the data entered on any one record. If a period does not immediately follow the data to be entered, then the STC utility enters all the data from the beginning of the operand field (position 16) through position 71. This will cause blanks to appear in the data record.

- Any nonblank character (except a semicolon):

  If a detail record with the ENT operation code follows a GSTAR to generate more than one record, then any nonblank character except a semicolon (;) entered in position 72 indicates continuation on position 16 of the next detail record. Positions 1–15 on the continuation detail record must be blank.

  If there is a nonblank character (except a semicolon) in position 72, then the last character entered in position 16–71 must be a comma or a period.

  One record can be initialized with multiple data generation detail records if:

  – More than one record is being generated through the ENT operation code,
  – Position 1–6 in the first detail record contains a symbolic DRIL address
  – The named field is continuous.

  Each detail record should have a semicolon in position 72 except for the last one. This detail record should have a nonblank character (except a semicolon) in position 72 and the required comma or period in position 16–71.

Table 10. Data Generation Detail Records

Data Operation	Operation Code	Format of Field Set Operands
Enter	ENT	V,V,...V
Enter Item	ENTIT	R1-S1-V,V,...,V
Repeat	REP	V1-R1-R2
Repeat Item	REPST	V-S1-S2
Add	ADD	V-D-R1-R2
Add Item	ADDST	V-D-S1-S2
Subtract	SUB	V-D-R1-R2

Table 10. Data Generation Detail Records (continued)

Data Operation	Operation Code	Format of Field Set Operands
Subtract Item	SUBST	V-D-S1-S2

Description of the Field Set Operands:

Operand	Description
V	Data to be entered. It may be character (1–55 characters) or hexadecimal. Hexadecimal data must be preceded by X' and followed by '. If commas or periods are required as imbedded text characters, they must be entered in pairs to distinguish them from field delimiters. Internal processing routines remove one of the pair so that only single commas or periods will appear in the output.
R1	1–4 decimal digits specifying the relative number of the first record in the set affected by the operand. Records are numbered starting with 1.
R2	1–4 decimal digits specifying the relative number of last record in the set affected by the operation.
S1	1–4 decimal digits specifying the relative number of the first item in the record affected by the operation. Items are numbered starting with 1.
S2	1–4 decimal digits specifying the number of the last item in the set affected by the operation.
D	1–4 decimal digits specifying a decimal constant value that is to be converted to hexadecimal and then to be added to or subtracted from the hexadecimal constants of the field or item in successive records in a set.

Data Generation Operations

Enter (ENT V.)

Enter the data (V) in record 1 or in successive records starting with record 1. Multiple record inputs must be separated by a comma (,). The last record input must be followed by a period. If commas or periods are to appear in the output text, they must be entered in pairs. Therefore, if the text *3.25 inches, color blue* is required in a record it must be entered as **3..25 inches, color blue**. The extra comma or period ***will not appear*** in the output. Multiple inputs must not exceed the number of records generated in this GSTAR set.

Figure 5 shows an example of Enter (ENT V.).

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
50       ENT       X'C1C2',1234.
WA0BID   ENT       WA,WA
```

Figure 5. Data Generation Operations – Enter (ENT V.) Example

At location 50, enter hex digits **C1C2** in record 1. Enter characters **1234** in record 2. At field WA0BID, enter the characters **WA** in record 1 and in record 2.

Enter Item (ENTIT R1-S1-V.)

Enter the data (V) in the specified item (S1) in the specified record (R1). See "Enter (ENT V.)" for more information about commas and periods.

ENTIT is only used with DRIL records. DRIL attributes determine whether the data is left-justified or right-justified in the item.

Figure 6 shows an example of the Enter item.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
PD1ITM   ENTIT 1-4-X'084340C1C3'.
PD1NAM   ENTIT 1-21-JOHN SMITH,MY NAME.
```

Figure 6. Data Generation Operations – Enter Item Example

At field PD1ITM, enter the hex characters **084340C1C3** into the fourth item in record 1. At field PD1NAM, enter the characters **JOHN SMITH** into item 21 and the characters **MY NAME** into item 22 in record 1.

Repeat (REP V-R1-R2.)

Enter the data (V) in each record in the specified sequence of records (R1 through R2).

Figure 7 shows an example of Repeat.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
35       REP       X'C1C2'-16-19.
WI0ARC   REP       AB-5-7.
```

Figure 7. Data Generation Operations – Repeat Example

At location 35, enter the hexadecimal characters **C1C2** in record 16, record 17, record 18, and record 19. At field WI0ARC, place the characters **AB** in record 5, record 6, and record 7.

Repeat Item (REPST V-S1-S2.)

Place the data (V) in each item in the specified sequence of items (S1 through S2), in the first or only record in the GSTAR set.

Figure 8 shows an example of the Repeat Item

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
40       REPST X'C1C2'-7-10.
P010AC   REPST NW-2-5.
```

Figure 8. Data Generation Operations – Repeat Item Example

At location 40, place the hex digits **C1C2** in item 7, item 8, item 9, and item 10. At field P010AC, place the characters **NW** in item 2, item 3, item 4, and item 5.

Subtract (SUB V-D-R1-R2.)

Place the data (V) in the first record (R1). Subtract the constant value (D) from the first record and place the balance in the second record. Continue to subtract by the constant value and place the balance in the sequence of records (R1 through R2).

Figure 9 shows an example of Subtract.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
ID2NGL   SUB   X'4F'-4-2-4.
```

Figure 9. Data Generation Operations – Subtract Example

At field ID2NGL, place hexadecimal **4F** in record 2. Subtract 4 from record 2 and place hexadecimal **4B** in record 3. Repeat and place hexadecimal **3F** in record 4.

Subtract Item (SUBST V-D-S1-S2.)

Place the data (V) in the first item (S1), in the first or only record of the set. Subtract the constant value (D) from the first item and place the balance in the next item. Repeat the process until the last item (S2) is reached.

Figure 10 shows an example of the Subtract Item.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
PQ5QGN   SUBST X'7B'-1-1-4.
```

Figure 10. Data Generation Operations – Subtract Item Example

At field PQ5QGN, place hexadecimal **7B** in the first item. Subtract 1 and place hexadecimal **7A** in the second item. Repeat and place hexadecimal **79** in item 3. Repeat and place hexadecimal **78** in item 4.

Add (ADD V-D-R1-R2.)

The data (V) is placed in the first specified record of the set (R1) and incremented by the constant (D) for each successive record through R2.

Figure 11 shows an example of Add.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
IH0IS2   ADD   X'2B'-2-16-19.
```

Figure 11. Data Generation Operations – Add Example

At field IH0IS2, place hexadecimal **2B** in record 16. Add 2 to record 16 value and place hex **2D** in record 17. Repeat and place hex **2F** in record 18. Repeat and place hex **31** in record 19.

Add Item (ADDST V-D-S1-S2.)

Place the data (V) in the first item specified (S1), increment that value by the constant (D), and place the sum in each item in the sequence of items through S2.

Figure 12 shows an example of the Add Item.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
NI0MAP   ADDST X'5B'-3-10-12.
```

Figure 12. Data Generation Operations – Add Item Example

At field NI0MAP, place hex **5B** in item 10. Increment by 3 and place hex **5E** in item 11. Repeat and place hex **61** in item 12.

Data Record Prefix Entries

The STC utility attaches a prefix to every data record written on its output tape. The prefix communicates address information to the loading program. File addresses must always be specified in FACE record type, ordinal number form; these formats are converted to machine addresses by the loading program.

You can generate the prefix as well as the data record itself with data generation detail record input. If the data record is called from SDMF, the SDMF prefix usually contains a load address and the necessary indicators for the loading program. Unless you want to change the prefix contents, the STC utility copies it.

The 4 areas of the prefix that can be generated or altered by using detail record input are:

- Fixed file load address
- Main storage load address
- Imbedded file address
- New record length.

Fixed File Load Address

To retrieve a data record at online test time, the prefix must contain a FACE-type file address that identifies where the record will be loaded. The following describes how to define such an address in a record being created; if the record is called from SDMF and already contains such an address, this section can be ignored. To specify a FACE-type file address, you must use the symbolic field name BSTA06, and either the ADD, ENT, or SUB data generation function.

Figure 13 shows an example of how to create 5 blank inventory detail records.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
PD1PD    GSTAR 5.
BSTA06   ADD   (#PDNRI)10-1-1-5.
```

Figure 13. Fixed File Load Address Example

The FACE-type record address must be written in the form shown: the SYSEQ symbol for the FACE record type (#PNDRI in this example) follows the data operation code, is enclosed in parentheses, and must be 6 characters in length. (Do not code the decimal equate value. You must code the SYSEQ symbol for the FACE record type.) This is immediately followed by the ordinal number of the first record affected by the data operation. In this example, the 5 inventory detail records are created. The STC prefix of each record will contain the system equate for the FACE-type indicator and ordinal number 10, 11, 12, 13, 14, respectively.

Main Storage Load Address

A main storage load address is only valid for data records on unit test tapes. You must specify the absolute main storage address through the symbolic field name. The ADD, ENT, or SUB data generation functions can be used.

Imbedded File Address (ORD Record)

One or more ORD records can be used.

Table 11. ORD Format

Position	Content	Description
1–9	Blank	Not Applicable
10–12	ORD	Record identifier.
16–71	F-L-B(,F-L-B...)	File address information.
72	X	Continuation indicator.

File Address Information:

F The first record within a GSTAR set of records to have the indicator bit set. 1–4 decimal digits which represent the record's number within the GSTAR set.

L The last record within a GSTAR set of records to have the indicator bit set. 1–4 decimal digits represent the record's number within the GSTAR set.

B The displacement of the imbedded file address within the data records. 1–4 decimal digits represent the displacement.

If more file address information is given on another record, then a comma (,) must be the last character entered within position 16–71 on the ORD record. The last character after the last or only F-L-B set must be a period (.).

Continuation Indicator: X must be specified in position 72 to indicate that a continuation of the ORD record begins at position 16 in the next record.

If position 72 is blank, the ORD information does not continue onto another record.

New Record Length (SIZ Record)

The SIZ record overrides the length in the prefix of a DRIL or SDMF record.

Table 12. SIZ Format

Position	Content	Description
1–4	XXXX	Record length.
10–12	SIZ	Record identification.
13–80		Comments.

Record length: 1–4, left-justified, decimal digits specifying the new length of the records to be created.

The GSTAR record sets up the creation of three 1055-byte AAA records. When the SIZ record is processed, the record length for each of the three records is changed to 381 bytes.

Figure 14 shows an example of a new record length.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
WA0AA    GSTAR 3.
381      SIZ
         ...
         GEND
```

Figure 14. New Record Length Example

Group Records

It is possible to create certain SDMF records, called *GROUP records*, that can greatly reduce the number of input statements needed for defining the contents of the STC output tape. With a single GSTAR data generation statement a GROUP record can be called from SDMF that in turn will call many other records from SDMF, just as though those records called were in the input stream. When a GROUP record is called, other records identified within it are placed on the STC

output tape in the section indicated, either DATA or MSG. The procedure to call a GROUP record that would place 3 global records on the STC output tape would be as follows.

If GROUP0000001 contains the following records, then, Figure 15 will place records on the STC output tape:

- GLOBA0000001
- GLOBB0000001
- GLOBC0000001.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
         GSTAR GROUP0000001.
         GEND
```

Figure 15. Group Records Example

A GROUP Record Calls Another Group of Records

The STC utility permits a group record to call another group of records. For example, to place 2 PNR records and 4 PNID records on the test unit tape (these are record types for the airline reservations application program):

If GROUP0000002 contains the following records:

- PD1PD0000001
- PD1PD0000002
- PD1PD0000003
- PD1PD0000004.

and GROUP000004 contains these records:

- PROPR0000002
- PROPR0000003
- GROUP0000002

Then, Figure 16 will place the records on the STC output tape.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
         GSTAR GROUP0000004.
         GEND
```

Figure 16. Group Record Example

Note: If one GROUP record calls another GROUP record, then the GROUP record being called must be the last record in the calling GROUP record.

Figure 17 on page 33 shows an example of records that will generate a group record.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 SDMU    UPDAT
 SDMU    ENTER GROUP0000001, LEON STRAUSS 84147
1055     GSTAR 1.identifier
0        ENT   GLOBA0000001.
12       ENT   GLOBB0000001.
24       ENT   GLOBC0000001.
         GEND
 SDMU    END
```

Figure 17. Example of Records that Generate a Group Record

Figure 18 shows the group record that would be generated.

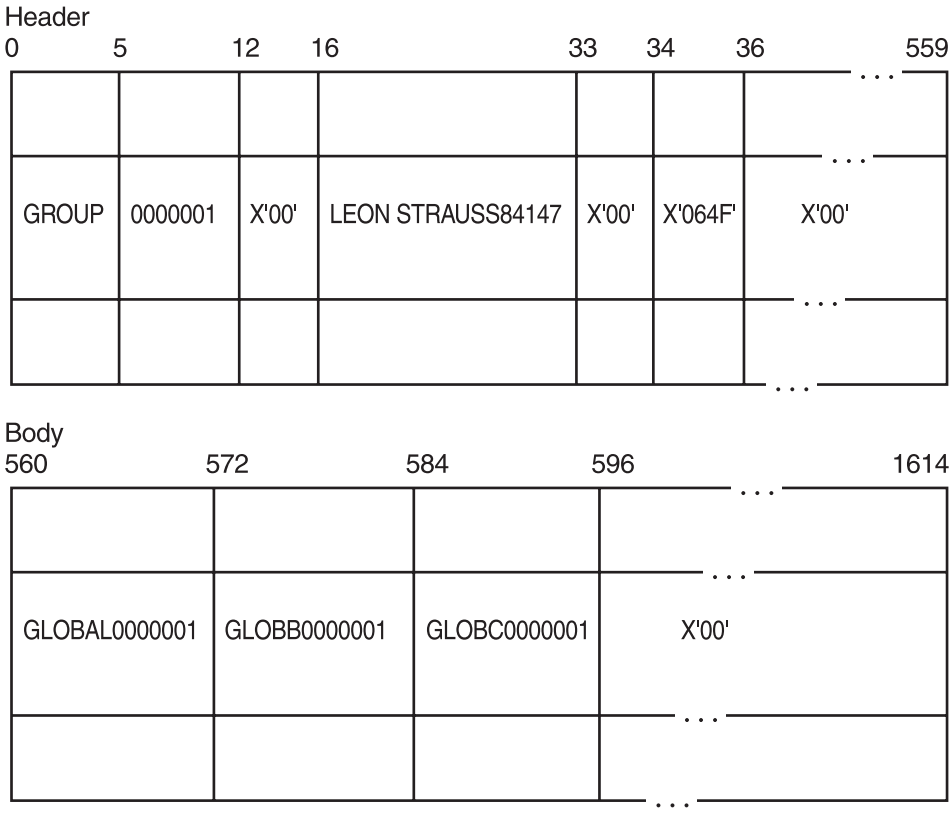

Figure 18. GROUP Record

During the normal functioning of a live system there would be no GROUP records out on file. Therefore, it is not necessary to assign a file address by means of a BSTA06 ... (GROUP) generation statement.

A GROUP record contains the standard SDMF data record prefix plus several 12-character SDMF IDs that refer to any combination of SDMF records.

The GROUP record, like all SDMF records, will have its own unique 12-character ID. The first 5 characters must be GROUP and the remaining 7 characters are used for identification and sequencing. All entries in the data portion of a GROUP record must be 12 bytes long and they must be IDs of SDMF records that are on the SDMF input tape. At least one 12-byte SDMF record ID must be present; the

maximum number of ID'2s is 86. The 12-byte entries must be contiguous, beginning at byte 0. There must be at least 12 bytes at the end of the record that is not used for a data entry.

The GROUP records are treated as ordinary SDMF records in that they reside in sequential order on the SDMF tape with the rest of the SDMF records.

GROUP records are created and placed on the SDMF tape during an SDMU UPDAT run. The GROUP record can be updated on an UPDAT run.

Note: The GSTAR card following an SDMU ALTER card should contain only 1. in columns 16 and 17, with no identifier.

Table 13. GROUP Record Format

	Relative Location				Length				
Level	**Decimal**	**Hex**	**Label**	**B**	**Bytes**	**Bit**	**Format**	**Name**	**Description**
1	0	0	HDR		560			SDMF	Data Record Prefix
2	0	0	GROUP	D	5		D	GROUP	
2	5	5	SEQ. NO.	N	7		D		Sequence Number
2	12	C		N	4		B		Spare
2	16	10	COMMENTS	N	17		D		Comments: (The first 17 characters from the comment field from the SDMU ENTER card creating the GROUP record)
2	33	21		N	1		B		Spare
2	34	22	LENGTH	H	2		B		Length
2	36	24		N	524		B		Spare
1	560	230	DATA	D					Data portion of the GROUP record
2	560	230	RCD ID	F	12		F		Record ID (12 characters)
2	56C	23C	RCD ID	F	12		D		Record ID - Room for up to 85 more 12 character entries
2	—	—	END	F	12		B		End of GROUP record

Examples of Data and Message Record Generation

Manual Generation of Data and Message Records

The following records will generate five 30-byte records, each with the STC prefix. In each case, the prefix will contain a FACE-type load address. The records will contain data in fields beginning at bytes 2, 9, 15 and 25.

Figure 19 on page 35 shows an example of the manual generation of data and message records.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
30       GSTAR 5.
2        ENT   ABC,DEC,X'99',X'EFFE'.
25       REP   123-1-3,X'1122'-4-5.
9        ADD   X'01'-3-1-3.
15       SUB   9-2-3-5,8-1-1-2.
BSTA06   ADD   (#ZZZFS)10-1-1-5.
         GEND
```

Figure 19. Manual Generation of Data and Message Records

Note: On the BSTA06 card, you must code the SYSEQ symbol for the FACE type record (#ZZZFS in this example). Do not code the decimal equate value.

The following data records would be generated.

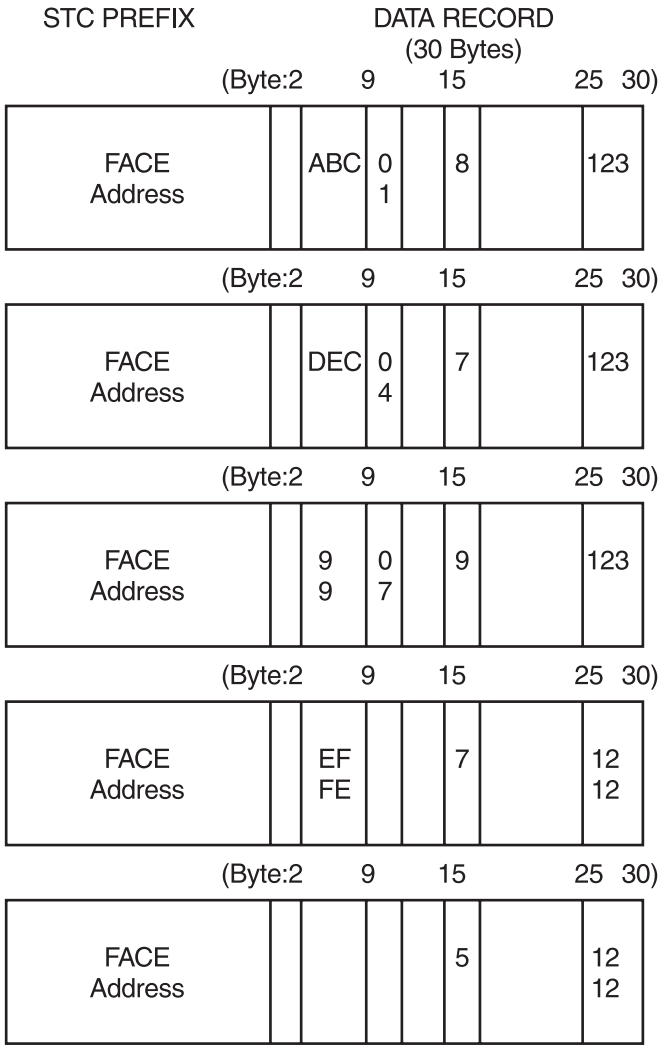

Figure 20. Manual Generation of Data and Message Records

Data Record Creation With DRIL

The following records will generate 2 AAA data records, one TIM record with fixed file addresses specified, and a standard input message.

Figure 21 shows an example of the manual generation of data and message records.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 DATA
WA0AA    GSTAR 2.
BSTA06   ENT   (#ZZZFL)21,
               (#ZZZFL)27.
WA0BID   REP   AA-1-2.
WA0IIS   ENT   THIS FIELD WILL BE ENTERED LEFT.                        ;
               JUSTIFIED IN A 100 BYTE AREA, THIS.                     ;
               FIELD WILL BE ENTERED INTO THE.                         ;
               SECOND RECORD.
         GEND
TI1TI    GSTAR 1.
BSTA06   ENT   (#ZZZFS)99.
TIT1BID  ENT   TI.
TI1TXT   ENT   LOMA.
         GEND
MSG
AM0SG    GSTAR 1.
AM0LIT   ENT   X'080806'.
AM0TXT   ENT   THIS IS A 3270 MESSAGE.
         GEND
```

Figure 21. Data Record Creation with DRIL

Note: On the BSTA06 card, you must code the SYSEQ symbol for the FACE type record (#ZZZFL and #ZZZFS in this example). Do not code the decimal equate value.

Mixed SDMF, DRIL, and Manual Generation

In Figure 22, the following records will cause the WA1AA0000005 and PD1PD0000000 records to be called from SDMF. The first will be placed on the output tape twice, unmodified. The second is modified, using both DRIL and absolute entries, to generate 3 different output records.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 DATA
WA1AA    GSTAR WA1AA0000005,2.
         GEND
PD1PD    GSTAR PD1PD0000000,3.
PD1BID   ENT   AA,BB,CC.
PD1ITM   ENTIT 1-5-X'0380000108325B0000000000003A45'.
PD1NAM   ENTIT 2-7-JOHN DOE.
9        REP   X'0034'-1-3.
         GEND
```

Figure 22. Mixed SDMF, DRIL, and Manual Generation

Examples of Input Messages

Figure 23 on page 37 and Figure 24 on page 37 show 3270 input messages with numeric displacements or with AM0SG format.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
42       GSTAR 1.
7        ENT   X'FF'.      3270 INDICATOR
16       ENT   X'000E'.    MESSAGE LENGTH
18       ENT   X'080806'.  LNIATA
21       ENT   X'6C'.      NOP 1
22       ENT   X'6C'.      NOP 2
23       ENT   3270 MSG+.  TEXT
         GEND
```

Figure 23. 3270 Input Message with Numeric Displacements

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
AM0SG    GSTAR 1.
AM0LIT   ENT   X'080806'.
AM0TXT   ENT   3270 MSG+.
         GEND
```

Figure 24. 3270 Input Message using AM0SG Format

Note: When using AM0SG format to create 3270 messages, AM0TXT must include 3270 screen controls (aid for enter, cursor positioning, set buffer address indicator, and input start position) followed by the actual message text.

Pilot Tape Generation

A pilot tape can be loaded on a TUT to be used during package test. The pilot tape will be loaded before PTV begins to process the test unit if you code this request in the RUNID. The following information should be entered on the RUNID at the appropriate position. Refer to “RUNID” on page 21 for the required format.

Pilot Identifier: The word PILOT in position 8–12 tells the STC utility that a pilot tape will be created.

Pilot System Identification Number: If a nonblank character is entered at position 14, a pilot tape for the basic subsystem is loaded. If position 14 is blank, the basic subsystem pilot tape is ***not*** loaded. However, if a previous test unit loaded a pilot, that pilot remains with all modifications made to it unless the database restore option is activated. The basic subsystem pilot tape ***cannot*** be loaded if the subsystem user tape is loaded.

Subsystem Pilot Tape Use: If a nonblank character is entered in position 15, a pilot tape for the subsystem user is loaded. If position 15 is blank, the subsystem user pilot tape is not loaded. The subsystem user pilot tape ***cannot*** be loaded if the basic subsystem pilot tape is loaded.

Global Storage Load Mode: Position 16 is one digit. Decimal digits 1–9 are selected to indicate the amount of storage that should be loaded. The storage amount that corresponds to the various digits 1–9 are set by you.

Standard Data Message File Update (SDMU) Program

The standard data message file update (SDMU) program enables you to:

- Alter or delete existing records on the SDMF tape
- Create new records on the SDMF tape.

The user-defined control records identify the SDMU function desired. The STC utility will transfer control to the appropriate SDMU program upon reading the input. You can request a dump of an SDMF record, all SDMF records, or a listing of the SDMF directory.

SDMF Generation

When a new SDMF tape is generated, the old SDMF tape must be mounted. Only the records generated during the CREAT run will be written on the new SDMF tape; no old SDMF tape records will be written on the new tape. Up to 5,000 records can be created during one CREAT run.

Input for SDMF Generation

SDMU CREAT: Notifies SDMU that a new SDMF tape will be generated.

Table 14. SDMU CREAT Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	CREAT	SDMU function.
15	Blank	Not Applicable
16–80	Comments	Not Applicable

SDMU ENTER: Indicates a new record will be written on the SDMF tape. The record identification begins at position 16 and must be 12 characters. The first 5 characters are the record name (for example, WA0AA for the AAA), and the last 7 characters are the sequence number.

After the record identification, the first 17 characters following the comma will be considered comments, and are placed in the header of each record created. By convention, the first 12 characters are the programmer name and the next 5 characters are the Julian date (YYDDD). If more than one record is created with the same SDMU ENTER, the sequence number of each record is automatically incremented by one.

Table 15. SDMU ENTER Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	ENTER	SDMU function.
15	Blank	Not Applicable
16–27	Record ID	Not Applicable
28	Comma	Not Applicable
29–80	Comments	Not Applicable

Data: Data generation detail records are used to create new data records or messages. See “DATA and MSG Records” on page 23 for more information about these records.

SDMU END: There must be an SDMU END record to identify the end of input for the SDMU activity (creation, generation, or maintenance).

Table 16. SDMU END Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–12	END	SDMU function.
13–80	Blank	Not Applicable

Sample SDMU CREAT Run

In Figure 25, the SDMU CREAT run creates 4 new records and writes them on the SDMF tape. The new records will have the following record IDs:

- WA0AA0000005
- WA0AA0000006
- WA0AA0000007
- WA0AA0000008.

The record ID sequence number are incremented automatically so that no 2 records in the library have the same ID.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 SDMU    CREAT
 SDMU    ENTER WA0AA0000005,JOHN JONES  66073
WA0AA    GSTAR 4.
WA0BID   REP   WA-1-4.
WACIT    ENT   ALB,NYC,CHI,CIN.
WA0TY2   SUB   X'40'-2-2-4.
         GEND
 SDMU    END
```

Figure 25. Sample SDMU CREAT Run

SDMF Maintenance

You may want to change an SDMF tape after it has been created. Records can be added, deleted, or altered in an SDMU UPDAT run. The old SDMF tape must be mounted. After all the input records are read, for example, after the SDMU END is read, SDMU will load all the new records on the SDMF tape together with all the records from the old SDMF tape that were ***not*** affected by the UPDAT run.

The number of new records added during an UPDAT run should not result in the 5,000 maximum being exceeded; any records exceeding the maximum will be ignored.

Input for SDMF Maintenance

SDMU UPDAT: Indicates to SDMU that an UPDAT or maintenance run will follow.

Table 17. SDMU UPDAT Format

Position	Content	Description
1	Blank	Not Applicable

Table 17. SDMU UPDAT Format (continued)

Position	Content	Description
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	UPDAT	SDMU function.
15	Blank	Not Applicable
16–80	Comments	Not Applicable

SDMU ENTER: See “SDMU ENTER” on page 38 for more information.

SDMU ALTER: Indicates a record from the old SDMF tape will be replaced with an altered record.

Table 18. SDMU ALTER Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	ALTER	SDMU function
15	Blank	Not Applicable
16–80	Record ID Comments	Not Applicable

SDMU DELET: Indicates a record from the old SDMF tape will be deleted.

Table 19. SDMU DELET Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	DELET	SDMU function.
15	Blank	Not Applicable
16–27	Record ID	Not Applicable

Data: See “Data Generation Detail Records” on page 25 for more information.

SDMU END: See “SDMU END” on page 39 for more information.

Sample SDMU UPDAT Run

In Figure 26 on page 41, this SDMU UPDAT runs adds records PD1PD0000007, PD1PD0000008, PD1PD0000009, and GROUP00000012 to the new SDMF tape. WA0AA0000002 will not appear on the new tape. WA0AA0000005 will be the same on the new SDMF tape as it was on the old with the exception that field WA0AGC will now contain AGAA. All other records on the old SDMF unaffected by this update will appear on the new SDMF tape as they existed on the old tape.

Note: On the BSTA06 card, you must code the SYSEQ symbol for the FACE type record (#PNDRI in this example). Do not code the decimal equate value.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 SDMU    UPDAT
 SDMU    ENTER PD1PD0000007,SMITH
PD1PD    GSTAR 3.
PD1BID   REP   PD-1-3.
PD1CHK   ENT   X'04',X'07',X'11'.
BSTA06   ADD   (#PNDRI)30-1-1-3.
         GEND
 SDMU    ALTER WA0AA0000005.
WA0AA    GSTAR WA0AA00000005,1.
WA0AGC   ENT   AGAA.
         GEND
 SDMU    DELET WA0AA0000002.
 SDMU    ENTER GROUP0000012.
36       GSTAR 1.
0        ENT   PG1PG0000001.
12       ENT   PG1PG0000002.
24       ENT   PG1PG0000003.
         GEND
 SDMU    END
```

Figure 26. Sample SDMU UPDAT Run

SDMF Regeneration

It may be desirable to change the SDMF tape if there have been changes in DRIL record formats. During an SDMU REGEN run, the old SDMF tape must be loaded. The REGEN run is similar to the UPDAT run: records on the old SDMF tape can be altered and deleted. However, all the records indicated by the input records will be regenerated before they are written on the new SDMF tape. The number of new records added during a REGEN run should not result in the 5,000 maximum being exceeded; any records exceeding the maximum will be ignored.

Input for SDMF Regeneration

SDMU REGEN: Indicates the SDMF tape is going to be regenerated.

Table 20. SDMU REGEN Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	REGEN	SDMU function.
15–80	Blank	Not Applicable

SDMU ENTER: See “SDMU ENTER” on page 38 for more information.

SDMU ALTER: See “SDMU ALTER” on page 40 for more information.

SDMU DELET: See “SDMU DELET” on page 40 for more information.

Data: See “Data Generation Detail Records” on page 25 for more information.

SDMU END: See “SDMU END” on page 39 for more information.

Sample SDMU REGEN Run

Figure 27 shows a sample SDMU REGEN run.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
 SDMU    REGEN
 SDMU    ENTER PD1PD0000012
PD1PD    GSTAR CREATE,2.
PD1BID   REP   PD-1-2.
         GEND
 SDMU    ALTER NP0CP0000026
NP0CP    GSTAR NP0CP0000026,1.
NP0HTL   ENT   X'02'.
         GEND
 SDMU    END
```

Figure 27. Sample SDMU REGEN Run

In this SDMU REGEN run, if PD1PD0000012 and PD1PD0000013 were already on the old SDMF tape, these 2 records will be regenerated with PD placed in the PD1BID field of each record before they are written to the new SDMF tape. The rest of each record will contain zeroes. Record NP0CP0000026 will be called from the old SDMF tape and a X'02' will be placed in the NP0HTL field before it is written to the new SDMF tape. All other records on the old SDMF tape will be written as they are onto the new SDMF tape.

SDMF Dump Requests

You can request a listing of all records on the old SDMF tape, a specific record, or the directory of all records on the old SDMF tape.

SDMU DUMPD

Dumps a listing of all the records on the old SDMF tape.

Table 21. SDMU DUMPD Format

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	DUMPD	SDMU function.
15–80	Blank	Not Applicable

SDMU DUMPR

Dumps the record with the ID specified in position 16–27.

Table 22. SDMU DUMPR

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	DUMPR	SDMU function.
15	Blank	Not Applicable
16–27	Record ID	Not Applicable
28–80	Blank	Not Applicable

SDMU DUMPT

Dumps all records from the old SDMF tape. The first part of the STC prefix is printed, followed by the record itself.

Table 23. SDMU DUMPT

Position	Content	Description
1	Blank	Not Applicable
2–5	SDMU	Record identification.
6–9	Blank	Not Applicable
10–14	DUMPT	SDMU function.
15–80	Blank	Not Applicable

Required Input Sequence

The STC utility expects input to be provided in a specific sequence. The sequence is determined by the type of tape the STC utility will generate.

Creating One Pilot System

1. Disk allocation record
2. Disk-type record
3. RUNID with pilot identifier
4. DATA
5. Data definition statements
6. END STC.

Creating One Test Unit

1. Disk allocation record
2. Disk-type record
3. RUNID
4. PTV options:
   - Terminal simulation
   - Dump options.
5. DATA
6. Data record generation detail records
7. MSG
8. Message record generation detail records
9. END STC.

Using SDMU Functions

1. Disk allocation record
2. Disk-type record
3. SDMU function
4. SDMU data generation records
5. Data generation detail records
6. SDMU END.

File Input

See *TPF Operations* for more information about the system test compiler (STC) utility and a description of all required file input for STC operations.

Component Programs

The following STC programs are required to create a link-edited, load module STC:

- STC compiler loader (the STCL segment)
- STC compiler control program (the STCC segment)
- STC data generation routine (the DGR0 segment)
- STC input/output routines (the STCI segment)
- standard data message update program (the SDMU segment).

Link-edited with these programs is one segment that is not exclusively used by the STC utility:

- Linkage editor (LEDT) program

File Output

The STC utility produces a listing of all processed input. If errors are encountered during processing, warning and error messages will be printed. Messages are also printed to indicate when input records are written and to indicate if a test unit or pilot has been completed.

Output tape files can be the test unit tape (TUT), the pilot tape (SDF), or a modified standard data/message file tape (SDMF).

Messages

See *Messages (System Error and Offline)* and *Messages (Online)* for more information about the following:

- User abends. The STC utility issues an IBM MVS user abend with a dump when certain errors (such as permanent I/O errors, insufficient main storage) occur. In general, the STC utility issues user abends when processing cannot continue.
- Error messages from the STC loader. The message and the record involved are printed on the same line.
- Error messages issued from the STC control program.
- Error messages issued from data and message generation.
- Error messages issued from the standard data message update (SDMU) program.
- STCC error messages issued by the linkage editor.
- Error messages issued from the STC I/O program.

Processing Overview

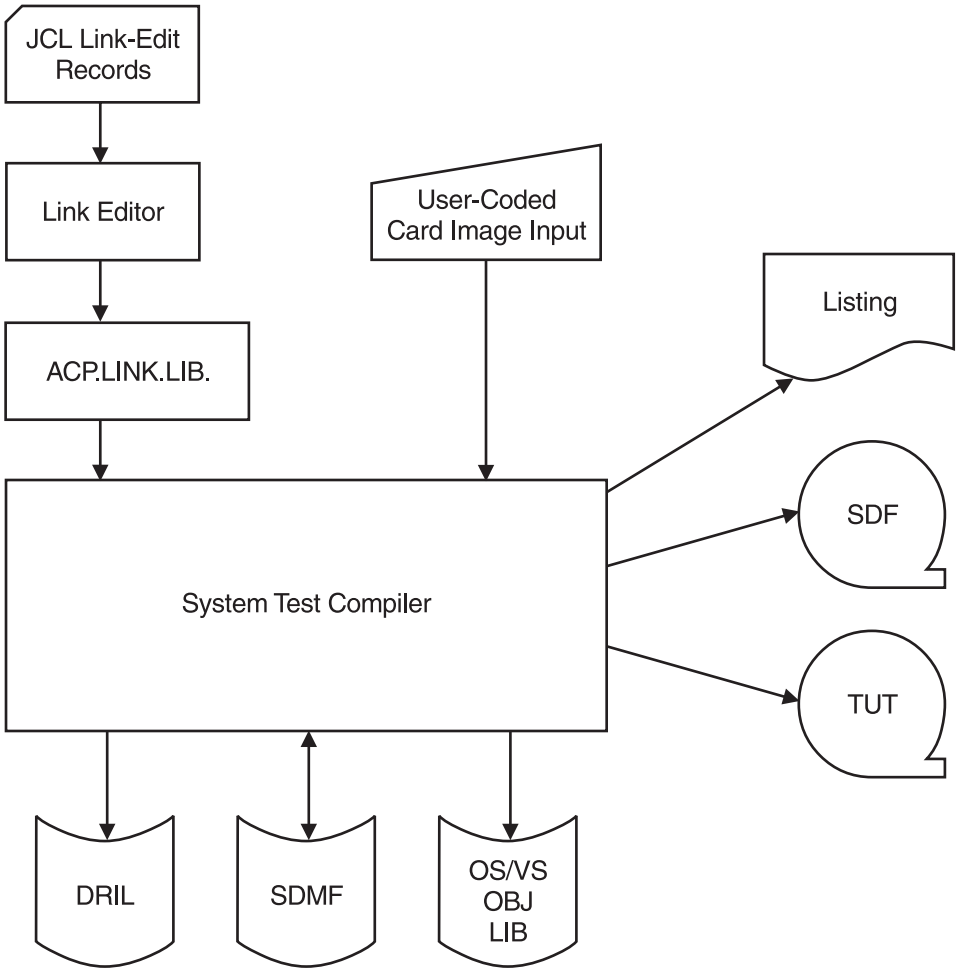

Figure 28. System Test Compiler Processing Overview

Real-Time Trace (RTT) System Utility

Real-time trace (RTT) is a system utility (referred to as the RTT utility in the remainder of this book) that counts or traces the use of specific macros and related system activity. The RTT utility is activated by commands issued from computer room agent set (CRAS) consoles. These commands may be included in the inputs to the program test vehicle (PTV) utility. For the trace option, the RTT utility traces any macros, including fast wait and fast link (non SVC) macros. The amount of trace data collected is determined by the options specified with the activating command. For the count option, the RTT utility creates a list of the selected macros providing a total of executions for each macro during the count period.

When the RTT utility is activated, it dynamically overlays the system code at several points. This enables the RTT utility to get control when:

- The TPF system creates an entry control block (ECB)
- An application program issues a macro
- A system error macro is issued
- At WAIT COMPLETE time for combination I/O macros (for example, FINWC, FIWHC, and Fast Wait).

When an ECB is created, the RTT utility determines whether you requested any macro tracing that could apply to the new entry. If so, the RTT utility sets indicators in the ECB to show the macro whose activity is to be traced in the entry, and the output required when the entry issues such a macro. If the ECB has a message block attached and the trace option is active, then the RTT utility writes a copy of the input message to the real-time tape.

When an application program issues a macro, the RTT utility checks the indicator bytes in the ECB (CE1REC) to see if tracing was requested. If tracing was requested for the entry and the macro just issued is among those to be traced, then the RTT utility collects the output you specified and writes it to tape.

When a system error macro is issued, the system error routines force system error dumps to the real-time tape. Any queued real-time tape output is bypassed. When system error macros are processed, the RTT utility sends a message indicating when the dump occurred so that the out of sequence RTT information can be understood.

At WAIT COMPLETE, the RTT utility intercepts the ECB for combination input/output (I/O) macros (FINWC, FIWHC, and Fast Wait) so that conditions at I/O completion time are recorded.

TPF application programming interface (API) library functions called by dynamic load modules (DLMs) issue TPF macros intercepted by the RTT utility from the macro decoder in the same way as those macros issued from basic assembler language (BAL) programs. However, there is no facility in the RTT utility to trace library function Enters or Exits.

Trace Input

The RTT utility is activated by entering the ZTRAC or ZCNTM command during online processing. Each ZTRAC or ZCNTM command must have a corresponding ZSTOP command. When you want to activate the RTT utility during a test, the address associated with the command must be defined in the system table of CRAS addresses.

See *TPF Operations* for more information about the ZTRAC, ZCNTM, and ZSTOP commands.

The minimum trace output includes the:

- Macro name and parameters
- Address of the macro in a program
- Contents of register 8 (R8) and register 9 (R9)
- Condition code at the time the macro was processed
- Terminal address.

You can optionally collect the contents of all application registers, ECB areas, data blocks, and the user register save areas.

Trace Output

RTT output is a historical record of macro activity that is written to the real-time tape (RTL/RTA) and processed by the diagnostic output formatter (DOF) offline utility. The level of detail in the printed output is determined by the option indicators selected with the activating command.

Example of RTT Output

Figure 29 on page 49 shows an example of the RTT output.

Note: This data record output can be shortened.

```
[A]  HEX ZONE        000
  START RTT       1 100ZTRAC 1FF+
[B] ADDR MACRO  PARAMETER PGID AGENT  I R0       R1       R2       R3       R4      R5       R6       R7
*****************************  THE NEW R8 AND R9 ARE     R8  01B92CF0 R9  0049F000   ******************************
[C] 01A2 FILNC FF01             CYYA 000000 1  FFFFFF00 C3D20000 18480017 00000001 81B92E92 01B92FF0 0021D000
00223E80 0012F5C4 00000004
[D]   LOAD SET NAME : BASE     VERSION: 40
[E]   WORKAREA OF ECB
0049F008 W000 00010000 C2FF0084      W008 00A580A0 006D0000      W016 00000000 00000000      W024 0000001C 01B9B6F0
   BCD                  B                                                                                    0
0049F028 W032 00200000 000077D0      W040 00001000 00002000      W048 0015CF90 0021FC00      W056 000433D0 00000000
0049F048 W064 00000000 00000000      W072 00000000 00000000      W080 00000000 0000007F      W088 00000000 00000000
0049F068 W096 00000000 00000000      WSW1 00000000 00000000
[F]    PORTION OF ECB   FROM DISPLACEMENT 0078
0049F078  FA0 00000000 00000000       FA1 00000000 00000000       FA2 00000000 00000000       FA3 00000000 00000000
0049F098  FA4 00000000 00000000       FA5 00000000 00000000       FA6 00000000 00000000       FA7 00000000 00000000
0049F0B8  FA8 00000000 00000000       FA9 00000000 00000000       FAA 00000000 00000000       FAB 00000000 00000000
0049F0D8  FAC 00000000 00000000       FAD 00000000 00000000       FAE 00000000 00000000       FAF C3D2039B 18480017
0049F0F8  FAP 00000000 00000000
0049F100  CR0 00223E80 0011007F       CR1 00000000 00010000       CR2 00000000 00010000       CR3 00000000 00010000
0049F120  CR4 00000000 00010000       CR5 00000000 00010000       CR6 00000000 00010000       CR7 00000000 00010000
0049F140  CR8 00000000 00010000       CR9 00000000 00010000       CRA 00000000 00010000       CRB 00000000 00010000
0049F160  CRC 00000000 00010000       CRD 00000000 00010000       CRE 00000000 00010000       CRF 0021D000 00D10FFF
0049F180  CRP 01B92CF0 00010000
0049F188  FX0 00000000 00000000       FX1 00000000 00000000       FX2 00000000 00000000       FX3 00000000 00000000
0049F1A8  FX4 00000000 00000000       FX5 00000000 00000000       FX6 00000000 00000000       FX7 00000000 00000000
0049F1C8  FX8 00000000 00000000       FX9 00000000 00000000       FXA 00000000 00000000       FXB 00000000 00000000
0049F1E8  FXC 00000000 00000000       FXD 00000000 00000000       FXE 00000000 00000000       FXF 00000000 00000000
[G]     USER REGISTER SAVE AREA OF ECB
0049F368  URA 00000000 00000000 URB   UR0 00000000 00000000 UR1   UR2 00000000 00000000 UR3   UR4 00000000 00000000 UR5
0049F388  UR6 00000000 00000000 UR7
[F]       USER WORK AREA OF ECB
0049F540 USA0 00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
0049F620      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
[I]         DATA RECORD LEVEL F
        +000  C K039B  B M T 5      +008  B000001 00000022      +010 00000000  1 9 3 6      +018  1 4 3 6 00DDD0 -
        +020 00DD LCC 00DDCF34      +028 00000034 00000064      +030  &A0 D8F 80FC01 U      +038 B0029000 00010000
        +040 000F0000 000001 4      +048 00000005 00780000      +050 00000000 00000000      +058 00000000 00000000
        +060 00000000 00000000      +068 00000000 00 &00A0      +070 00000000 01000100      +078 000003 Y 000000 H
        +080 000001 4 0000015E      +088 000000 . 000000FA      +090 00 .0019 000F0019      +098 000F000A 000A0032
        +0A0 00320032 000A0019      +0A8 0380003C 00010016      +0B0 00020002 00C00010      +0B8 000000 J 00000012
        +0C0 00000000 00000E V      +0C8 00080000 01FF 000      +0D0 006D 000 01B4FFFF      +0D8 00000000 00000003
        +0E0 00000000 00000000      +0E8 00000000 00000000      +0F0 00000000 00000000      +0F8 00000000 00000000
        +1E0 00000000 00000000      +1E8 00000000 00000000      +1F0 00000000 00000000      +1F8 00000000 001B3000
        +200 00510FFF 000000FF      +208 00000000 00000000      +210 00000000 00000000      +218 00000000 00000000
        +220 00000000 00000000      +228 00000000 00000000      +230 00000000 00000000      +238 00000000 00000000
[J] DECB AT 00923020
NAME 00000000 00000000 00000000 CBRW 0092D000 00510FFF FARW FC27FF00 00000000 FC180001 FAXW 00000000 00000000 SUD  00
DATA RECORD ON DECB
        +000 FC27FF00  Q Z Z 4      +008 00000000 00000001      +010  T H0000 00000000      +018 00000000 00000001
        +020 00000000 00000002      +028 000010 H  L H8080      +030 00 BB40E 09257495      +038 56830001  C L 3 1
        +040 00000000 00000000      +048 00000001 00000002      +050 000000 - 03445528      +058 035124C0 03EBA000
        +060 034C5FC0 034AFE08      +068 034951B0 0344E158      +070 00000000  4003001      +078 0001003A 000A065F
        +080 91080000 00000000      +088 0018A010 8000FC44      +090 00000001 034455AC      +098 03445478 05 81008
        +0A0 00010000 02655400      +0A8 02655810 00000000      +0B0 00000000 00000FFF      +0B8 FC440000  C T S X
        +0C0 00000000 00000000      +0C8 07D00000 07D00005      +0D0 00000000 00000000      +0D8 00000000 00000000
        +0E0 00000000 00000000      +0E8 00000000 00000000      +0F0 00000000 00000000      +0F8 00000000 00000000
        +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 00000000

   ⋮
```

Figure 29. RTT Output Example

Description of Alpha Pointers in the RTT Output

The following describes the alpha pointers to key areas in the RTT output.

Alpha Pointer	Description
(A)	Contents of the start RTT message, which is the ZTRAC IFF command. You requested that the RTT utility collect maximum output (1F) for all macros (F).
(B)	This header identifies the trace information on each macro line.

Note: Registers are identified by their numeric label.

(C) Macro output. The CYYA program issued a FILNC macro. Register contents are displayed because they were requested by the output option specified in the ZTRAC command.

(D) Trace output also shows the load set name and version of the program issuing the macro being traced.

(E) Work area of the ECB. Trace output shows the contents of the ECB work area because it was specified in the ZTRAC command. The EBCDIC translation is shown underneath each line, where applicable.

(F) This header identifies the location of the portion of the ECB that was considered relevant to this macro. The relative address of this portion within the ECB (X'78') is shown on the same line as the title.

(G) User register save area of the ECB is shown in hexadecimal with main storage address.

(H) User work area of the ECB is shown in hexadecimal with main storage address.

(I) Core blocks in output are preceded by this header. This header also contains the number of the data level (X'F'). The data block output follows with hexadecimal displacements into the block; where valid, the output is converted to EBCDIC.

(J) Core blocks attached to data event control blocks (DECBs) are preceded by this header. This header also contains the address of the DECB. The data block output follows with hexadecimal displacements into the block; where valid, the output is converted to EBCDIC.

Trace Count (ZCNTM) Output Example

Figure 30 on page 51 shows an example of the trace count (ZCNTM) output.

```
  HEX ZONE        000
START RTT       1 100ZCNTM 1FF+
COUNT OUTPUT                DLAYC            30
                            EXITC           142
                            FILEC             5
                            FINHC             1
                            FINWC            22
                            FIWHC            20
                            GETCC           302
                            RELCC           214
                            SENDCL           79
                            MAXBC             5
                            SENDCC           18
                            UNFRC            22
                            MONTC             1
                            CINFC           138
                            FILNC             2
                            CRETCS            7
                            CREEC             9
                            GETPC             6
                            KEYRC            22
                            CONKC             6
                            SWISC             7
                            ROUTC            97
                            RIDCC            11
                            WTOPC            20
                            EVNTC            10
                            POSTC            10
                            CEBIC            48
                            TIMEC            12
                            CROSC            10
                            UATBC            42
                            CCIDC            18
                            CRATC            99
                            DETAC             9
                            ATTAC             9
                            IGATC             2
                            FACSC           106
                            WAITC             5
                            ENTRC           910
                            ENTNC           276
                            ENTDC            16
                            BACKC           908
                            FLIPC           104
END OF THIS TRACE
```

Figure 30. Trace Count (ZCNTM) Output Example

Processing Overview

Figure 31 on page 52 shows a real-time trace processing overview.

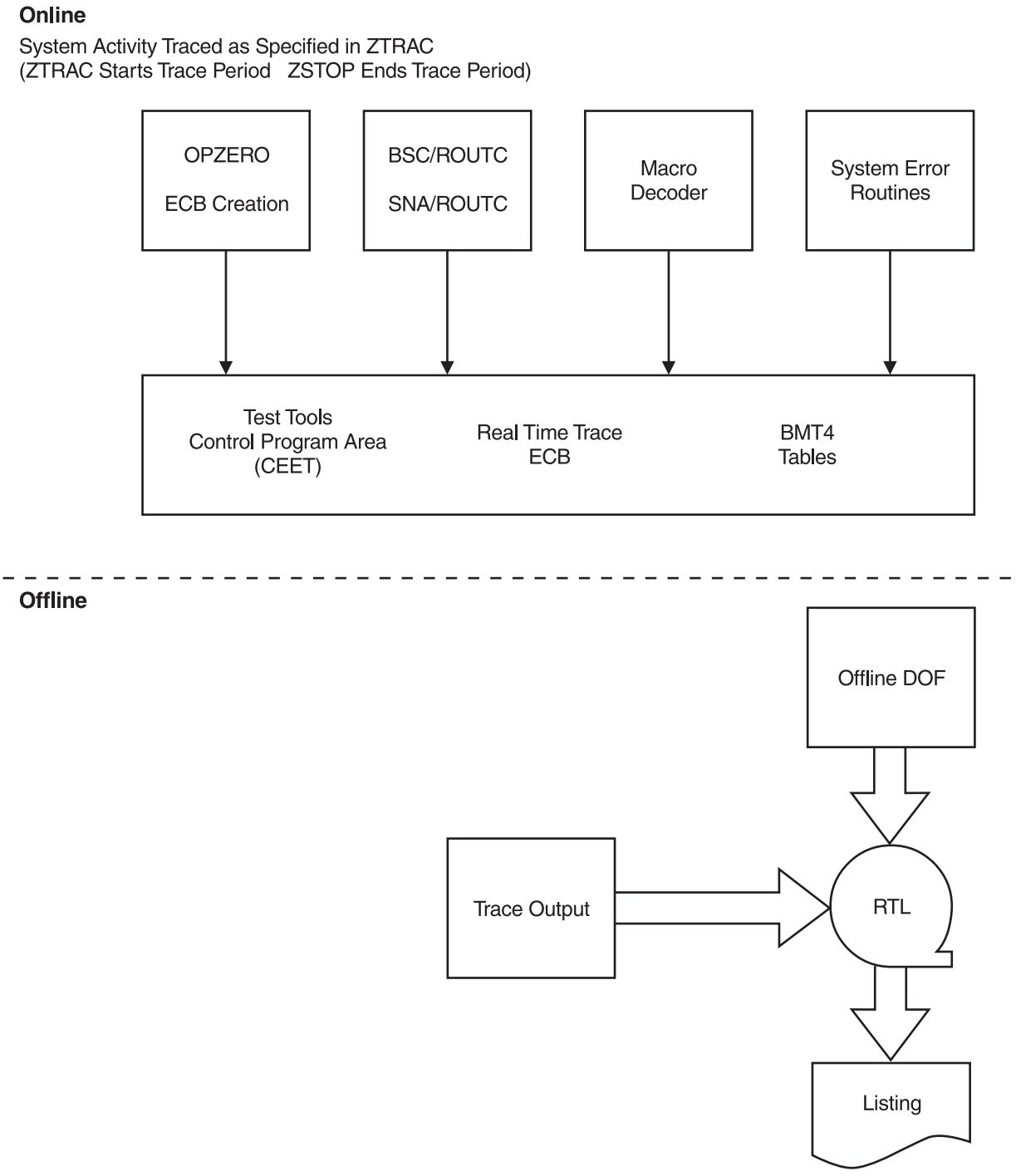

Figure 31. Real-Time Trace Processing Overview

Selective File Dump and Trace (SFDT) Debugging Tools

Selective file dump and trace (SFDT) are debugging tools (also referred to as the selective file trace (SFT) debugging tool and the selective file dump (SFD) debugging tool) designed to help you locate file-related errors during online operations or while testing under the control of the program test vehicle (PTV) utility. Both functions are activated by commands. The commands are entered from a CRAS console or included as part of the PTV input message stream generated by the system test compiler (STC) utility. Although only one function of the SFT debugging tool can be active at one time, the SFD debugging tool can be activated at any time, even while an SFT trace period is active.

Selective File Trace (SFT) Debugging Tool

The SFT debugging tool monitors the updating of selected file addresses during a specified file trace period. A trace period is defined by the following commands:

- ZTRCE
- ZTHLT.

The ZTRCE command specifies the file addresses to be monitored. When the ZTHLT command (halt trace) is entered, the SFT dumps to the real-time tape the monitored records that were updated during the trace period. If a record was marked for monitoring but was not updated during the trace period, the SFT debugging tool does not dump that record to tape. Intermediate contents of a monitored record are not written to tape; only the contents at the end of the trace period are recorded.

The SFT debugging tool has an optional *file address logging* subfunction. If the logging subfunction is requested, the SFT debugging tool provides a list of the addresses of all file records that were updated during the trace period but not specified for monitoring when the ZTRCE command was entered. This log helps check for unanticipated file activity. If a record is filed more than once, its address appears the same number of times in the log. The log is written to tape in addition to the trace output requested with the ZTRCE command.

See *TPF Operations* for more information about the ZTRCE and ZTHLT commands.

Selective File Dump (SFD) Debugging Tool

The selective file dump (SFD) debugging tool writes the contents of specified file records to the real-time tape whenever the ZSELD command is entered. If requested, SFD will also dump all records chained to those specified in the ZSELD command. You can enter the ZSELD command during online operations or included within the input message stream for a PTV test unit. Several SFD requests can be interspersed within the input message stream to locate the point when one or more records are altered. Alternatively, you may only need to verify the contents of a file record at the beginning and end of a test unit.

SFD uses one main storage block to hold the lists of file addresses to be dumped to tape.

See *TPF Operations* for more information about the ZSELD command.

Trace Output

All output from the SFDT debugging tools is written to the real-time tape (RTL/RTA). SFD and SFT output may be interspersed with other test or control program output also on the real-time tape. Use the offline diagnostic output formatter (DOF) utility to process the SFDT output.

Selective File Dump (SFD) Output Example

Figure 32 shows an example of the SFD output.

```
                                  (A)  PAGE  SELD
(C)  START OF SELECTIVE DUMP                        (B)
(D)   SELD RECORD       000000004770000F            FACE REC TYPE ORD NO:  0099   0000000000000001
          +000  Y 277C0  Q A F 4      +008 00000000 00000000      +010  # R B 1  L D          +018 00000001 00000000
          +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038  Y 277C0 4770000F
(E)       +040 00000000 00000000      +048 00000000 00000000      +50  00000000 00000000      +058 00000000 00000000
          +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 000000
      SELD RECORD       0000000047700017            FACE REC TYPE ORD NO:  0099   00000000000002
          +000  Y 277C0  Q A F 4      +008 00000000 00000000      +010  # R B 1  L D          +018 00000002 00000000
          +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038  Y 277C0 47700017
          +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
          +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 000000
(F)  END OF SELECTIVE DUMP

 ********************************************************************************************************************************
  SS NAME,  SS USER,  DATE AND TIME FROM TOD  (G)                        BSS       HPN        08/19/85   10:37:07
 ********************************************************************************************************************************
```

Figure 32. Selective File Dump (SFD) Output Example

An explanation of the alpha pointers to key areas of the SFD output follows.

Alpha Pointer	Description
(A)	Page header. If the PTV utility was active, it will contain PTV RUNID. Otherwise, the header only contains the page number.
(B)	SELD header indicating that this is output from SFD activity.
(C)	START OF SELECTIVE DUMP header printed before the output from each ZSELD command request.
(D)	SELD RECORD header printed to identify each printed record followed by the record's file address, FACE record type, and ordinal number.
(E)	The content of each record specified in the command, at the time the message was processed.
(F)	END OF SELECTIVE DUMP header printed after output from each ZSELD command.
(G)	Trailer that gives the subsystem name, subsystem user (SSU), date, and time (in that order) on the right side of the line.

Selective File Trace (SFT) Output Example

Figure 33 on page 55 shows an example of the SFT output.

```
(A)
(C) START OF SELECTIVE TRCE                                          (B)
(D)   TRCE RECORD       000000007D180007
         +000  A 377C0  Q A C 4      +008 00000000 00000000      +010  # Q B 2  4 D         +018 003A0000 FFFFFFFF
         +020 FFFFFFFF FFFFFFFF      +028 FFFFFFFF FFFF0000      +030 00000000 00000000      +038  A 377C0  '180007
         +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
         +060  F F 9 5 00000000      +068  H P N    9 5 9 6      +070 00000000 00000000      +078 90000000  A 30000
         +080 00000000 0000003A      +088 00000000 00000000      +090  M C 9 5 00000000      +ABS  H P N    9 5 9 6
         +0A0 00000000 00000000      +0A8 90000000  A 30000      +0B0 00000000 0000003A      +0B8 00000000 00000000
         +0C0  D I 9 5 00000000      +0C8  H P N    9 5 9 6      +0D0 00000000 00000000      +0D8 90000000  A 30000
         +0E0 00000000 0000003A      +0E8 00000000 00000000      +0F0  M I 9 5 00000000      +0F8  H P N    9 5 9 6
         +100 00000000 00000000      +108 90000000  A 30000      +110 00000000 0000003A      +118 00000000 00000000
(E)      +120 00000000 00000000      +128 00000000 00000000      +130 00000000 00000000      +138 00000000 000
         +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 000000
      TRCE RECORD       000000007D18000F
         +000  A 377C0  Q A C 4      +008 00000000 00000000      +010  # Q B 2  4 D         +018 00080001 00000000
         +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038  A 377C0  '18000F
         +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
         +060  N E 2 0 00000000      +068  H P N    2 0 2 1      +070 00000000 00000000      +078 90000000  A 30000
         +080 00000000 00000008      +088 00000000 00000000      +090  J D 2 0 00000000      +ABS  H P N    2 0 2 1
         +0A0 00000000 00000000      +0A8 90000000  A 30000      +0B0 00000000 00000008      +0B8 00000000 00000000
         +0C0  J D 2 0 00000000      +0C8  H P N    2 0 2 1      +0D0 00000000 00000000      +0D8 90000000  A 30000
         +0E0 00000000 00000008      +0E8 00000000 00000000      +0F0 00000000 00000000      +0F8 00000000 00000000
         +100 00000000 00000000      +108 00000000 00000000      +110 00000000 00000000      +118 00000000 00000000
         +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 000000
      TRCE RECORD       000000007D180017
         +000  A 377C0  Q A C 4      +008 00000000 00000000      +010  # Q B 2  4 D         +018 00180002 00000000
         +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038  A 377C0  '180017
         +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
         +060  C F 2 0 00000000      +068  H P N    2 0 2 1      +070 00000000 00000000      +078 90000000  A 30000
         +080 00000000 00000018      +088 00000000 00000000      +090  D A 2 0 00000000      +ABS  H P N    2 0 2 1
         +0A0 00000000 00000000      +0A8 90000000  A 30000      +0B0 00000000 00000018      +0B8 00000000 00000000
         +0C0  L B 2 0 00000000      +0C8  H P N    2 0 2 1      +0D0 00000000 00000000      +0D8 90000000  A 30000
         +0E0 00000000 00000018      +0E8 00000000 00000000      +0F0 00000000 00000000      +0F8 00000000 00000000
         +100 00000000 00000000      +108 00000000 00000000      +110 00000000 00000000      +118 00000000 00000000
         +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 000000
(F) END OF SELECTIVE TRCE
```

Figure 33. Selective File Trace (SFT) Output Example

An explanation of the alpha pointers to key areas in the SFT output follows.

Alpha Pointer	Description
(A)	Page header.
(B)	TRCE header indicates printout contains output from the SFT debugging tool.
(C)	START OF SELECTIVE TRCE header precedes output from each ZTRCE command.
(D)	TRCE RECORD header printed for each record, followed by the record's file address.
(E)	The contents of the requested record. Except for the first 8 fullwords, sections of the record that do not contain data are not printed.
(F)	END OF SELECTIVE TRCE header printed after output from each ZTRCE command.

Optional Log Function Example

Figure 34 on page 56 shows an example of the optional log function.

```
                                                 (G)PAGE 125 TLOG
START OF LOGGED FILE ADDRESSES
                      0000000046800003   (H)
                      000000004680000B
END OF LOGGED FILE ADDRESSES
************************************************************************
SS NAME,  SS USER,  DATE,  TIME:     BSS   HPN   04/18/82   07:52:28
************************************************************************
```

Figure 34. Optional Log Function Example

An explanation of the alpha pointers to key areas in the optional log function output follows.

Alpha Pointer	Description
(G)	TLOG header indicates SFT log option was active.
(H)	Output from the log function consists of the file addresses not being traced but which had activity.

Processing Overview

Figure 35 on page 57 gives an overview of selective file dump (SFD) processing showing the process flow for the following:

- ZSELD command
- ZTRCE command
- ZTHLT command
- System interrupt handler.

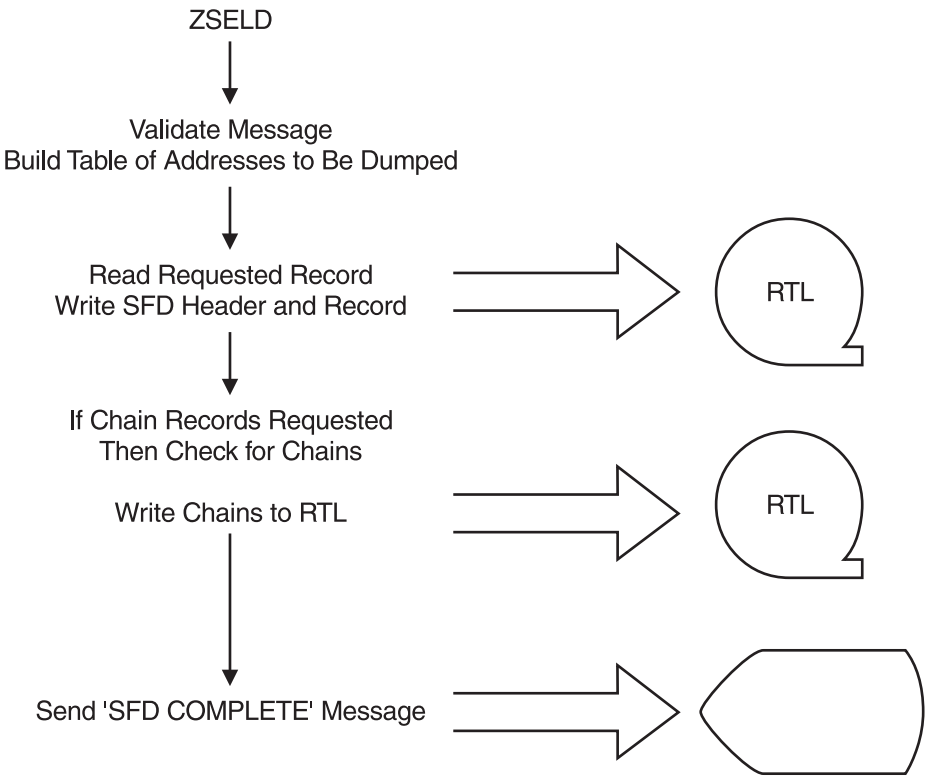

Figure 35. Selective File Dump Processing Overview (Part 1 of 4)

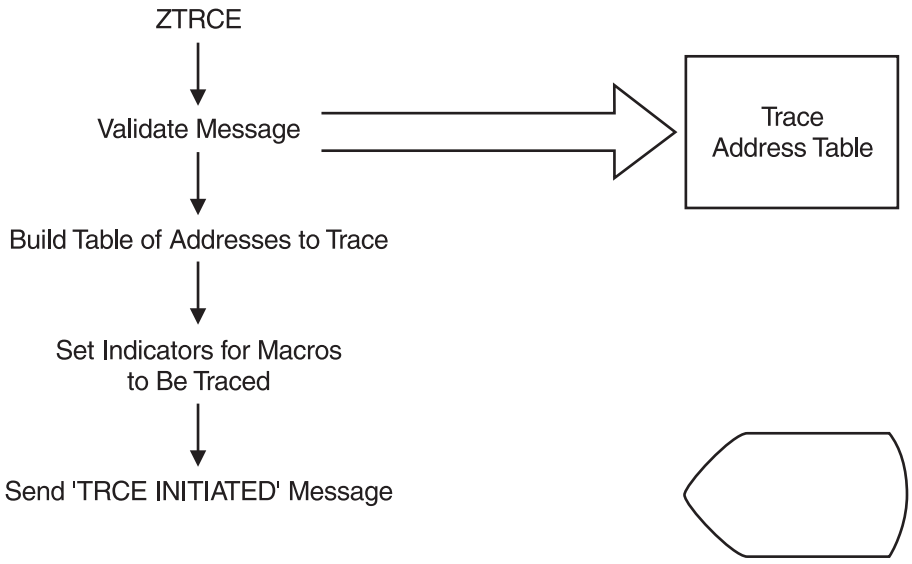

Figure 35. Selective File Dump Processing Overview (Part 2 of 4)

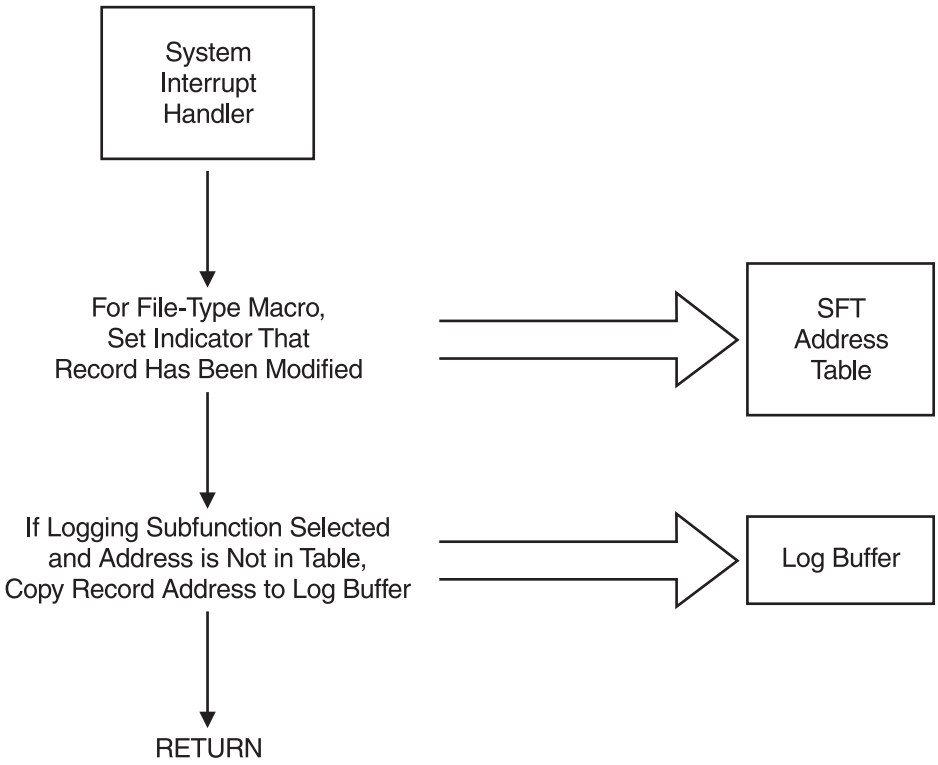

Figure 35. Selective File Dump Processing Overview (Part 3 of 4)

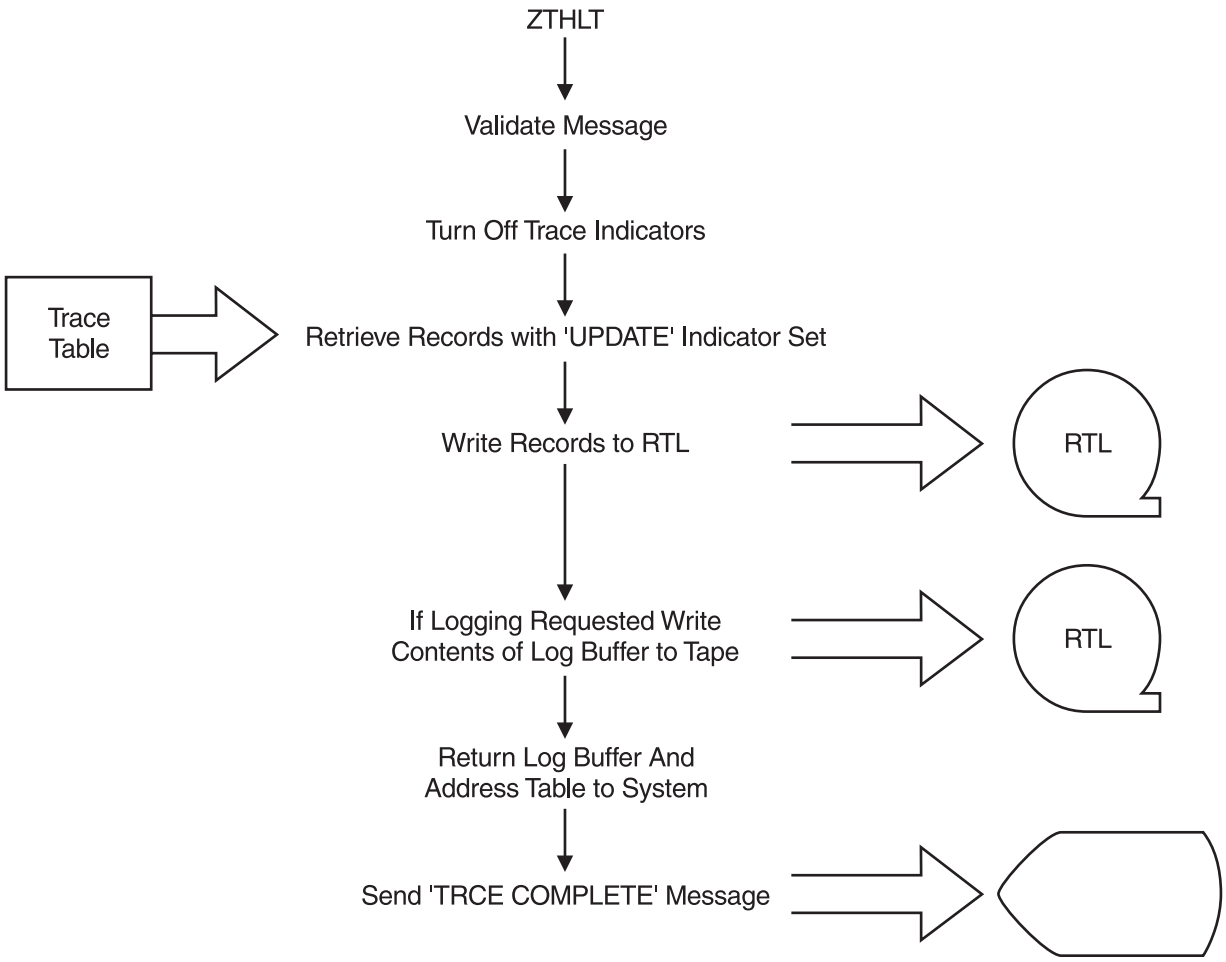

Figure 35. Selective File Dump Processing Overview (Part 4 of 4)

Diagnostic Output Formatter (DOF) Online and Offline Utility

The *diagnostic output formatter* (DOF) utility formats dumps offline and online. The offline part prints a listing from real-time tape data. The online part, also referred to as the *in-core dump formatter* (ICDF), formats and prints system error dumps.

The offline DOF utility formats and prints real-time tape data, such as:

- Output from system error routines
- Trace tables
- Output from the program test vehicle (PTV) utility
- Communication control program trace
- Selective file dump (SFD) debugging tool
- Selective file trace (SFT) debugging tool
- Output from terminal simulation.

The offline DOF utility formats the diagnostic data read from the real-time log tape (RTA/RTL).

The online DOF utility formats and prints system error dumps when the system error routines process a dump record and the write utility determines from system error options that the dump should be printed online, rather than dumped to tape for offline processing.

For both offline and online dump processing, the DOF utility produces a listing of formatted RTA/RTL data that is printed in the following sequence:

1. Real-time trace output, system error dumps, and communication control program (CP) input/output (I/O) trace (in that order)
2. Program test vehicle I/O message stream
3. Selective file dump and selective file trace (SFDT) output
4. Program test vehicle package test output:

   Each individual test unit output, including simulation output, is printed as an entity

   Each user input (that is, options, programs loaded, and data records) is printed at the start of the test unit's output
5. The `END OF TEST UNIT` message is printed at the conclusion of DOF processing.
6. Output from system test terminal simulator (STTS) follows the `END OF TEST UNIT` message.

Both the online and offline DOF utilities format system error dumps and manual dumps with specific headings and storage labels to help you identify particular storage locations. This chapter focuses on how to use these headings and labels to read a dump. For information about:

- System error processing and tailoring dump content, see the *TPF Main Supervisor Reference*
- Displaying and modifying dump options and creating dumps, see *TPF Operations*.

Program Event Recording (PER) Facility

The *program event recording* (PER) facility assists in problem determination for programs running in a native TPF environment. You can trace the following events:

- Storage alteration events
- Instruction fetching events
- Successful branching events (only IBM ESA/390 processors).

Enter the ZSPER command to display PER status, and activate and deactivate PER on IBM ESA/390 processors. The ZSPER command allows you to provide a data parameter to be compared to a particular event you are monitoring in the range you specified. If a match is detected, the PER interrupt is processed. Use the printer option on the ZSPER command to designate an output device that is meaningful to your installation. The value that you specify for this parameter is passed, together with the PER interrupt data, to the UPER user exit. The default output device is the read-only (RO) computer room agent set (CRAS) console.

At interrupt time, the PER facility provides the following set of common data:

- System clock
- Program old PSW
- General registers (ECB virtual memory (EVM) addresses)
- General registers (system virtual memory (SVM) addresses)
- Control registers
- Program interrupt word
- PER code
- PER address
- Active I-stream.

Access to these data items is provided to the UCCPER PER control program (CP) user exit in the CCUEXT CSECT.

User exits provide a means for users of the TPF system to add user-unique processing at various points in the TPF programs, without having to modify the released programs.

Trace Tables

The online and offline DOF utilities format and print the following trace tables:

- Collated macro trace table
- Collated input/output (I/O) trace table
- Systems Network Architecture (SNA) input/output (I/O) trace table
- Path information unit (PIU) input/output (I/O) trace table

Collated Macro Trace Table

The *collated macro trace table* provides a snapshot of system activity by scanning every ECB when a system error occurs. Each ECB in the TPF system maintains its own separate macro trace table. The most recent trace entries are extracted and displayed. The ECB address, macro, program that issued the macro, location where the SVC occurred, parameter information (as appropriate), SVC old PSW information, the SVC code identifying the macro, the SVC index (as appropriate),

and the time stamp. This provides an abbreviated real-time trace of all system activity. The ZSTRC command controls system trace options.

The DOF utility prints the trace table when a dump is issued. The entries are displayed in reverse chronological order, beginning with the most resent. A collated macro trace display is included for each I-stream that was active at the time of the dump. There is no LAST entry. This is ***not*** a wraparound trace table. One entry of the macro trace table is written per line. For a sample of the collated macro trace table see “Collated Macro Trace Table” on page 81.

Collated Input/Output (I/O) Trace Table

The *collated input/output (I/O) trace table* shows the I/O activity at the time an I/O error occurs. The table contains information for each I-stream that was active at the time of the I/O error dump. See “Collated Input/Output (I/O) Trace Table” on page 88 for more information.

Systems Network Architecture Input/Output (I/O) Trace Table

The *Systems Network Architecture input/output (I/O) trace table* contains trace information for SNA I/O interrupts that occur during Network Control Program (NCP) exchange identifier (XID) exchanges, adjacent link station (ALS) XID exchanges, and channel-to-channel (CTC) XID exchanges. Information about significant steps in XID7 processing for CTC devices is also logged in this table. It also contains detailed trace information for NCP, ALS, and CTC asynchronous interrupts. Normal data transfer operations and attention-only interrupts that occur during NCP and ALS data transfer operations are not traced. For more information about the SNA I/O trace table, see the *TPF ACF/SNA Data Communications Reference*.

The SNA I/O trace table is included in system error dumps that have the control program keyword (ICP) specified in the dump override table. The DOF utility prints the SNA I/O trace table when these dumps are issued and labels it with the XID dump tag. See the IOTBL DSECT for information about how to interpret the contents of the SNA I/O trace table.

Path Information Unit (PIU) Trace Table

The *path information unit (PIU) trace table* contains a copy of the data transferred between the TPF system and remote resources. This trace table is included in system error dumps that have the SNA keyword (ISNA) specified in the dump override table.

When system error dumps are issued, the DOF utility prints the PIU trace table and labels it with the PIU dump tag.

See *TPF ACF/SNA Data Communications Reference* for more information about the PIU trace table. See the IPTBL DSECT for more information about how to interpret the contents of the PIU trace table.

Real-Time Trace (RTT) Utility Output

Output from the *real-time trace (RTT) utility* is printed as it is read from the log tape. The trace output is formatted according to the parameters specified with the trace commands that were entered online or read from the test unit tape by the program test vehicle (PTV) utility. The activating command is included in the trace output.

Output from the RTT utility may be interrupted by system error processing. When a system error macro is issued, the system error routines force system error dumps to the real-time tape. When the dump occurs, queued real-time tape output is

bypassed. The SYSTEM ERROR NUMBER *nnn* OCCURRED HERE message, which contains the dump sequence number, is written to tape.

This allows the system error dump to be precisely placed within the traced period. Another real-time trace header is printed before the DOF utility continues to process the trace output after the dump processing is completed.

When it is a catastrophic system error, the catastrophic system error dump may cause the TPF system to enter a WAIT state. After a catastrophic system error dump is completely written, any queued RTT output to the real-time tape is written by the same tape write routine used by the system error routines. Therefore, queued RTT information is not lost when the WAIT state prohibits tape processing.

Communication Control Program (CPP) Trace Routine

The *communication control program (CCP) trace routine* provides a sequential record of CCP macros, start I/O and halt I/O activity, and multiplexor interrupts that occur on one line, or on one or more groups of lines. The CPP trace routine is initiated, modified, and terminated by commands. These messages and the collected CCP trace data are stored in a main storage block. You can optionally use this block in main storage only as a wrap-around log, or when full, to write its contents to tape. If a main storage log only, it is output unformatted whenever a dump occurs. If written to the RTL, the DOF utility produces a formatted list of all items traced including the commands used for the CCP trace activity. On any dump that occurs, the *wrap-around* CCP trace main storage block appears with a record of the last forty activities traced. See the *TPF Non-SNA Data Communications Reference* for more information about CPP trace operations.

The DOF utility formats and prints:

- The CCP command that activated the trace and the time of execution
- The trace block header and time it was logged
- The trace block description header
- The log of communication line activity collected as specified with the ZTHLN command and the macros associated with the I/O activity. See the *TPF Non-SNA Data Communications Reference* for more information about the CCP trace routine.

Program Test Vehicle (PTV) Utility

The output from the program test vehicle (PTV) utility includes:

- I/O message stream
- Warning and control messages
- Dump option request output
- Terminal simulation.

For package tests, each test unit is process as an entity. The diagnostic output formatter (DOF) utility passes through the input/output (I/O) message stream once and repeat the pass for each, if any, terminal simulation options selected for the test unit. If no terminal simulation option records are read on the log tape, then DOF only passes through the message stream once. Messages are identified as input or output and are numbered. This facilitates pairing, particularly in a multithread message test.

DUMP Options

This output is very similar to output from the real-time trace (RTT) utility. The program test vehicle (PTV) utility uses the real-time trace programs to collect the data and write it to the real-time log tape (RTA/RTL).

Terminal Simulation

The DOF simulator programs format input and output messages, using the online printer, as the message would appear on the specified terminal. The printed output from each of the simulators is used as a debugging aid to check message format and content as follows.

- The CRT simulator formats messages within an outlined screen. Format errors are indicated by error messages printed to the right of the outlined screen as they would appear on a 2915 or 4505 CRT.
- 1977 simulator format messages within an outlined border depicting the approximate size of the carriage for these terminals. Format errors are indicated by messages printed to the right of the bordered carriage size outline.
- 1980-21 and 1980-24 simulators format messages using the entire available print line, as that is the size of the carriage of these terminals.
- The 3270 simulator formats messages based on the local 3270 terminal types (CRT or printer). It is designed to simulate 3270 mapping by maintaining the contents of a *terminal buffer* for each terminal. It follows all 3270 commands and orders during simulation. Format errors for the hardcopy 3270 terminals are indicated by messages printed to the right of the bordered carriage size outline. Format errors for 3270 CRTs are indicated by error messages printed to the right of the outlined screen. Format errors also include map violations, invalid commands, and *attributed byte* simulation. For local 3270 simulation, the terminals are identified by the line number, interchange address, and terminal address (LNIATA).

All simulator programs use the same header for each page of printed output. The header contains the following information:

- Simulator identifier

  For example, `PTV 1977 SIM` indicates PTV package test 1977-type simulation.
- RUNID record and date when DOF processed the simulation output
- Page number as sequenced by the current simulator in process.

All simulators also use a message header that contains:

- Terminal identification, which is the Line number, interchange address, and terminal address (LNIATA) for CRT, 1977, 1980-021, 1980-024, and local 3270 terminals.
- Message identifier, which is input or output, except for 1980-021 and 1980-024 terminals where the identifier is IN or OUT.
- Message sequence number as sequenced in the I/O message stream.

Simulated output is sorted according to priority by using the SM option for PTV system test output. This option results in messages simulated according to terminal type. For example, all output to a particular CRT terminal is simulated together even though output to other CRT terminals or other terminal types was dispersed throughout the I/O message stream. See *TPF Operations* for more information about the diagnostic output formatter (DOF) commands and how to specify the SM option.

BMP0 Tables

BMP0 is the terminal simulator control segment. The BMP0 segment contains a table of terminal addresses to be used in system tests. This segment must be created during system initialization, with the addresses for the terminals that are used in system tests at the user's location.

Selective File Dump (SFD) Debugging Tool

The content of records collected by the selective file dump (SFD) debugging tool on the real-time log tape are formatted for printing by the diagnostic output formatter (DOF) utility. The DOF utility writes the `START OF SELECTIVE DUMP` heading at the beginning of output from each selective file dump request and the trailer `END OF SELECTIVE DUMP` at the end of the particular dump request output. The DOF utility identifies every printed file record with the `SELD RECORD` heading or the `SELD CHAINED RECORD` heading and the record file address.

Selective File Trace (SFT) Debugging Tool

The DOF utility formats and prints the file address and final contents of each ZTRCE-specified record that was updated during the defined trace period. Any intermediate updates that occurred are not recorded. DOF writes the `START OF SELECTIVE TRCE` heading and the trailer `END OF SELECTIVE TRCE` at the start and end of each trace period's output. This process is repeated for each trace period recorded on the real-time tape.

Optional Logging Function

If you requested the optional logging function, the DOF utility prints the logged information after the last selective dump or selective trace file in each test unit has been printed. The start of a log for a test unit is identified by the `START OF LOGGED FILE ADDRESSES` heading. The trailer, `END OF LOGGED FILE ADDRESSES`, marks the end of the log for that test unit. If multiple ZTRCE commands with the log function active are entered, each pair of ZTRCE and ZTHLT commands would produce a separate log.

Keyboard Lock

When the DOF utility is processing output messages for a 3270 terminal and the output message locks the keyboard, then the DOF utility expects an additional output message to unlock the keyboard. You can use the screen reset indicator in the RUNID card to handle keyboard lock conditions during tests.

Position 17 determines what action the DOF utility takes when keyboard lock occurs during package tests. If position 17 on the RUNID is blank, the DOF utility assumes the CLEAR key was used and assumes the current screen is complete. The current screen is printed and the area cleared to prepare for the next output message. If position 17 is R, then the DOF utility assumes the RESET key was used and does not assume the current screen is complete. The RESET option is always in effect when keyboard lock occurs during system tests.

Description of a Main Storage Dump

This section shows the format and content of a typical main storage dump. As each section of the dump is presented, it is followed by an explanation of the headings and labels that appear in that section of the dump.

Cover Page

Figure 36 shows an example of a cover page.

```
                                                                                                    DUMP 1623       PAGE     1
SYSTEM ERROR NUMBER   CTL-I000004   VIRTUAL STORAGE ERROR IN OP                  DATE - 16AUG      TIME - 19.53.03
                                    CAUSE OF ERROR - PAGE TRANSLATION EXCEPTION
*APPENDED CONSOLE MESSAGE  -  PSW    071D2000 801A4A70
                              R0-R7  00000000 0030F000 00319C00 00319A80 00010000 00000029 00306390 0030639A
                              R8-R15 001A4A10 00306000 000070B0 00001000 00002000 0016BB90 00328000 801A4A18

                                       THIS PRODUCT CONTAINS 'RESTRICTED MATERIALS OF IBM'
                                       5748-T14 (C) COPYRIGHT IBM CORP 1979, 1993
                                       LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
                                       REFER TO COPYRIGHT INSTRUCTIONS FOR  NUMBER G120-2083
*************************************************************************************************************************
                                  TRANSACTION PROCESSING FACILITY 4.1 PUT LEVEL nn
*************************************************************************************************************************
*LOGICAL CPU ID - B   *SUBSYSTEM NAME - BSS   *SUBSYSTEM USER NAME - HPN

*STATE OF FAILING SS          1052
*SYSTEM ERROR WORD            N/A
*SYSTEM ERROR OPTIONS         LONG  OPR DUMP
*DUMP KEYWORDS                ICAPT  ICNFTBL  ICPSE  ISMDT  ICPLKMP  IISG1  IISG2  IISG3  IIUG1  IIUG2  IIUG3  IMFST  IDCLS
                              IEVMDAT
*SYSTEM TRACES ACTIVE         MACRO  ENTER  SYSLOG  I/O  BRANCH

*CPU ID                       FF1144823090
*SIZE OF SVM                  0020 MB
*NO. OF ACTIVE I-STREAMS      03
*FAILING I-STREAM             03
*V EQUAL R MODE               NO
*BLK CHECK MODE               NO
*ECB PRIVATE AREA             00300000 TO 00400000
*ERROR OCCURRED IN            EVM
*FAILING PROGRAM              CVZZKW  +0060  LOADSET-BASE

*STORAGE PROTECTION KEYS
00000000 TO 000C8FFF - KEY IS F
000C9000 TO 000F7FFF - KEY IS E
000F8000 TO 000F9FFF - KEY IS C
000FA000 TO 000FCFFF - KEY IS 1
000FD000 TO 00104FFF - KEY IS C
00105000 TO 00108FFF - KEY IS 1
00109000 TO 0010DFFF - KEY IS C
0010E000 TO 00156FFF - KEY IS F
00157000 TO 00158FFF - KEY IS 1
00159000 TO 00173FFF - KEY IS F
00174000 TO 00254FFF - KEY IS 0
00255000 TO 004F3FFF - KEY IS  VARIABLE - 4KB COMMON/FRAMES/ECBS
004F4000 TO 0110FFFF - KEY IS F
   .           .        .
   .           .        .
   .           .        .
01347000 TO 0134AFFF - KEY IS C
0134B000 TO 0134EFFF - KEY IS 1
0134F000 TO 01356FFF - KEY IS C
01357000 TO 0135AFFF - KEY IS 1
0135B000 TO 01366FFF - KEY IS C
01367000 TO 01372FFF - KEY IS 1
01373000 TO 0137AFFF - KEY IS C
0137B000 TO 013EDFFF - KEY IS F
```

Figure 36. Dump Cover Page

Table 24 explains the headings and labels that appear in this section of the main storage dump.

Table 24. Headings in the Main Storage Dump for the Cover Page Section

Heading	Description
*DUMP-*xxxx*	This item in the dump header line contains the dump sequence number (decimal 1623 in this case). The sequence number indicates when a dump was issued in relation to a series of dumps.

Table 24. Headings in the Main Storage Dump for the Cover Page Section (continued)

Heading	Description
*SYSTEM ERROR NUMBER CTL-*xxxxxx* or OPR-*xxxxxx*	This heading includes the prefix and the hexadecimal system error number (*xxxxxx*). Together they identify the error that was detected by the control program (CP) or an operational program. The heading shown in the example dictates that the CP detected an I000004 error, which is a page-translation exception in an operational application program. The prefix I indicates that this is an IBM-detected error. See *Messages (System Error and Offline)* for more information about the system error numbers for IBM errors. The heading MANUAL DUMP is used rather than the heading shown previously when a system error dump is forced by an operator. See *TPF Operations* for more information about the ZDUMP command.
*DATE-*ddmmm*	This heading shows the TPF day and month date of the system error dump.
*TIME-hh.mm.ss	This heading contains the value of the TPF 24-hour local time clock when the dump was taken. The time is provided in hours, minutes, and seconds (decimal).
*CAUSE OF ERROR	This heading describes the reason for the system error. It only appears in the case of hard errors (for example, not software initiated).
*APPENDED CONSOLE MESSAGE	An appended message to a system error is an additional informative message provided by the coder of the SERRC macro, and displayed on the prime computer room agent set (CRAS) as well as in the dump. In the case of hard errors such as a page-translation exception, the appended message contains the program status word (PWB) and failing program's registers at the time of the error.
*LOGICAL CPU ID	The TPF logical central processing unit (CPU) ID of the failing processor in the loosely coupled complex of a high performance option (HPO) configuration.
*SUBSYSTEM NAME	The multiple database function subsystem name associated with the failing entry control block (ECB) in an HPO configuration.
*SUBSYSTEM USER NAME	The multiple database function subsystem user name associated with the failing ECB in an HPO configuration.
*STATE OF FAILING SS	The state of the multiple database function subsystem in which the error occurred. If state change was in progress, this indication expands to show the cycle-from and cycle-to states.
*SYSTEM ERROR WORD	The SERRC macro expansion appears here on software initiated errors.
*SYSTEM ERROR OPTIONS	The ZASER options in effect when the dump was taken that have an effect on dump content.
*DUMP KEYWORDS	The list of storage keywords identifying the large areas of storage that do not normally appear in system storage dumps but are included in this dump. See the *TPF Main Supervisor Reference*.
*SYSTEM TRACES ACTIVE	The system traces active at the time of the dump. The trace options are controlled online by entering the ZSTRC command.
*CPU ID	The hardware CPU ID of the executing processor.
*SIZE OF SVM	The size of the system virtual memory (SVM) on the main I-stream, expressed in megabytes.
*NO. OF ACTIVE I-STREAMS	The number of active I-streams is a function of the hardware configuration, plus any ZCNIS commands that have been issued.
*FAILING I-STREAM	Identifies the I-stream on which the failure occurred. The status information for this I-stream appears first in the dump.

Table 24. Headings in the Main Storage Dump for the Cover Page Section (continued)

Heading	Description
*V EQUAL R MODE	Indicates whether the TPF system is running in VEQR mode or normal mode.
*BLK CHECK MODE	Indicates whether the TPF system is running in block check mode.
*ECB PRIVATE AREA	The starting and ending ECB virtual addresses of the ECB private area.
*ERROR OCCURRED IN	Indicates whether the error occurred in the SVM or in an ECB virtual memory (EVM).
*FAILING PROGRAM	If an ECB was associated with the error, the following information is displayed: • Name and version of the failing program. If the TPF system was not generated with support for tracking program versions, the version of the failing program is not displayed. • Displacement into the listing where the error occurred. • Name of the loadset in which the failing program is contained. If the failing program is not contained in an E-type loader loadset, it is considered to be a base program and the word BASE is displayed. If there is no ECB associated with the error, then N/A is displayed.
*STORAGE PROTECTION KEYS	During processing of a system error dump, the storage protection keys are dynamically captured and displayed in the dump. When compared with the storage layout illustrated in the *TPF Main Supervisor Reference* this table provides an indication of where the various storage areas reside in the online system.

Dump Label Index

Figure 37 on page 70 shows an example of the dump label index.

Note: If you are dumping multiple subsystems, the addresses are in ascending order.

```
                        DUMP LABEL INDEX                                    DATE - 16AUG      TIME - 19.53.03
    LABEL   I-S   ADDRESS
-------------------------------------------------------------------------------------------------------------------------
     ALS          01166000
     ANT          00154C40
     BBT          001192C0
     BIT          011BD000
     BSA          00154B20
     BVT          001505D0
     CBT          00528000
     CCB          01209000
     CCC          00150748
     CCT          00014ECE
     CDB          012293E0
     CIO          013DB000
     CLK          00001DB0
     CLV          00000BBC
     CMB          00255000
     CNF          0000112C
     CPY          00000C78
     CRS          0014FE40
     CST          0013B6D0
     CTL1    01   013D9000
     CTL2    02   013DA000
     CTT          0014F9A8
     CWA          0122A148
     CWP          00150510
     CYA          0010F010
     DAA          0121B000
     DCL1    01   0059F000
     DCL2    02   005A1E00
     DCL3    03   005A4C00
     DCR1    01   0059E000
     DCR2    02   0059E080
     DCR3    03   0059E100
     DK1          00255000
     DK2          00255420
     DNT          01227660
     DSD          00000C08
     ECB          003C8000
     ECT          00538600
     FCA          0010F050
     FCT          00528E00
     FC0          013AF000
     FHD          0012B270
     FRM          00300000
     FRT          0014ED30
     FS0          0013DB38
     FS1          0013DD20
     FS2          0014D120
     GFN          01227178
      .             .
      .             .
      .             .
```

Figure 37. Dump Label Index

Table 25 explains the headings and labels that appear in this section of the main storage dump.

Table 25. Headings in the Dump Label Index

Heading	Description
*DUMP LABEL INDEX	This is an alphabetically sorted index of all dump labels that appear in the system error dump. Multiple addresses are listed for labels that are I-stream or subsystem unique. See “Main Storage Dump Labels” on page 112 for more information about the individual dump labels.

I-Stream Status Display

Figure 38 on page 71 shows an example of the I-stream status display.

```
*IS STATUS DISPLAY    IS-0001    CPUADDR-0000

*GENERAL REGISTERS
  0-7     R0  00000000 000619F0  R1    R2  00000200 00061A50  R3    R4  00007D98 00007D9C  R5    R6  00002280 00000002  R7
  8-15    R8  001A4A10 004D6000  R9    R10 00064F38 00001000  R11   R12 00002000 00159F90  R13   R14 000619F0 00061120  R15

*CONTROL REGISTERS
  0-7     C0  1CB0BEE0 0059300F  C1    C2  00000000 70000000  C3    C4  00000000 00000000  C5    C6  80000000 0113800F  C7
  8-15    C8  0000FFFF 00000000  C9    C10 00000000 00000000  C11   C12 813DFF24 0113800F  C13   C14 D7000000 00000000  C15

*FLT PT REGISTERS
  0-6     FR0 00000000 00000000        FR2 00000000 00000000        FR4 00000000 00000000        FR6 00000000 00000000

*CURRENT PSW AT TIME OF DUMP - 070EC000 800616B8

*PSWS               OLD                   NEW            INTRPT CODE
*   RESTART   00000000 00FF0000   000C0000 00000F80
*   EXTERNL   070EC000 800616B8   040CC000 8000C420   00000000 00021202
*   SVC       071D0000 801A4A30   050C0000 8001F200   000200B6
*   PROGRAM   071D0000 8000307C   000CC000 80034008   00040040
*   MACHINE   00000000 00000000   0008C000 8002F000   00000000 00000000
*   I/O       070EC000 800616B8   040CC000 813DBAD0   00010007 013D3800

*CLOCK COMP   FFFFFFFF FFFFF000
*CPU TIMER    FFFFFFFF FFFDC700
*TOD CLOCK    A62D139E 20711812

*TRANS EXC ID 00000000
*MONITOR CLS  0001
*PER CODE     0000
*PER ADDRESS  00000000
*MONITOR CODE 00000000

*PREFIX REG   013DE000
*SYSTEM LOG   01131878
*CIO IRB      10804007 0013DD90 0C000000 00800000
*CIO SENSE    00000000 00000000 00000000 00000000
              00000000 00000000 00000000 00000000
*CIO SDA      04E0
*CIO SENSE COMPLETION  0C000018

*LAST 10 BRANCH TABLE ENTRIES
              80067D8E
              80060E44
              80060F00
              80060CE8
              80053D8A
              800C0250
              800BFB70
              800C0250
              80004C00
              80060E44 LAST
```

Figure 38. I-Stream Status Display

Table 26 explains the headings and labels that appear in this section of the main storage dump.

Table 26. Headings in the I-Stream Status Display

Heading	Description
*IS STATUS DISPLAY	This heading indicates which I-stream information is about to be displayed. The heading includes the I-stream number and control program (CPU) address. An I-stream status display is included for each I-stream that was active at the time of the system error dump. The status display in the previous example is for the main I-stream.
*GENERAL REGISTERS	This is a general heading for the list of the contents of all general registers. The contents reflect the status of the registers when the system error dump was issued.
* R0–R7	These general registers may be used by operational programs and the control program. Their contents are variable.

Table 26. Headings in the I-Stream Status Display (continued)

Heading	Description
* R8	When a system error dump is issued for an error that is not related to an operational program, the content of R8 is unpredictable. When a dump reflects an operational program error, R8 should contain the base address of the operational program processing the entry that caused the dump. **Note:** The system error message sent to the prime computer room agent set (CRAS) notifying the operator of the dump contains the program name referenced through 4(R8).
* R9	When a system error dump is issued for an error that is not related to an operational program, the contents of R9 are unpredictable. When a dump reflects an operational program error, R9 should contain the base address of the entry control block (ECB) in use by the entry that caused the system error dump.
* R10–R13	R10 is a floating base register for use by the control program (CP). R11 and R12 are preset by the CP initializer to the base addresses of the second and third 4096-byte modules of the CCNUCL CSECT CP. The first module, which is located in the first 4096 bytes of main storage, does not require a base register allocation. R13 is used throughout the CP as a stack pointer. The following values should be in these registers. **Register** **Value** R10 Variable R11 X'1000' R12 X'2000' R13 Variable
* R14–R15	Same as R0.
*CONTROL REGISTERS	This is a general heading for the list of the contents of all control registers. The contents reflect the status of the registers when the system error dump was issued. Control register 1 contains the segment table designation for the currently active ECB virtual memory (EVM). Control register 13 contains the segment table designation for the system virtual memory (SVM) on this I-stream. Bits 1–19 of a segment table designation, with 12 zeros appended on the right, form the real address of the segment table for an address space. See *ESA/370 Principles of Operation* and *ESA/390 Principles of Operation* for more information. Control registers 9–11 contain the event masks, starting address and ending address associated with program event recording (PER) support. If branch trace is active, control register 12 identifies the real address of the next available branch trace entry.
*FLT PT REGISTERS	This is a general heading for the list of the contents of all floating point registers. The contents reflect the status of the registers when the system error dump was issued.
*CURRENT PSW AT TIME OF DUMP	This is the program status word at the time the error occurred.
*PSWS	See *ESA/370 Principles of Operation* and *ESA/390 Principles of Operation* for information about the program status words (PSWs) and interrupt codes.

Table 26. Headings in the I-Stream Status Display (continued)

Heading	Description
* RESTART	Old and new restart PSWs. If system error processing is unable to cause all I-streams to enter CPSE, the error is forced catastrophic with the SIGP EXTERNAL CALL FAILED message, and a SIGP restart order is used. In this situation the address field of the restart old PSW in the I-stream status displays for the other I-streams indicates where the I-streams were processing when the SIGP failure occurred.
* EXTERNAL	Old and new external PSWs, and external interruption code. The address field of the external old PSW in the I-stream status displays for the other I-streams indicates where the I-streams were processing when the failing I-stream caused them to interrupt what they were doing and enter CPSE.
* SVC	Old and new supervisor call PSWs, and SVC interruption code. The SVC old PSW references the last SVC issued. The address portion locates the place from where it was issued (if the system error dump was issued during a macro service routine) or points to the next sequential instruction after the SVC and its parameters, if control has been returned to the program issuing the SVC. The SVC new PSW contains the address of the macro decoder CSECT. See *TPF Main Supervisor Reference* for more information about the macro decoder.
* PROGRAM	Old and new program PSWs, program interruption code. For program interrupts for which the failing operations nullified (for example, segment- and page- translation exceptions), the address portion of the program old PSW points to the failing instruction. For all other program interrupts, the address portion of the program old PSW points to location of the next sequential instruction after the instruction that caused the interrupt.
* MACHINE	Old and new machine check PSWs, and machine interruption code.
* I/O	Old and new input/output (I/O) PSWs, and I/O interruption code.
*CLOCK COMP	Identifies the contents of the clock comparator.
*CPU TIMER	Identifies the contents of the central processor unit (CPU) timer.
*TOD CLOCK	Identifies the contents of the time-of-day (TOD) clock.
*TRANS EXC ID	On a segment- or page-translation exception, bits 1–19 identify the virtual address that caused the exception, rounded down to the nearest 4 KB boundary. Bits 30–31 are copied from the address space control bits in the PSW at the time the error occurred. If these bits are zeros the error occurred in an ECB virtual memory (EVM); if they are ones the error occurred in the system virtual memory (SVM). In the previous example, a virtual address of X'0030F000' caused a page translation exception in an EVM.
*MONITOR CLS	Identifies the monitor class of the most recent monitor call issued.
*PER CODE	Identifies the type of PER event associated with the most recent PER interrupt.
*PER ADDRESS	Identifies the address of the instruction that triggered the most recent PER interrupt.
*MONITOR CODE	Identifies the monitor code of the most recent monitor call issued.
*PREFIX REG	Identifies the real address of the prefix page for this I-stream.

Table 26. Headings in the I-Stream Status Display (continued)

Heading	Description
*SYSTEM LOG	Identifies the system virtual address of the start of the system log trace buffer for this I-stream.
*CIO IRB	Identifies the first 4 words of the interrupt response block. This includes the subchannel-status word and the extended-status word for the most recent I/O interrupt.
*CIO SENSE	Identifies the sense data, if any, last obtained by the common I/O routine.
*CIO SDA	Identifies the symbolic device address (SDA) that was last used by the common I/O routine. The IRB and sense data presented previously are for this device.
*CIO SENSE COMPLETION	Identifies the sense completion code last obtained by CIO.
*LAST 10 BRANCH TABLE ENTRIES	The destination addresses of the last 10 BALR, BASR, BASSM, or BAKR instructions issued. This display appears only if branch trace is active.

Area of Program Error

Figure 39 shows an example of an area of program error.

```
*****************************************************************************************************************
*AREA OF PROG ERROR
001A4270      93EC0B06 D5005000          8204D200 93B81000          DC0093B8 82D20000          00000010 00000000
001A4290      00000000 00000000          00000000 40404040          40404040 40404040          40404040 40404040
001A42B0      40404040 40404040          40400000 00000006          2CC2C5C7 C9D540C4          C9E2D7D3 C1E8404B
001A42D0      4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B40404B
001A42F0      4B4B4B4B 4B4B4B60          404B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A4310      4B4B4B4B 4B4B4B4B          4B4B4B4B 404B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B2BC5
001A4330      D5C440D6 C640C4C9          E2D7D3C1 E8406040          E9C5D9D6 C5C440D3          C9D5C5E2 40D5D6E3
001A4350      40C4C9E2 D7D3C1E8          C5C44B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A4370      4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A4390      4B4B4B4B 4B4B4B4B          4B4B404B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A43B0      4B4B4B4B 4B5B5C4B          4B4B6061 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A43D0      4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A43F0      4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B          4B4B4B4B 4B4B4B4B
001A4410      4B4B4B4B 4B4B4B4B          4B4B4BC1 C2C3C4C5          C6C7C8C9 4B4B4B4B          4B4B4BD1 D2D3D4D5
001A4430      D6D7D8D9 4B4B4B4B          4B4B4B4B E2E3E4E5          E6E7E8E9 4B4B4B4B          4B4BF0F1 F2F3F4F5
001A4450      F6F7F8F9 4B4B4B4B          4B4B0000 00000000          00000A00 00000000          F0F761F2 F161F9F2
001A4470      F0F14BF4 F1F2F400          00FF0597 C3E5C9E3          94009368 9501910D          4770812A 95019105
001A4490      47808412 41209008          58309100 D2009328          200ED200 932B2010          D2009348 200FD201
001A44B0      92E22013 D2009323          2015BFE3 20110AFC          200ED203 30042008          4D70837A 4DE0B128
001A44D0      01DC18EE 9180E000          4710807C 9108E001          4710807C 91C03000          471080CC 58003008
001A44F0      47F081AC 96209328          91803000 4710808C          96029328 91403000          4780809C 91409368
001A4510      471080B0 48F01010          41EF1012 926DE000          41F0F001 40F01010          4DE0B230 00017218
001A4530      91409368 47108330          96029328 4D70837A          47F0808C 91409368          4710807C 05EB0108
001A4550      18214D70 837A0A34          4008D6D4 D2032008          90849501 91154780          80F60A16 100005EB
   .             .        .                .        .                 .        .                 .        .
   .             .        .                .        .                 .        .                 .        .
   .             .        .                .        .                 .        .                 .        .
001A5250      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000

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

Figure 39. Dump Cover Page

Table 27 on page 75 explains the headings and labels that appear in this section of the main storage dump.

Table 27. Headings in the Dump Cover Page

Heading	Description
*AREA OF PROG ERROR	Only applicable for hard errors (CTL-I000001 through CTL-I000004). Displays an area starting 2 K below the address in the program old PSW and ending 2 K above this address. This display occurs only for the failing I-stream. If an input/output (I/O) interrupt occurred within the range shown, the I/O INTERRUPT OCCURRED AT *xxxxxxxx* heading appears to call attention to this fact.

Areas Referenced by General Registers

Figure 40 shows an example of areas referenced by general registers.

```
*AREA REFERENCED BY REGISTER 2
00319400      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319780      00000000 0021A5E8          00000000 FFFFFFFF          00840000 0001C811          4040C3E2 D4D7F0F0
003197A0      F9F7C940 F1F94BF5          F34BF0F3 40C3D7E4          60C240E2 E260C2E2          E24040E2 E2E460C8
003197C0      D7D54040 C9E260F0          F315C7D6 C7D6F0F0          F0F1C940 F1F94BF5          F34BF0F3 40C1D7D7
003197E0      D3C9C3C1 E3C9D6D5          40C6C9E7 C5C440C3          D6D9C540 D3D6C1C4          C5C415D4 D6C4C540
00319800      F715E2E2 E440C8D7          D5406060 40C960E2          40F0F315 19FF0000          00000000 00000000
00319820      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
003198E0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130000
00319900      00000000 00211500          00000000 00000000          00670100 00F1C211          0000C3E2 D4D7F0F0
00319920      F9F7C940 F1F94BF5          F34BF0F1 40C3D7E4          60C240E2 E260C2E2          E24040E2 E2E460C8
00319940      D7D54040 C9E260F0          F1110000 C3E2D4D7          F0F0F9F9 C940F1F9          4BF5F34B F0F140F0
00319960      F1F0F0F0 F060C240          E9E3D4D5 E340D9E3          C140F4F2 F040C2D7          4E000000 00000000
00319980      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319A60      00000000 00000000          00000000 00000000          00000000 000031E7          B000327B E0130000
00319A80      0E4B4B4B 4B4B4B60          C240E9E3 C5E2E300          00000000 00000000          00000000 00000000
00319AA0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319BE0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130000
00319C00      00000000 0051185A          00000000 00000000          00090100 00E9E3C5          E2E34E00 00000000
00319C20      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319D60      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130000
00319D80      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319E80      00000000 0051185A          00000000 00000000          00090100 00E9E3C5          E2E34E00 00000000
00319EA0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319FE0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130057

*AREA REFERENCED BY REGISTER 3
00319280      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319780      00000000 0021A5E8          00000000 FFFFFFFF          00840000 0001C811          4040C3E2 D4D7F0F0
003197A0      F9F7C940 F1F94BF5          F34BF0F3 40C3D7E4          60C240E2 E260C2E2          E24040E2 E2E460C8
003197C0      D7D54040 C9E260F0          F315C7D6 C7D6F0F0          F0F1C940 F1F94BF5          F34BF0F3 40C1D7D7
003197E0      D3C9C3C1 E3C9D6D5          40C6C9E7 C5C440C3          D6D9C540 D3D6C1C4          C5C415D4 D6C4C540
00319800      F715E2E2 E440C8D7          D5406060 40C960E2          40F0F315 19FF0000          00000000 00000000
00319820      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
003198E0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130000
00319900      00000000 00211500          00000000 00000000          00670100 00F1C211          0000C3E2 D4D7F0F0
00319920      F9F7C940 F1F94BF5          F34BF0F1 40C3D7E4          60C240E2 E260C2E2          E24040E2 E2E460C8
00319940      D7D54040 C9E260F0          F1110000 C3E2D4D7          F0F0F9F9 C940F1F9          4BF5F34B F0F140F0
00319960      F1F0F0F0 F060C240          E9E3D4D5 E340D9E3          C140F4F2 F040C2D7          4E000000 00000000
00319980      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319A60      00000000 00000000          00000000 00000000          00000000 000031E7          B000327B E0130000
00319A80      0E4B4B4B 4B4B4B60          C240E9E3 C5E2E300          00000000 00000000          00000000 00000000
00319AA0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00319BE0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00130000
   .             .        .                .        .                .        .                 .        .
   .             .        .                .        .                .        .                 .        .
   .             .        .                .        .                .        .                 .        .
```

Figure 40. Areas Referenced by General Registers

Table 28 on page 76 explains the headings and labels that appear in this section of the main storage dump.

Table 28. Headings in the Areas Referenced by General Registers

Heading	Description
*AREA REFERENCED BY REGISTER *n*	The contents of each general register on the failing I-stream are examined in turn. If a register contains a value that can be interpreted as a valid address in the failing address space, an area beginning 2 K below this address and ending 4 KB above this address is displayed. Areas referenced by the registers are not displayed if they are also displayed later in the system error dump.

C/C++ Language Register Environments

There are 2 register environments, which follow here.

- One for TARGET(TPF), which is the same as the TPF system environment. For TARGET(TPF), the registers are affected by how TARGET(TPF) C operates. The compiler makes assumptions about registers:
  - After the ICPLOG macro call, the support prolog
  - Before a macro or function call
  - After a macro or function call
  - Preceding an ICELOG macro call, the support epilogue.
- One for ISO-C. For ISO-C, most registers are left to the compiler and the user. The compiler makes assumptions about registers when calling and returning from functions.

The compiler makes assumptions about registers for C/C++ language. These register assumptions are displayed in Table 29.

Table 29. Register Conventions for C/C++ Language

Register	Environment	Description
R0	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Not specified by the compiler.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler.
	TARGET(TPF): Before macro call	Not specified by the compiler, except as required by the macro entry interface.
	TARGET(TPF): After macro return	Same as at call unless modified by the macro service routine
	TARGET(TPF): Preceding ICPLOG	Not specified by the compiler.
R1	ISO-C function entry	C linkage parameter list.
	ISO-C function return	Address of a function returned from.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler.
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface..
	TARGET(TPF): After macro return	Same as at call unless modified by the macro service routine. (The function or macro base register set up before the call is required until R8 is restored from CSTKLBAS.)
	TARGET(TPF): Preceding ICPLOG	Not specified by the compiler.

Table 29. Register Conventions for C/C++ Language (continued)

Register	Environment	Description
R2	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler. If the register is used by the function, it must have been saved by coding the ICPLOG HIGHREG parameter
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG	Not specified by the compiler.
R3	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler. If the register is used by the function, it must have been saved by coding the ICPLOG HIGHREG parameter
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG	Not specified by the compiler.

Table 29. Register Conventions for C/C++ Language (continued)

Register	Environment	Description
R4	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler. If the register is used by the function, it must have been saved by coding the ICPLOG HIGHREG parameter
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler.
R5	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler. If the register is used by the function, it must have been saved by coding the ICPLOG HIGHREG parameter
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler.

Table 29. Register Conventions for C/C++ Language (continued)

Register	Environment	Description
R6	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	C parameter list.
	TARGET(TPF): Before macro call	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Base register or not specified by the compiler. At least one of these must be a base register for the function or the macro. Other than that, not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG call	Return value (if by register).
R7	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Current stack pointer. If the function does not allocate a stack frame (ICPLOG FRAMESIZE=NO), R7 points to the caller's stack frame. If the function does allocate a stack frame, R7 points to the called function's stack frame. The stack frame tags that are of use to library functions are: CSTKMIN An absolute symbol for the beginning of the function's automatic storage. The function specifies its automatic storage by coding the ICPLOG FRAMESIZE parameter as the number of bytes of automatic storage to be allocated. CSTKLBAS A relative symbol for the function's base address. The function stores R8 in this field if it will need to reload its base address after a macro call
	TARGET(TPF): Before macro call	Not specified by the compiler. Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Not specified by the compiler. Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG call	Stack Pointer.

Table 29. Register Conventions for C/C++ Language (continued)

Register	Environment	Description
R8	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Program base. Conventionally, this address is also stored in CSTKLBAS.
	TARGET(TPF): Before macro call	Program base.
	TARGET(TPF): After macro return	Same as at Call.
	TARGET(TPF): Preceding ICPLOG call	Stack Pointer.
R9	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	ECB page 1.
	TARGET(TPF): Before macro call	ECB page 1.
	TARGET(TPF): After macro return	ECB page 1.
	TARGET(TPF): Preceding ICPLOG call	ECB page 1.
R10	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler.
	TARGET(TPF): Before macro call	Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler.
R11	ISO-C function entry	Not specified by the compiler.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	X'1000'
	TARGET(TPF): Before macro call	X'1000'
	TARGET(TPF): After macro return	X'1000'
	TARGET(TPF): Preceding ICPLOG call	X'1000'
R12	ISO-C function entry	Address of TRC.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	X'2000'
	TARGET(TPF): Before macro call	X'2000'
	TARGET(TPF): After macro return	X'2000'
	TARGET(TPF): Preceding ICPLOG call	X'2000'
R13	ISO-C function entry	Stack Pointer.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Not specified by the compiler.
	TARGET(TPF): Before macro call	Not specified by the compiler.
	TARGET(TPF): After macro return	Not specified by the compiler.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler.

Table 29. Register Conventions for C/C++ Language (continued)

Register	Environment	Description
R14	ISO-C function entry	NSI of Caller.
	ISO-C function return	Same as at Call.
	TARGET(TPF): After ICPLOG call	Return address (saved by ICPLOG) if a library call, otherwise unknown.
	TARGET(TPF): Before macro call	Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Same as Call except as modified.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler.
R15	ISO-C function entry	Function entry point.
	ISO-C function return	Return value.
	TARGET(TPF): After ICPLOG call	Function entry point if a library call, otherwise unknown.
	TARGET(TPF): Before macro call	Not specified by the compiler unless required by the macro entry interface.
	TARGET(TPF): After macro return	Same as at Call except as modified.
	TARGET(TPF): Preceding ICPLOG call	Not specified by the compiler

Collated Macro Trace Table

Figure 41 on page 82 shows an example of the collated macro trace table.

```
*COLLATED ECB MACRO TRACE ENTRIES
*ECB ADDRESS      *MACRO *PROG *LOCATION   *PARAMETERS                          *SVC OLD PSW    *CODE *INX   *TIME STAMP
--------           CXFRC  CHSZ* 00049371 ECB-IMMED ADR-02400000 FMT-A5          CTL-XFER 00049382 CF00      B43B4E81 B784D303
--------           *DUMP* ----  ----     PROGRAM CHECK INT CODE-01             071D0000 866C108E CB00      B43B4E81 B7816583
023FA000 *         DECBC  QDCB  +050     TYPE-CREATE  D-00922720               071D1000 866C85BA 003B 0035 B43B4E81 B77E5BC3
023FD000           MONWC  WGR1  +???                                           071D2000 8000227A 007E      B43B4E81 B77C1D03
023FD000           FINHC  WGR1  +186     FA-F4017001 RECID-C3D6 LVL-D3         071D3000 804F1D28 0020      B43B4E81 B76AD083
023FD000           ENTRC  CVIQ  +570     ENTER TO WGR1                                 84A306E4 FF12      B43B4E81 B76A8F03
023FD000           BACKC  CVI1  +1CA     RETURN TO CVIQ                                84A310DE FF02      B43B4E81 B769F343
023FD000           GETCC  CVI1  +1A0     AD-0091FE80 TYPE-L1   LVL-D7          071D1000 84A310B2 002C      B43B4E81 B769DF43
023FD000           RELCC  CVI1  +164     AD-0091FE80 BKTYP-0021 LVL-D9         071D1000 84A31076 0032      B43B4E81 B769CE83
023FD000           CEBIC  CVI1  +0E4                                           071D0000 84A30FF6 00FC      B43B4E81 B769B243
023FD000           CEBIC  CVI1  +0C4                                           071D0000 84A30FD6 00FC      B43B4E81 B7695B03
023FD000           GETCC  CVI1  +018     AD-0091FA80 TYPE-L1   LVL-D9          071D3000 84A30F2A 002C      B43B4E81 B7692003
023FD000           ENTRC  CVIQ  +0F2     ENTER TO CVI1                                 84A30266 FF12      B43B4E81 B7691443
023FD000           BACKC  UOP1  +008     RETURN TO CVIQ                                84A30EFC FF02      B43B4E81 B7690343
023FD000           ENTRC  CVIQ  +0BA     ENTER TO UOP1                                 84A3022E FF12      B43B4E81 B768E583
023FD000           GETCC  CVIQ  +072     AD-0091FC00 TYPE-L0   LVL-D4          071D0000 84A301E4 002C      B43B4E81 B768B303
023FD000           ENTNC  -CP-  023FD2D4 ENTER TO CVIQ                                 823FD2D8 FF10      B43B4E81 B7688083
023FA000 *         DLAYC  QDCB  +???                                           071D1000 8465C704 000C      B43B4E81 B7669743
--------           CXFRC  CHSZ* 0000BBE5 ECB-DEFER ADR-026FD000 FMT-A5         CTL-XFER 0000BBF6 CF00      B43B4E81 B7665F03
023FA000 *         WTOPC  QDCB  +???                                           071D0000 816059AA 00F0      B43B4E81 B7658983
023FA000 *         KEYRC  QDCB  +???                                           070D1000 84661056 0076      B43B4E81 B7650F43
023FA000 *         CINFC  QDCB  +???                                           071D2000 81604FF2 0060      B43B4E81 B764E683
023FA000 *         MALOC  QDCB  +???     SIZE-00001000 BLOCK-00A0CFD8          071D1000 845ED068 003B 0005 B43B4E81 B762EF03
023FA000 *         KEYRC  QDCB  +???                                           070D1000 84661056 0076      B43B4E81 B761EDC3
023FA000 *         CINFC  QDCB  +???                                           071D2000 81604FF2 0060      B43B4E81 B761C903
023FA000 *         KEYRC  QDCB  +???                                           070D1000 84661056 0076      B43B4E81 B7616A83
023FA000 *         CINFC  QDCB  +???                                           071D2000 81604FF2 0060      B43B4E81 B7613F83
023FA000 *         KEYRC  QDCB  +???                                           070D1000 84661056 0076      B43B4E81 B760F943
023FA000 *         CINFC  QDCB  +???                                           071D2000 81604FF2 0060      B43B4E81 B760E643
023FA000 *         KEYRC  QDCB  +???                                           070D1000 84661056 0076      B43B4E81 B760B043
023FA000 *         CINFC  QDCB  +???                                           071D2000 81604FF2 0060      B43B4E81 B7606443
023FA000 *         GSWBC  CGSW  +026                                           071D1000 84677108 003B 002B B43B4E81 B7600A43
023FA000 *         MALOC  QDCB  +???     SIZE-00000108 BLOCK-00A02370          071D1000 845ED068 003B 0005 B43B4E81 B75FA003
023FA000 *         FREEC  QDCB  +???     BLOCK-00A02460                        071D1000 845DAA70 003B 0007 B43B4E81 B75F5583
023FA000 *         MALOC  QDCB  +???     SIZE-00000007 BLOCK-00A02460          071D1000 845ED068 003B 0005 B43B4E81 B75E2303
023FA000 *         RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE         071D0000 81604FF2 0032      B43B4E81 B75D8103
023FA000 *         FINWC  QDCB  +???     FA-FC14000D RECID-FC2A LVL-DE         071D0000 81604FF2 0026      B43B4E81 B6FEF041
023FA000 *         FREEC  CJ00  +096     BLOCK-00A02208                        071D1000 846E3FB0 003B 0007 B43B4E81 B6FDAE41
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-009226A0               071D0000 847D097A 003B 0035 B43B4E81 B6FD8581
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-00922620               071D1000 847D097A 003B 0035 B43B4E81 B6FD5C81
023FA000 *         FREEC  CJ00  +026     BLOCK-00A07598                        071D1000 8470729C 003B 0007 B43B4E81 B6FCD581
023FA000 *         FREEC  CJ00  +026     BLOCK-00A0CF70                        071D1000 846E6D38 003B 0007 B43B4E81 B6FC7F01
023FA000 *         MALOC  CJ00  +022     SIZE-00001068 BLOCK-00A0CF70          071D1000 84707CB4 003B 0005 B43B4E81 B6FAF101
023FA000 *         RELCC  CJ00  +046     AD-0092C000 BKTYP-0051 LVL-DF         071D1000 847FA0F4 0032      B43B4E81 B6FA5301
023FA000 *         MALOC  CJ00  +022     SIZE-00000A00 BLOCK-00A07598          071D1000 847F9FE4 003B 0005 B43B4E81 B6F98781
023FA000 *         FINWC  CJ00  +0E2     FA-0000000000145831 ID-FC16 D-00922620 071D1000 847F3198 0026      B43B4E81 B646AC87
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-009226A0               071D1000 847D07D2 003B 0035 B43B4E81 B6465687
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-00922620               071D1000 847D07D2 003B 0035 B43B4E81 B6462B07
023FA000 *         CALOC  CJ00  +028     SIZE-00000270 BLOCK-00A02208          071D1000 847EA762 003B 0006 B43B4E81 B645AB47
023FA000 *         RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE         071D0000 81604FF2 0032      B43B4E81 B643FC87
023FA000 *         FINWC  QDCB  +???     FA-FC140005 RECID-FC2A LVL-DE         071D0000 81604FF2 0026      B43B4E81 B58A7C87
023FA000 *         FREEC  CJ00  +096     BLOCK-00A02208                        071D1000 846E3FB0 003B 0007 B43B4E81 B5892A47
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-009225A0               071D0000 847D097A 003B 0035 B43B4E81 B588F807
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-00922520               071D1000 847D097A 003B 0035 B43B4E81 B588CF07
023FA000 *         FREEC  CJ00  +026     BLOCK-00A07468                        071D1000 8470729C 003B 0007 B43B4E81 B5884187
023FA000 *         FREEC  CJ00  +026     BLOCK-00A0CF70                        071D1000 846E6D38 003B 0007 B43B4E81 B587E787
023FA000 *         MALOC  CJ00  +022     SIZE-00001068 BLOCK-00A0CF70          071D1000 84707CB4 003B 0005 B43B4E81 B5864A47
023FA000 *         RELCC  CJ00  +046     AD-0092C000 BKTYP-0051 LVL-DF         071D1000 847FA0F4 0032      B43B4E81 B6FA5301
023FA000 *         MALOC  CJ00  +022     SIZE-00000A00 BLOCK-00A07598          071D1000 847F9FE4 003B 0005 B43B4E81 B6F98781
023FA000 *         FINWC  CJ00  +0E2     FA-0000000000145831 ID-FC16 D-00922620 071D1000 847F3198 0026      B43B4E81 B646AC87
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-009226A0               071D1000 847D07D2 003B 0035 B43B4E81 B6465687
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-00922620               071D1000 847D07D2 003B 0035 B43B4E81 B6462B07
023FA000 *         CALOC  CJ00  +028     SIZE-00000270 BLOCK-00A02208          071D1000 847EA762 003B 0006 B43B4E81 B645AB47
```

Figure 41. Collated Macro Trace Table (Part 1 of 5)

```
023FA000 *         RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE          071D0000 81604FF2 0032      B43B4E81 B643FC87
023FA000 *         FINWC  QDCB  +???     FA-FC140005 RECID-FC2A LVL-DE          071D0000 81604FF2 0026      B43B4E81 B58A7C87
023FA000 *         FREEC  CJ00  +096     BLOCK-00A02208                         071D1000 846E3FB0 003B 0007 B43B4E81 B5892A47
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-009225A0                071D0000 847D097A 003B 0035 B43B4E81 B588F807
023FA000 *         DECBC  CJ00  +0C0     TYPE-RELEASE D-00922520                071D1000 847D097A 003B 0035 B43B4E81 B588CF07
023FA000 *         FREEC  CJ00  +026     BLOCK-00A07468                         071D1000 8470729C 003B 0007 B43B4E81 B5884187
023FA000 *         FREEC  CJ00  +026     BLOCK-00A0CF70                         071D1000 846E6D38 003B 0007 B43B4E81 B587E787
023FA000 *         MALOC  CJ00  +022     SIZE-00001068 BLOCK-00A0CF70           071D1000 84707CB4 003B 0005 B43B4E81 B5864A47
023FA000 *         RELCC  CJ00  +046     AD-0092C000 BKTYP-0051 LVL-DF          071D1000 847FA0F4 0032      B43B4E81 B585A247
023FA000 *         MALOC  CJ00  +022     SIZE-00000B30 BLOCK-00A07468           071D1000 847F9FE4 003B 0005 B43B4E81 B584C787
023FA000 *         FINWC  CJ00  +0E2     FA-0000000000014582D ID-FC16 D-00922520 071D1000 847F3198 0026      B43B4E81 B4B59A84
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-009225A0                071D1000 847D07D2 003B 0035 B43B4E81 B4B54444
023FA000 *         DECBC  CJ00  +098     TYPE-CREATE  D-00922520                071D1000 847D07D2 003B 0035 B43B4E81 B4B51784
023FA000 *         CALOC  CJ00  +028     SIZE-00000270 BLOCK-00A02208           071D1000 847EA762 003B 0006 B43B4E81 B4B48984
023FA000 *         RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE          071D0000 81604FF2 0032      B43B4E81 B4B2CDC4
023FA000 *         FINWC  QDCB  +???     FA-FC140001 RECID-FC2A LVL-DE          071D0000 81604FF2 0026      B43B4E81 B20A8747
023FA000 *         MALOC  QDCB  +???     SIZE-00000015 BLOCK-00A02488           071D1000 845ED068 003B 0005 B43B4E81 B20972C7
023FA000 *         MALOC  QDCB  +???     SIZE-00000030 BLOCK-00A024B0           071D1000 845ED068 003B 0005 B43B4E81 B2091E87
023FA000 *         MALOC  QDCB  +???     SIZE-0000012C BLOCK-00A024F0           071D1000 845ED068 003B 0005 B43B4E81 B208B987
023FA000 *         CENVC  QDCB  +???                                            071D1000 8462E46C 003B 0021 B43B4E81 B20854C7
023FA000 *         KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B2078F07
023FA000 *         CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B2076A47
023FA000 *         KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B2070C87
-------------------------------------------------------------------------------------------------------------------------------------
--------           $FINDC ----          PIA-00137C54 DECB-00F53138 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 A6E74C42
--------           D-CROS ----  ----     IS-14 P1-026FA800 P2-00000000          PIA-ADDR 00139790 CE00      B43B4E81 A3D63F87
--------           D-CROS ----  ----     IS-14 P1-026FA400 P2-00000000          PIA-ADDR 00139790 CE00      B43B4E81 A3CEF807
--------           D-CROS ----  ----     IS-14 P1-026FA000 P2-00000000          PIA-ADDR 00139790 CE00      B43B4E81 A3CB1987
--------           $FINDC ----          PIA-00137A28 DECB-00F53138 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 9CF71D40
--------           $FINDC ----          PIA-00137A28 DECB-00F530B0 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 9B4D0840
--------           $FINDC ----          PIA-00137A28 DECB-00F53468 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A58486
--------           $FINDC ----          PIA-00137A28 DECB-00F533E0 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A4F846
--------           $FINDC ----          PIA-00137A28 DECB-00F53358 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A45E46
--------           $FINDC ----          PIA-00137A28 DECB-00F532D0 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A3D006
--------           $FINDC ----          PIA-00137A28 DECB-00F53248 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A32C46
--------           $FINDC ----          PIA-00137A28 DECB-00F531C0 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99A24B46
--------           $FINDC ----          PIA-00137A28 DECB-00F53138 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99972606
--------           $FINDC ----          PIA-00137A28 DECB-00F530B0 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 99898946
--------           $FINDC ----          PIA-001378D6 DECB-00F53028 ID-00FF CD-00 FA-00000000     CA00      B43B4E81 97E7E487
-------- *         CXFRC  CHSZ* 00019247 ECB-IMMED ADR-023FA000 FMT-B5          CTL-XFER 0001925A CF00      B43B4E81 95894208
023F7000           EXITC  CL23  +13A                                            070D0000 8050364C 0014      B43B4E81 925E7208
023F7000           CINFC  CL23  +066                                            071D2000 80503578 0060      B43B4E81 925E4C08
023F7000           CRETCS CL23  +058     TO ACTIVATE CL23                       071D2000 8050356A 0064      B43B4E81 925E2408
023F7000           CEBIC  CL23  +00C                                            071D1000 8050351E 00FC      B43B4E81 925E0908
023F7000           ENTNC  -CP-  023F72D4 ENTER TO CL23                                   823F72D8 FF10      B43B4E81 925DB988
--------           CXFRC  CHSZ* 000029F1 ECB-DEFER ADR-026F9C00 FMT-A5          CTL-XFER 00002A00 CF00      B43B4E81 925A1008
023F4000           KEYRC  CUIU  +???                                            070D1000 84661056 0076      B43B4E81 6C065583
023F4000           CINFC  CUIU  +???                                            071D2000 81604FF2 0060      B43B4E81 6C064083
023F4000           FREEC  CUIU  +???     BLOCK-00A01768                         071D1000 845DAA70 003B 0007 B43B4E81 6C05B983
023F4000           RELPC  CUIU  +???                                            071D1000 8160550A 006E      B43B4E81 6C058703
023F4000           EXITC  CUIU  +???                                            070D1000 845DA34C 0014      B43B4E81 6C04E383
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C04B903
023F4000           KEYRC  CUIU  +???                                            070D1000 84661056 0076      B43B4E81 6C049883
023F4000           BACKC  CL23  +13C     RETURN TO CUIU                                 80503650 FF02      B43B4E81 6C048083
023F4000           CINFC  CL23  +066                                            070D0000 80503578 0060      B43B4E81 6C046883
023F4000           CEBIC  CL23  +00C                                            070D1000 8050351E 00FC      B43B4E81 6C045D83
023F4000           EFCALL CUIU  +0164CEC ENTER TO CL23                                   8666ED0E FFFC      B43B4E81 6C0436C3
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C03C803
023F4000           CEBIC  CUIU  +???                                            070D1000 8465ACCC 00FC      B43B4E81 6C03A403
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C037A83
023F4000           CEBIC  CUIU  +???                                            070D1000 8465AD70 00FC      B43B4E81 6C035043
023F4000           KEYRC  CUIU  +???                                            070D1000 84661056 0076      B43B4E81 6C02D6C3
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C027CC3
023F4000           CINFC  CUIU  +???                                            071D2000 81604FF2 0060      B43B4E81 6C025483
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C023703
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C020543
023F4000           CINFC  CUIU  +???                                            070D2000 81604FF2 0060      B43B4E81 6C004F83
023F4000           MALOC  CUIU  +???     SIZE-00000018 BLOCK-00A01740           070D1000 845ED068 003B 0005 B43B4E81 6BF92AC3
023F4000           MALOC  CUIU  +???     SIZE-00000048 BLOCK-00A01768           070D1000 845ED068 003B 0005 B43B4E81 6BF1E703
```

Figure 41. Collated Macro Trace Table (Part 2 of 5)

```
023F4000          MALOC  -CP-  0012D99C SIZE-00009548 BLOCK-00A02A90
023F4000          GETPC  CUIU  +???
023F4000          CINFC  CUIU  +???
023F4000          GETPC  CUIU  +???
023F4000          MALOC  -CP-  0012D99C SIZE-00000818 BLOCK-00A017C0
023F4000          ENTNC  -CP-  023F42D4 ENTER TO CUIU
0216F000          KEYRC  CUIU  +???
0216F000          CINFC  CUIU  +???
0216F000          EXITC  CUIU  +???
0216F000          CINFC  CUIU  +???
0216F000          GETPC  CUIU  +???
0216F000          MALOC  -CP-  0012D99C SIZE-00000818 BLOCK-00A017C0
0216F000          ENTNC  -CP-  0216F2D4 ENTER TO CUIU
--------          CXFRC  CHSZ* 00010235 ECB-DEFER ADR-026F8C00 FMT-A5
--------          CXFRC  CHSZ* 000101CF ECB-DEFER ADR-026F8800 FMT-A5
--------          D-CROS ----  ----     IS-14 P1-026F7800 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F7400 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F7000 P2-00000000
023F1000          EXITC  CL23  +13A
023F1000          CINFC  CL23  +066
023F1000          CRETCS CL23  +058     TO ACTIVATE CL23
023F1000          CEBIC  CL23  +00C
023F1000          ENTNC  -CP-  023F12D4 ENTER TO CL23
--------          CXFRC  CHSZ* 000029F1 ECB-DEFER ADR-026F6800 FMT-A5
--------          D-CROS ----  ----     IS-14 P1-026F5800 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F5400 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F5000 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F3400 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F3000 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F2C00 P2-00000000
0217B000          EXITC  CL23  +13A
0217B000          CINFC  CL23  +066
0217B000          CRETCS CL23  +058     TO ACTIVATE CL23
0217B000          CEBIC  CL23  +00C
0217B000          ENTNC  -CP-  0217B2D4 ENTER TO CL23
--------          CXFRC  CHSZ* 000029F1 ECB-DEFER ADR-026F2400 FMT-A5
--------          D-CROS ----  ----     IS-14 P1-026F1400 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F1000 P2-00000000
--------          D-CROS ----  ----     IS-14 P1-026F0C00 P2-00000000
0257A000          KEYRC  CEL5  +???
0257A000          CINFC  CEL5  +???
0257A000          EXITC  CEL5  +???
0257A000          BACKC  CLEK  +03A     RETURN TO CEL5
0257A000          POSTC  CLEK  +02E
0257A000          EFCALL CEL5  +00006CA ENTER TO CLEK
0257A000          EFRETN CELW  +0000214 RETURN TO CEL5
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW

                  8012D9A0 FF22      B43B4E81 6BD07400
070D0000 8160580A 006C      B43B4E81 6BD00780
071D2000 81604FF2 0060      B43B4E81 6BCF76C0
050C0000 8012D90E 006C      B43B4E81 6BCEFC80
                  8012D9A0 FF22      B43B4E81 6BCC6B00
                  823F42D8 FF10      B43B4E81 6BC9F380
070D1000 84661056 0076      B43B4E81 6BC7D640
071D2000 81604FF2 0060      B43B4E81 6BC7B0C0
070D1000 845DA34C 0014      B43B4E81 6BC74600
071D2000 81604FF2 0060      B43B4E81 6BC71180
050C0000 8012D90E 006C      B43B4E81 6BC61600
                  8012D9A0 FF22      B43B4E81 6BC2F280
                  8216F2D8 FF10      B43B4E81 6BBF1C80
CTL-XFER 00010248 CF00      B43B4E81 6B8BC580
CTL-XFER 000101E2 CF00      B43B4E81 6B8B8980
PIA-ADDR 00139790 CE00      B43B4E81 04777EC6
PIA-ADDR 00139790 CE00      B43B4E81 04694D06
PIA-ADDR 00139790 CE00      B43B4E81 04681F46
070D0000 8050364C 0014      B43B4E80 9E1C4D00
071D2000 80503578 0060      B43B4E80 9E1C1C40
071D2000 8050356A 0064      B43B4E80 9E1BE580
071D1000 8050351E 00FC      B43B4E80 9E1BB780
                  823F12D8 FF10      B43B4E80 9E1B6580
CTL-XFER 00002A00 CF00      B43B4E80 9E171F40
PIA-ADDR 00139790 CE00      B43B4E80 8AE9A986
PIA-ADDR 00139790 CE00      B43B4E80 8ADC9606
PIA-ADDR 00139790 CE00      B43B4E80 8ADB7386
PIA-ADDR 00139790 CE00      B43B4E80 10891884
PIA-ADDR 00139790 CE00      B43B4E80 107C0004
PIA-ADDR 00139790 CE00      B43B4E80 107B2044
070D0000 8050364C 0014      B43B4E7F A9F74E00
071D2000 80503578 0060      B43B4E7F A9F6F240
071D2000 8050356A 0064      B43B4E7F A9F6CB00
071D1000 8050351E 00FC      B43B4E7F A9F69D00
                  8217B2D8 FF10      B43B4E7F A9F632C0
CTL-XFER 00002A00 CF00      B43B4E7F A9F16880
PIA-ADDR 00139790 CE00      B43B4E7F 96B61248
PIA-ADDR 00139790 CE00      B43B4E7F 96ABE648
PIA-ADDR 00139790 CE00      B43B4E7F 96AAFE08
070D1000 84661056 0076      B43B4E7F 70E8E707
071D2000 81604FF2 0060      B43B4E7F 70E8B387
071D1000 845DA34C 0014      B43B4E7F 70E7E447
                  83F0903E FF02      B43B4E7F 70E7A487
071D0000 83F09030 00F6      B43B4E7F 70E75287
                  865386EC FFFC      B43B4E7F 70E70887
                  86516236 FFFE      B43B4E7F 70E6C947
                  86512CF4 FFFE      B43B4E7F 70E68A07
```

Figure 41. Collated Macro Trace Table (Part 3 of 5)

```
0257A000          RELCC  CLEM  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 70E63587
0257A000          UNFRC  CLEM  +???     FA-F8048005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 70E4E347
0257A000          FIWHC  CLEM  +???     FA-F8048005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 70629185
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 705E5D05
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW                                86512CF4 FFFE      B43B4E7F 705E1A85
0257A000          RELCC  CLEM  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 705DD045
0257A000          UNFRC  CLEM  +???     FA-F8046005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 705CA585
0257A000          FIWHC  CLEM  +???     FA-F8046005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 6FF4F045
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 6FF39F45
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW                                86512CF4 FFFE      B43B4E7F 6FF35285
0257A000          RELCC  CLEM  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 6FF30485
0257A000          UNFRC  CLEM  +???     FA-F8044005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 6FF1BA05
0257A000          FIWHC  CLEM  +???     FA-F8044005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 6F804640
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 6F7EE980
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW                                86512CF4 FFFE      B43B4E7F 6F7E95C0
0257A000          RELCC  CLEM  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 6F7E0000
0257A000          UNFRC  CLEM  +???     FA-F8042005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 6F7CAA40
0257A000          FIWHC  CLEM  +???     FA-F8042005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 6F19FE03
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 6F188F03
0257A000          RELCC  CELW  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 6F184D83
0257A000          UNFRC  CELW  +???     FA-004C0005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 6F174F03
0257A000          EFRETN CELV  +00003D2 RETURN TO CELW                                865153F4 FFFE      B43B4E7F 6F170443
0257A000          FILEC  CELV  +???     FA-004C0009 RECID-00E0 LVL-D2        071D0000 81604FF2 0016      B43B4E7F 6F09CC43
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELV                                86512CF4 FFFE      B43B4E7F 6F090B43
0257A000          FINWC  CLEM  +???     FA-004C0009 RECID-00E0 LVL-D2        071D0000 81604FF2 0026      B43B4E7F 6E9B5B84
0257A000          EFCALL CELV  +0000224 ENTER TO CLEM                                 86515246 FFFC      B43B4E7F 6E9A1B04
0257A000          CINFC  CELV  +???                                           071D2000 81604FF2 0060      B43B4E7F 6E99FB84
0257A000          EFCALL CELW  +00001C0 ENTER TO CELV                                 865161E2 FFFC      B43B4E7F 6E993944
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW                                86512CF4 FFFE      B43B4E7F 6E98E404
0257A000          FIWHC  CLEM  +???     FA-004C0005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 6E075446
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 6E05BF86
0257A000          RELCC  CELW  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 6E057846
0257A000          UNFRC  CELW  +???     FA-00440005 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 6E044586
0257A000          EFRETN CELV  +00003D2 RETURN TO CELW                                865153F4 FFFE      B43B4E7F 6DFC1C06
0257A000          FILEC  CELV  +???     FA-00440009 RECID-00E0 LVL-D2        071D0000 81604FF2 0016      B43B4E7F 6DEFE106
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELV                                86512CF4 FFFE      B43B4E7F 6DEF5146
0257A000          FINWC  CLEM  +???     FA-00440009 RECID-00E0 LVL-D2        071D0000 81604FF2 0026      B43B4E7F 6D868286
0257A000          EFCALL CELV  +0000224 ENTER TO CLEM                                 86515246 FFFC      B43B4E7F 6D854506
0257A000          CINFC  CELV  +???                                           071D2000 81604FF2 0060      B43B4E7F 6D852386
0257A000          EFCALL CELW  +00001C0 ENTER TO CELV                                 865161E2 FFFC      B43B4E7F 6D845E46
0257A000          EFRETN CLEM  +0000CD2 RETURN TO CELW                                86512CF4 FFFE      B43B4E7F 6D840D06
0257A000          FIWHC  CLEM  +???     FA-00440005 RECID-00E0 LVL-D1        071D0000 81604FF2 0028      B43B4E7F 6CDEF2C5
0257A000          EFCALL CELW  +00001A8 ENTER TO CLEM                                 865161CA FFFC      B43B4E7F 6CDD2905
0257A000          RELCC  CELW  +???     AD-0091A000 BKTYP-0051 LVL-D1        071D0000 81604FF2 0032      B43B4E7F 6CDCE045
0257A000          UNFRC  CELW  +???     FA-F403B805 RECID-00E0 LVL-D1        071D0000 81604FF2 0050      B43B4E7F 6CDB6845
0257A000          EFRETN CELV  +00003D2 RETURN TO CELW                                865153F4 FFFE      B43B4E7F 6CDB1F05
0257A000          FILEC  CELV  +???     FA-F403B809 RECID-00E0 LVL-D2        071D0000 81604FF2 0016      B43B4E7F 6CCC7505
--------          CXFRC  CHSZ* 0000FF81 ECB-DEFER ADR-026EFC00 FMT-A5       CTL-XFER 0000FF94 CF00      B43B4E7F 65BE0745
--------          D-CROS ----  ----      IS-14 P1-026EE400 P2-00000000       PIA-ADDR 00139790 CE00      B43B4E7F 1C63FB84
--------          D-CROS ----  ----      IS-14 P1-026EE000 P2-00000000       PIA-ADDR 00139790 CE00      B43B4E7F 1C56FC04
--------          D-CROS ----  ----      IS-14 P1-026EDC00 P2-00000000       PIA-ADDR 00139790 CE00      B43B4E7F 1C561544
023EE000          EXITC  CL23  +13A                                           070D0000 8050364C 0014      B43B4E7E B5BF2C02
023EE000          CINFC  CL23  +066                                           071D2000 80503578 0060      B43B4E7E B5BF1182
023EE000          CRETCS CL23  +058     TO ACTIVATE CL23                      071D2000 8050356A 0064      B43B4E7E B5BEFD02
023EE000          CEBIC  CL23  +00C                                           071D1000 8050351E 00FC      B43B4E7E B5BEED02
023EE000          ENTNC  -CP-  023EE2D4 ENTER TO CL23                                 823EE2D8 FF10      B43B4E7E B5BEC102
```

Figure 41. Collated Macro Trace Table (Part 4 of 5)

```
024FF000          EXITC  ARPK  +222                                        071D1000 84A97AE4 0014      B43B4E7E B5BCD1C2
024FF000          KEYRC  ARPK  +220                                        07CD1000 84A97AE2 0076      B43B4E7E B5BCBC42
024FF000          KEYCC  ARPK  +218                                        071D0000 84A97ADA 0074      B43B4E7E B5BCAB02
024FF000          CRETCS ARPK  +208     TO ACTIVATE ARPK                   071D0000 84A97ACA 0064      B43B4E7E B5BC8A42
024FF000          GETCC  ARPK  +224     AD-0091EBE0 TYPE-L2   LVL-D6       071D0000 84A97AE6 002C      B43B4E7E B5BB4282
024FF000          KEYRC  ARPK  +030                                        07CD0000 84A978F2 0076      B43B4E7E B5BB1242
024FF000          KEYCC  ARPK  +010                                        071D1000 84A978D2 0074      B43B4E7E B5BAF102
024FF000          ENTNC  -CP-  024FF2D4 ENTER TO ARPK                              824FF2D8 FF10      B43B4E7E B5BAA702
--------          CXFRC  CHSZ* 000029F1 ECB-DEFER ADR-026ED400 FMT-A5      CTL-XFER 00002A00 CF00      B43B4E7E B5B73D82
--------          CXFRC  CHSZ* 000029F1 ECB-DEFER ADR-026ED000 FMT-A5      CTL-XFER 00002A00 CF00      B43B4E7E B5B6B302
--------          D-CROS ----  ----      IS-14 P1-026EC000 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E A27C2084
--------          D-CROS ----  ----      IS-14 P1-026EBC00 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E A27BAF84
--------          D-CROS ----  ----      IS-14 P1-026EB800 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E A27AD404
--------          D-CROS ----  ----      IS-14 P1-026E9C00 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E 2884A087
--------          D-CROS ----  ----      IS-14 P1-026E9800 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E 28775E00
--------          D-CROS ----  ----      IS-14 P1-026E9400 P2-00000000      PIA-ADDR 00139790 CE00      B43B4E7E 28769740
023E5000          EXITC  ARPT  +0E8                                        071D0000 84A97C1A 0014      B43B4E7D C2E63183
023E5000          CRETCS ARPT  +0E0     TO ACTIVATE ARPT                   071D0000 84A97C12 0064      B43B4E7D C2E60A83
023E5000          BACKC  CYYM  +196     RETURN TO ARPT                             84A0D72A FF02      B43B4E7D C2E5ED03
023E5000          RELCC  CYYM  +192     AD-0091EE80 BKTYP-0011 LVL-DF      071D0000 84A0D724 0032      B43B4E7D C2E5B503
023E8000          EXITC  CSAG  +CB8                                        071D0000 83F84CBA 0014      B43B4E7D C2A28903
023E8000          UNFRC  CSAG  +C5A     FA-F40231D5 RECID-FC22 LVL-D1      071D2000 83F84C5C 0050      B43B4E7D C2A17303
023E8000          RELCC  CSAG  +C56     AD-00923000 BKTYP-0051 LVL-D1      071D2000 83F84C58 0032      B43B4E7D C2A14A83
023E8000          FIWHC  CSAG  +D18     FA-F40231D5 RECID-FC22 LVL-D1      071D2000 83F84D1A 0028      B43B4E7D C28F0B83
023E8000          GETCC  CSAG  +05A     AD-00918E80 TYPE-L1   LVL-DE       071D0000 83F8405C 002C      B43B4E7D C28EAB03
023E8000          CINFC  CSAG  +042                                        070D1000 83F84044 0060      B43B4E7D C28E9E43
023E8000          BACKC  CSAI  +4EA     RETURN TO CSAG                             83F854EE FF02      B43B4E7D C28E5C03
023E8000          UNFRC  CSAI  +4E2     FA-F40231D5 RECID-FC22 LVL-DF      070D0000 83F854E4 0050      B43B4E7D C28D1F03
023E8000          RELCC  CSAI  +4DE     AD-00918000 BKTYP-0051 LVL-DF      070D0000 83F854E0 0032      B43B4E7D C28CEE83
023E8000          CINFC  CSAI  +4A0                                        071D0000 83F854A2 0060      B43B4E7D C28CBB83
023EB000          EXITC  CL23  +13A                                        070D0000 8050364C 0014      B43B4E7D C1E7FA43
023EB000          CINFC  CL23  +066                                        071D2000 80503578 0060      B43B4E7D C1E7D903
023EB000          CRETCS CL23  +058     TO ACTIVATE CL23                   071D2000 8050356A 0064      B43B4E7D C1E7A583
023EB000          CEBIC  CL23  +00C                                        071D1000 8050351E 00FC      B43B4E7D C1E78D83
023EB000          ENTNC  -CP-  023EB2D4 ENTER TO CL23                              823EB2D8 FF10      B43B4E7D C1E75D43
023E8000          FIWHC  CSAI  +494     FA-F40231D5 RECID-FC22 LVL-DF      071D2000 83F85496 0028      B43B4E7D C1D5DB03
```

Figure 41. Collated Macro Trace Table (Part 5 of 5)

Table 30 on page 87 explains the headings and labels that appear in this section of the main storage dump.

Table 30. Headings in the Collated Macro Trace Table

Heading	Description
*COLLATED ECB MACRO TRACE ENTRIES	Each entry control block (ECB) in the TPF system maintains its own separate macro trace table. When a system error occurs, the trace table for every ECB is scanned, and the 256 most recent trace entries are extracted and displayed as shown previously. This is done to provide a general snapshot of system activity at the time the error occurred. A collated macro trace display is included for each I-stream that was active at the time of the system error dump. The trace entries in Figure 41 on page 82, are taken from the collated macro trace table for the main I-stream. This is not a wraparound trace table. There is no LAST indicator. The entries are displayed in reverse chronological order, beginning with the most recent. The 3 question marks (???) indicate the location is an ISO-C library. The ECB addresses shown on the left hand side of the dump are system virtual addresses. The asterisks shown next to the trace entries having ECB address X'004D6000' identify the failing ECB. (The ECB appears later in the dump, but since it will be dumped in its ECB virtual memory (EVM), its address is an ECB virtual address.) The collated macro trace has a line of dashes after the last entry that belongs to the first ECB in the trace. This indicates that there may be other entries that belong to this ECB that did not make the collated macro trace. The line of dashes helps the person reading the dump determine where the collated macro trace may stop being accurate for a specific ECB. Enter the ZSTRC command to control the system trace options. The following is a list of possible macro trace options: • MACRO starts tracing of all SVC macros associated with an ECB. **Note:** Fast link macros are not traced. • ENTER initiates the inclusion of ENTxC and BACKC program segments in the macro trace. • SYSLOG starts tracing of events formerly included in macro or I/O traces that are not associated with an ECB or an I/O device. Entries are generated in the collated macro trace table for the CXFRC macro and D-CROS when SYSLOG is set on. D-CROS indicates that something was taken off the cross list. In Figure 41 on page 82, the ZSTRC macro options of MACRO, ENTER, and SYSLOG are set on. See *TPF Operations* for more information about the ZSTRC command.
*MACRO	The name of the macro issued. Entering the ZSTRC command, you can select which macros you want to trace. For example you can trace all SVC macros associated with an ECB, all ENTxC and BACKC macros, and events not associated with an ECB. See *TPF System Macros* for more information about system macros and *TPF General Macros* for more information about general macros.
*PROG	The name of the program that issued the macro.
*LOCATION	Identifies the displacement in the program listing where the SVC occurred.

Table 30. Headings in the Collated Macro Trace Table (continued)

Heading	Description
*PARAMETERS	For the CXFRC, ENTER/BACK, SVC macros and D-CROS, parameter information is displayed. The parameter information is macro specific and indicates the status of the macros. The parameter information is self explanatory except for SYSLOG: • For D-CROS, the IS-n parameter describes the origin I-stream. While the P1 and P2 parameters describe the event-unique $CRISC parameters. • For CXFRC, the ECB-IMMED/DEFER parameter describes the create type; the ADR *xxxxxx* parameter describes the block address; the FMT *ff* parameter describes the format flat for a new ECB.
*SVC OLD PSW	If the dump occurred during a macro service routine, the address is the location of the last SVC. Otherwise, the address points to the next sequential instruction after the SVC. The following describes the SVC OLD PSW information for SYSLOG: • For D-CROS, PIA-ADDR *xxxxxxxx* is the SVC old PSW, which is the post interrupt address. • For CXFRC, CTL-XFER *xxxxxxxx* is the SVC old PSW, which is the address of the CXFRC statement.
*CODE	The SVC code identifying the macro. For ENTxC and BACKC macros, the code contains a X'FF' in the high-order byte.
*INX	The SVC index is displayed here if an indexed SVC was issued.
*TIME STAMP	The contents of the time-of-day (TOD) clock at the time the macro was issued.

Collated Input/Output (I/O) Trace Table

Figure 42 on page 89 shows an example of a collated I/O trace table.

```
*COLLATED LDEVBK I/O TRACE ENTRIES
*CIO MACRO       *SDA   *SUBCHANNEL STATUS WORD/PARMS        *I/O OLD PSW       *DORPRM   *INT HDLR *SCH  *TIME STAMP       *ISN
 MSDAC           000E   MDR-00025110 RC-0 PAM-80 OVRLY-NO    CP-MSDAC 80022E52                    000A  ABB4DDA8 92C71244  01
 SPNDC                                                       CP-SPNDC 800100EE                          ABB4DDA8 7EEE9D44  01
                 0463   10804007 01E74870 0C000000 00800000  070CC000 8006FC34  00A33400  0007706E  001D  ABB4DDA8 76F4C440  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA8 76444B40  01
                 0EE5   10804007 01E74470 0C000000 00800000  070CC000 8006FC34  00A34000  0007706E  0003  ABB4DDA8 7637F040  01
 SIOSC           0022   CPA-0016B098 CC-0 KEY-0 LPM-FF       CP-SIOSC 80026572                    0007  ABB4DDA8 75A8C844  01
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA8 7573B944  01
                 0022   10804007 0016B090 0C000FD5 00800000  070CC000 8006FC34  0016B080  00026000  0007  ABB4DDA8 75432444  01
 SIOSC           0022   CPA-0016B080 CC-0 KEY-0 LPM-FF       CP-SIOSC 80026572                    0007  ABB4DDA8 74A15E42  01
                 0022   00000011 00000000 80000000 00800000  070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA8 749DC742  01
                 0022   10804017 0016B0A8 0C400000 00800000  070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA4 A6A97444  01
 SIOSC           0022   CPA-0016B098 CC-0 KEY-0 LPM-FF       CP-SIOSC 80026572                    0007  ABB4DDA4 90506242  01
                 0022   10804017 0016B0A8 0C400000 00800000  070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA4 904BAA42  01
                 0EE5   10804007 01E74488 0C000000 00800000  070CC000 800700BC  00A33D00  0007706E  0003  ABB4DDA4 86D55640  04
 SIOSC           0EE5   CPA-01E74470 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 86C0B040  04
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A33300  0007706E  0003  ABB4DDA4 86BC6640  04
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A33800  0007706E  001D  ABB4DDA4 86A57341  02
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 86746640  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 86324642  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A33F00  0007706E  0003  ABB4DDA4 862E6041  04
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A33600  0007706E  001D  ABB4DDA4 86287D40  03
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 85A30B42  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 855F8542  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A33500  0007706E  0003  ABB4DDA4 855B3640  03
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A33700  0007706E  001D  ABB4DDA4 8554BC41  04
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 84D01F40  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 84A56540  01
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A32E00  0007706E  001D  ABB4DDA4 823FF843  03
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 81F6D144  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A32D00  0007706E  0003  ABB4DDA4 81E4F240  03
 SIOSC           0EE5   CPA-01E74430 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 816CE641  01
                 0EE5   10804007 01E74488 0C000000 00800000  070CC000 800700BC  00A34200  0007706E  0003  ABB4DDA4 81245940  03
 SIOSC           0EE5   CPA-01E74470 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 810D8C40  03
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A32B00  0007706E  0003  ABB4DDA4 81086340  03
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A33000  0007706E  001D  ABB4DDA4 8101F442  04
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 80732240  03
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A33A00  0007706E  0003  ABB4DDA4 806FCD40  03
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 80075244  02
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A33200  0007706E  001D  ABB4DDA4 8004E044  02
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7FC20142  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 8006FC34  00A32700  0007706E  0003  ABB4DDA4 7FBEE742  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 7EEA3741  02
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A32900  0007706E  001D  ABB4DDA4 7EE72341  02
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7E99BD42  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 7E77F242  01
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A32800  0007706E  001D  ABB4DDA4 7E51DE44  02
                 0EE5   10804007 01E74488 0C000000 00800000  070CC000 80070248  00A32C00  0007706E  0003  ABB4DDA4 7E088144  03
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 7DC6C942  01
 SIOSC           0EE5   CPA-01E74470 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7DB63C44  03
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A32F00  0007706E  0003  ABB4DDA4 7DB40F44  03
 SIOSC           0EE5   CPA-01E74430 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7D742842  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CE000 8006F658  00A32600  0007706E  0003  ABB4DDA4 7D71CF42  01
                 0463   10804007 01E74870 0C000010 00800000  070CC000 80070248  00A32300  0007706E  001D  ABB4DDA4 7D362144  04
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7D102142  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 7CE2AF42  01
                 0EE5   10804007 01E74470 0C000010 00800000  070CC000 80070248  00A32200  0007706E  0003  ABB4DDA4 7CBD7A44  03
 SIOSC           0EE5   CPA-01E74430 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 7B2A7742  01
                 0EE5   10804007 01E74470 0C000000 00800000  070CC000 80070248  00A32A00  0007706E  0003  ABB4DDA4 79B63F44  04
 SIOSC           0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    0003  ABB4DDA4 79039442  01
                 0463   10804007 01E74870 0C000000 00800000  070CC000 8006FC34  00A31F00  0007706E  001D  ABB4DDA4 78F9B042  01
 SIOSC           0022   CPA-0016B098 CC-0 KEY-0 LPM-FF       CP-SIOSC 80026572                    0007  ABB4DDA4 78A46842  01
 SIOSC           0463   CPA-01E74840 CC-0 KEY-0 LPM-80       CP-SIOSC 80079054                    001D  ABB4DDA4 7841E642  01
                 0022   10804007 0016B090 0C000FD5 00800000  070CC000 8006FC34  0016B080  00026000  0007  ABB4DDA4 78134342  01
 SIOSC           0022   CPA-0016B080 CC-0 KEY-0 LPM-FF       CP-SIOSC 80026572                    0007  ABB4DDA4 776B5A42  01
```

Figure 42. Collated I/O Trace (Part 1 of 2)

```
             0022   00000011 00000000 80000000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA4 77682B42  01
             0EE5   10804007 01E74470 0C000010 00800000   071D0000 8031AE84  00A32500  0007706E  0003  ABB4DDA4 54649140  01
SIOSC        0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      0003  ABB4DDA4 54054F40  01
             0EE5   10804007 01E74488 0C000000 00800000   070CC000 80070248  00A31C00  0007706E  0003  ABB4DDA2 7115EA40  03
SIOSC        0EE5   CPA-01E74470 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      0003  ABB4DDA2 70EF5E41  01
             0EE5   10804007 01E74470 0C000010 00800000   071D0000 8031D05C  00A31D00  0007706E  0003  ABB4DDA2 6EE42242  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A32400  0007706E  001D  ABB4DDA2 6ECCDF40  04
SIOSC        0EE5   CPA-01E74440 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      0003  ABB4DDA2 6E93CA42  01
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA2 6E6C1642  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 8006FC34  00A31B00  0007706E  001D  ABB4DDA2 6E5E2542  01
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA2 6D908542  01
             0EE5   10804007 01E74470 0C000000 00800000   070CC000 8006FC34  00A31900  0007706E  0003  ABB4DDA2 6D7A3142  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A32100  0007706E  001D  ABB4DDA2 6B186344  03
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA2 6A4E0C42  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A31400  0007706E  001D  ABB4DDA2 67E3C340  02
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA2 67866041  01
             0022   10804017 0016B0A8 0C400000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA0 2E686C40  01
SIOSC        0022   CPA-0016B098 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DDA0 15A05742  01
             0022   10804017 0016B0A8 0C400000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DDA0 159B6942  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A30F00  0007706E  001D  ABB4DDA0 0D63BE44  04
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA0 0D0A3644  02
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A30C00  0007706E  001D  ABB4DDA0 0D089244  02
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA0 0C9A2E44  02
             0463   10804007 01E74870 0C000010 00800000   070CC000 80070248  00A30800  0007706E  001D  ABB4DDA0 0C975444  02
SIOSC        0463   CPA-01E74840 CC-0 KEY-0 LPM-80         CP-SIOSC 80079054                      001D  ABB4DDA0 0BFD3840  01
             0463   10804007 01E74870 0C000010 00800000   070CC000 8006FC34  00A30600  0007706E  001D  ABB4DDA0 0BF40D40  01
SIOSC        0022   CPA-0016B098 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD9F FDFC0E40  01
             0022   10804007 0016B090 0C000FD3 00800000   070CC000 8006FC34  0016B080  00026000  0007  ABB4DD9F FDA1C640  01
SIOSC        0022   CPA-0016B080 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD9F FD304542  01
             0022   00000011 00000000 80000000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DD9F FD29DA42  01
             0022   10804017 0016B0A8 0C400000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DD98 92E80C40  01
SIOSC        0022   CPA-0016B098 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD98 7B210043  01
             0022   10804007 0016B090 0C000FE3 00800000   070CE000 8006F658  0016B080  00026000  0007  ABB4DD98 7AD99143  01
SIOSC        0022   CPA-0016B080 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD98 7A128442  01
             0022   00000011 00000000 80000000 00800000   071D0000 8031D05C  0016B098  00026000  0007  ABB4DD98 7A0F6342  01
             0022   10804017 0016B0A8 0C400000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DD96 929BAB41  01
SIOSC        0022   CPA-0016B098 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD96 7B61AE43  01
             0022   10804017 0016B0A8 0C400000 00800000   070CC000 8006FC34  0016B098  00026000  0007  ABB4DD96 7B5CA443  01
SIOSC        0022   CPA-0016B098 CC-0 KEY-0 LPM-FF         CP-SIOSC 80026572                      0007  ABB4DD96 539E6441  01
           * 0022   10804017 0016B0A8 0C400000 00800000   040CC000 81FDA332  0016B098  00026000  0007  ABB4DD96 5273E743  01
RESMC               RESUME PSW-070CC000 8006F648           CP-RESMC 8006F9F0                      ABB4DD96 52721643  01
RESMC               RESUME PSW-040CC000 800105B4           CP-RESMC 800105B4                      ABB4DD96 21EFD143  01
TIOSC        0580   CC-0 TYPE-PEEK TIO-0002509C            CP-TIOSC 80022444                      001E  ABB4DD95 2CB65043  01
```

Figure 42. Collated I/O Trace (Part 2 of 2)

Table 31 explains the headings and labels that appear in this section of the main storage dump.

Table 31. Headings in the Collated Input/Output (I/O) Trace Table

Heading	Description
*COLLATED LDEVBK I/O TRACE ENTRIES	Each CIO LDEVBK in the TPF system maintains its own separate I/O trace table. When a system error occurs, the trace table for every LDEVBK is scanned, and the 256 most recent trace entries are extracted and displayed as shown previously. This is done to provide a general snapshot of I/O activity at the time the error occurred. A collated I/O trace display is formatted for the main and Multi-Processor Interconnect Facility (MPIF) I-streams. All application I-stream I/O entries, excluding MPIF I/O, are merged into the main I-stream I/O trace table. The MPIF I-stream I/O trace table contains only MPIF entries. The trace entries in the preceding example are taken from the collated I/O trace table for the main I-stream. This is not a wraparound trace table. There is no LAST indicator. The entries are displayed in reverse chronological order, beginning with the most recent. The asterisks shown next to certain I/O interrupts identify interrupts that were queued in CIO when preemptive I/O was last performed.
*CIO MACRO	The name of the CIO macro issued.

Table 31. Headings in the Collated Input/Output (I/O) Trace Table (continued)

Heading	Description
*SDA	The symbolic device address (SDA) of the device that is the target of the CIO macro, or which is presenting the I/O interrupt.
*SUBCHANNEL STATUS WORD/PARMS	For an I/O interrupt, this field contains the 12-byte subchannel status word, and the first 4 bytes of the extended status word, containing the last path used mask (LPUM) in byte 1. For a CIO macro, this field displays the applicable parameters associated with the macro.
*I/O OLD PSW	For an I/O interrupt, the contents of the current program status word (PSW) when the interrupt occurred. For a CIO macro, the return address identifying the location of the macro call.
*DORPRM	The value of the I/O request-unique parameter returned to the device handlers by CIO when an I/O interrupt occurs.
*INT HDLR	The address of the device handler to which CIO passes control after handling this interrupt.
*SCH	The subchannel number of the device.
*TIME STAMP	The contents of the time-of-day (TOD) clock at the time the I/O trace event occurred.
*ISN	The number of the I-stream on which the event took place. A zero in this field indicates that the event occurred early in IPL prior to initialization of the I-stream number field in the prefix page.

Prefix Page

Figure 43 on page 92 shows an example of the prefix page.

```
*PREFIX PAGE - ADDRESSES ARE ABSOLUTE
013DE000  RNP 000C0000 00000F80       ROP 00000000 00FF0000           47F00F8A 47F00F80       EOP 070EC000 800616B8
013DE020  SOP 071D0000 801A4A30       POP 040C0000 800468A4       MOP 00000000 00000000       IOP 060C1000 800466B4
013DE040      00000000 00000000           00000000 00000000           00000000 00000000       ENP 040CC000 8000C420
013DE060  SNP 050C0000 8001F200       PNP 000CC000 80034008       MNP 0008C000 8002F000       INP 040CC000 813DC468
013DE080      00000000 00021202 ECD   SCD 000200B6 00040040 PCD       0030F000 00030000           00000000 00000000
013DE0A0      00000000 00000000           00000000 00000000           00000000 00000000       IOS 00010012 013CD000 IOR
013DE0C0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE0E0      00000000 00000000       MCD 00000000 00000000           00000000 00000000           00000000 00000000
013DE200  IRB 00804007 00049048           0C000000 00800000       SNS 00000000 00000000           00000000 00000000
013DE220      00000000 00000000           00000000 00000000           0C000018 04200400           040CC000 800953A0
013DE240      80045680 00000000           00000004 00048620           001103FC 001113C0           00035C50 0003843E
013DE260      80046678 00000420           000465C0 00048730           00048600 00096488           800949E6 00000000
013DE280      8004688E 00000004           00000000 FFFFFFFE           013CC000 001113C0           00035C50 0003843E
013DE2A0      80046678 00000420           000465C0 813DC468           013D9000 00001000           00002000 0015ED50
013DE2C0      813DC0B0 813DBAD0           013DB000 013D9000           813DBED8 813DB960           813DC230 813DB448
013DE2E0      813DB5A0 813DB640           813DCBB8 813DCD64           813DD260 813DB720           043DC468 00031D40
013DE300  ISA FF000001 00000001 MSA       0001F478 0002747C           00000000 0059E060 RCV   XPF 013E0000 013DE000 RPF
013DE320  SSV 00159F90 00157750 FSV   ISV 00161390 0015EF90 ESV   PSV 0015EE30 00160EA8 MSV       00161390 00161B90
013DE340      0015DF90 0015BF90           0015AF90 0015FF90           01138004 0113804C           00000000 00160EA8 MSR
013DE360  CID FF114482 30900000       X12 813DF000 01131878 LOG   DMT 00040000 0059E000           0059E020 0059E040
013DE380      0059E060 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE3A0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE400      00000000 000619F0           00000200 00061A50           00007D98 00007D9C           00002280 00000002
013DE420      001A4A10 004D6000           00064F38 00001000           00002000 00159F90           000619F0 00061120
013DE440  PRV 00000004 00048620           001103FC 001113C0           00035C50 0003843E           80046678 00000420
013DE460      000465C0 00048730           00048600 00001000           00002000 0015ED50           80045680 00000000
013DE480      1CB0BEE0 1C000000           071D0000 801A4A2A       IPP 040CC000 813DCEEA           1CB0BEE0 1C000000
013DE4A0      1CB0BEE0 10000000           00000000 00000000           00000000 00000000           1CB08000 1C000000
013DE4C0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 0000000C
013DE4E0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE520      00000000 00000000           00000000 00000000           00000000 001577C0           0009D960 00000000
013DE540      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE680  CPR 00000000 00000000       RSV 00000000 00000000           00000000 00000000           00000000 00000000
013DE6C0      00000000 00000000       ARM 040C2000 8003FDFE       PTR 00000000 00000000       MCR 040C0000 800468A4
013DE6E0  RVR 040C0000 8004688E       RVV 00000004 00048620           001103FC 001113C0           00035C50 0003843E
013DE700      80046678 00000420           000465C0 00048730           00048600 00001000           00002000 0015ED50
013DE720      80045680 0004688E           00000000 00000000           00000000 00000000           00000000 00000000
013DE740      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DE780      00000000 00000000       OLD A62D139E 1DC56111       NEW A62D139E 1DC4B311       IDL 000619F0 00061120 CPL
013DE7A0      32004000 00000000           00000000 00000000           A62D139A F3DFD013           A62D139B 04F8F413
   .             .        .                .        .                  .        .                  .        .
   .             .        .                .        .                  .        .                  .        .
   .             .        .                .        .                  .        .                  .        .
013DEF80      50F00F9C 58F00F94           07FF50F0 0F9C58F0           0F9807FF 000340E8           00034118 00000000
013DEFA0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
013DEFE0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
```

Figure 43. Prefix Page

Table 32 explains the headings and labels that appear in this section of the main storage dump.

Table 32. Headings in the Prefix Page

Heading	Description
*PREFIX PAGE – ADDRESSES ARE ABSOLUTE	A prefix page is displayed for each I-stream that was active at the time of the dump. Each prefix page shows the I-stream's unique page 0 contents at the time the error occurred. The area is labeled throughout with dump labels used to identify specific fields. See the DCTPFX data macro for more information.

Branch Trace Table

Figure 44 on page 93 shows an example of the branch trace table.

```
*BRANCH TRACE TABLE
013DF000      800452B8 800452B8          800452B8 800452B8          800452B8 800465C0          813DCD64 813DCD64
013DF020      813DCBB8 80060D66          8008A8C2 813DB960          8005C318 80089408          80060E44 80060E44
013DF040      80060E44 80060E44          80060E44 80060E44          80060F00 80060CE8          80060F00 80060CE8
013DF060      80060F00 80060CE8          80060F00 80060CE8          80060F00 80060CE8          80065770 80060D66
013DF080      80060D66 8008A8C2          813DB960 8005C318          80089408 80060E44          80060E44 80060E44
013DF0A0      80060E44 80060E44          80060E44 80060F00          80060CE8 80060F00          80060CE8 80060F00
013DF0C0      80060CE8 80060F00          80060CE8 80060F00          80060CE8 80065770          80060D66 80060D66
013DF0E0      8008A8C2 813DB960          8005C318 80089408          80060E44 80060E44          80060F00 80060CE8
013DF100      80005F2E 800C0608          80060CE8 800C0758          80006448 80006844          800C1BB0 800C0758
013DF120      80006B74 800C0608          8000CBE0 800C22F8          80060E44 80060E44          80060E44 80060E44
013DF140      80060E44 80060F00          80060CE8 80060F00          80060CE8 80060F00          80060CE8 80060F00
   .             .       .                .       .                  .       .                  .       .
   .             .       .                .       .                  .       .                  .       .
   .             .       .                .       .                  .       .                  .       .
013DFDC0      8000A2EA 8000AACE          800BF9B0 800063E8          800066C4 800C0608          80006B00 80060E44
013DFDE0      800C0250 800C0250          80003758 800BFB70          800521A0 800517C4          80060D66 80060F00
013DFE00      80060CE8 80066330          80067E8E 813DC0B0          80006448 80006844          800C1BB0 800C14C0
013DFE20      800C0758 80006B74          80001000 800BFB70          800BFB70 80001000          80001000 80001000
013DFE40      800C0250 800BFB70          80001000 8000A4C0          80001000 80001000          800BFB70 800C2848
013DFE60      80001000 800BFB70          800BFB70 800BFB70          80001000 800C0250          800C0250 800C0250
013DFE80      80001000 800C0250          800C1960 8000A4A4          80006308 800065C4          80006844 800C1BB0
013DFEA0      80006B00 800C0EF0          800C14C0 80060E44          800C0250 800C0250          800C22F8 80060F00
013DFEC0      80060CE8 8000215A          800C1960 800C0EF0          80012594 800C1960          800C1960 8001A7D8
013DFEE0      8000E8E6 813DC0B0          80012594 80012594          80012594 800BFFA8          800C22F8 80067D8E
013DFF00      80060E44 80060F00          80060CE8 80053D8A          800C0250 800BFB70          800C0250 80004C00
013DFF20 LAST 80060E44 813DCD64          80044CC0 813DD260          800465C0 813DCD64          813DCD64 813DCBB8
013DFF40      800465C0 813DCD64          813DCD64 813DCBB8          8003C898 8003BB18          800452B8 8003FBA8
013DFF60      800452B8 8003FBA8          8003FBA8 800452B8          8003FBA8 800452B8          8003FBA8 800452B8
013DFF80      8003FBA8 800452B8          8003FBA8 800452B8          8003CFC4 8003D038          8003FBA8 800452B8
013DFFA0      8003D052 8003FBA8          800452B8 8003FBA8          800452B8 800465C0          813DCD64 813DCD64
013DFFC0      813DCBB8 80040148          813DCD64 813DCD64          8003FBA8 800452B8          800452B8 800452B8
013DFFE0      800452B8 800452B8          800452B8 800465C0          813DCD64 813DCD64          813DCBB8
```

Figure 44. Branch Trace Table

Table 33 explains the headings and labels that appear in this section of the main storage dump.

Table 33. Headings in the Branch Trace Table

Heading	Description
*BRANCH TRACE TABLE	When branch tracing is active, an entry is made in a 4 KB wraparound trace table for each execution of the branch instructions BALR, BASR, BASSM, and BAKR. Each 4-byte entry in the trace table identifies the destination address of the branch, as well as a 31-bit addressing mode indication in the high-order bit. The branch trace option is controlled by the ZSTRC command. The LAST indicator identifies the most recent branch trace entry.

ECB Virtual Memory (EVM)

Figure 45 on page 94 shows an example of the ECB virtual memory.

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

ENTRY REFERENCED VIA R9 - DUMP OF ECB VIRTUAL MEMORY FOLLOWS


*ENTRY BLK, GENERATED BY 1052/3215 ENTRY
00902000  CHW 00000000 80130A78 BAD
00902008 W000 C4C5C3C2 010000C2      W008 80B00000 00000000      W016 00000000 00000000      W024 00000000 00000000
   BCD         D E C B        B
00902028 W032 00000000 00000000      W040 01000000 00000000      W048 00000000 E2D4D7C2      W056 010000C2 80B00000
   BCD                                                                      S M P B                B
00902048 W064 00000000 00000000      W072 00008400 04000000      W080 E3C5E2E3 00000000      W088 00000000 05961458
   BCD                                                                    T E S T
00902068 W096 00000000 00000000      WSW1 00000000 00000000
00902078  FA0 00000000 00000000       FA1 00000000 00000000       FA2 00000000 00000000       FA3 00000000 00000000
00902098  FA4 00000000 00000000       FA5 00000000 00000000       FA6 00000000 00000000       FA7 00000000 00000000
009020B8  FA8 00000000 00000000       FA9 00000000 00000000       FAA 00000000 00000000       FAB 00000000 00000000
009020D8  FAC 00000000 00000000       FAD 00000000 00000000       FAE 00000000 00000000       FAF 00000000 00000000
009020F8  FAP 00FF0000 FC0A048D
00902100  CR0 00919E80 0021017D       CR1 00922000 00010FFF       CR2 00000000 00010000       CR3 00000000 00010000
00902120  CR4 00000000 00010000       CR5 00000000 00010000       CR6 00000000 00010000       CR7 00000000 00010000
00902140  CR8 00919C00 0001017D       CR9 00000000 00010000       CRA 00000000 00010000       CRB 00000000 00010000
00902160  CRC 00000000 00010000       CRD 00000000 00010000       CRE 00000000 00010000       CRF 00000000 00010000
00902180  CRP 066C1020 00010000
00902188  FX0 00000000 00000000       FX1 00000000 00000000       FX2 00000000 00000000       FX3 00000000 00000000
009021A8  FX4 00000000 00000000       FX5 00000000 00000000       FX6 00000000 00000000       FX7 00000000 00000000
009021C8  FX8 00000000 00000000       FX9 00000000 00000000       FXA 00000000 00000000       FXB 00000000 00000000
009021E8  FXC 00000000 00000000       FXD 00000000 00000000       FXE 00000000 00000000       FXF 00000000 00000000
00902208  SUD 00000000 00000000           00000000 00000000       SUP 0000FF00 FF000000           00000000 00213E60
00902228      B43B4E81 B77C2803           00000000 00000000           00000000 01CFF880           00000014 00000000
00902248      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00902288      00904E3C 00000000           00000000 00000000       GLA 001CA000 001DA000 GLY
009022A0  R14 8012D9A0 00000000 R15
009022A8  R0  01605290 00A01C40 R1    R2  016051F8 00922720 R3    R4  066C3BF8 00A01E00 R5    R6  066C4870 8464561A R7
009022C8  R8  066C1020
009022CC  CPL 009022D0 00904A68 FPL
009022D4  CXR 8012D90E C3E5C1F1       ISN 00010000 00008400       STP 00000000 000062F0 CNF   CSC 00000000 00000000 CSP
009022F4  EHT 00000000 000FDEE8 TRT   TCA 00000000 00000000 STK   VCT 00320000 00902288 DET       00000000 00000000
00902314      00000000 00000000           00FF0000 00000000       CRS 00000000 000000C2 CPX
0090232C  SBI FFFFFFFF FF00FF00 DBI   SSU FF000000
00902338  PSW 071D1000 8465C704       CTL 00000000 39000000       REC 00010000 00000000           00000040 0001925A TRC
00902358      0080E819 58790000       TTA 00000000 00000000 AUT
00902368  URA 00000000 00000000 URB   UR0 00000000 00000000 UR1   UR2 00000000 00000000 UR3   UR4 00000000 00000000 UR5
00902388  UR6 00000000 00000000 UR7
00902390 X000 01008400 8000E3C5      X008 E2E3C3E5 E9E90004      X016 00000100 00000000      X024 00919E80 00919EAB
   BCD                     T E            S T C V  Z Z
009023B0 X032 00919E80 05961458      X040 00000880 00000015      X048 001C3FF8 00000000      X056 00000000 00000000
   BCD                                                                    8
009023D0 X064 001C3FF8 05961458      X072 00000000 C3E5C1F3      X080 00000000 00000000      X088 804A9E60 00000004
   BCD              8                              C V A 3                                          -
009023F0 X096 00000000 00000000      XSW0 02000000 00000000           00000000 00000000           00000000 000000B5
00902410      FFFFFFFF 04588198           00000000 00000000           00000000 00000000           00000000 00000000
00902430      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00902530      00000000 00000000           00000000 00000000
00902540 USA0 00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00903000  AV1 0293D7A0 00000000 AV2    AV4 02940A20 0305200F STD   SGT 03052000 03056800 PGT   NVA 000000B0 023FA000 SVA
00903020  EVA 00902000 02940C00 FRM   EHT 02A697F4 03052004 GLB   GLX 03052188 00000000 FG2   GMT 03053800 03053834 GMH
00903040  TFC 000D00A0 10000A00           00200001 00A01000 HFC   HTR 00A01000 00000000 ASC   EVC 00000000 00000000 RET
00903060  TOD B43B4E81 95879588           00000001 00000000           04586028 00000000       GSP 03052800 03052834 GSH
00903080  ISO 01600010 04000007           0001000D 03056800           00010000 00000000           00000000 00000000
009030A0      00000000 00000000           00000000 00000000           00000000 03052600 PCB   TCB 00000000 00000000 FCH
009030C0  BCH 00000000 03052400 TSA   STA 01600000 03053800 EGS   TYP 00000000 00000000       TMC 00000000 00A01710
009030E0      B43B4E81 95879588       DEC 00922000 00A018C0           0000FF00 00000000           00000000 00000000
00903100      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00903120      00000000 00A018C0       AVM 00000000 00A01FC0           00000000 00000000 PER       00000000 00000000
00903140      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00903B00  USR 00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00903C00  SVT 00000000 00000000           823FA000 823FB000           823FC000 00000000           00000000 00000000
```

Figure 45. ECB Virtual Memory (EVM) (Part 1 of 5)

```
00903C20      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00903C60      00000000 0293D700           00000000 00000000           00000000 00000000           00000000 00000000
00903C80      00000000 00000000           0293E000 00000000           00000000 00000000           00000000 00000000
00903CA0      00000000 00000000           00000000 00000000           02940A00 00000000           00000000 00000000
00903CC0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00904000  MTH 009044F8 00000030           00904018 00904A38           023FC018 023FCA38
*ECB MACRO TRACE  *MACRO *PROG *LOCATION   *PARAMETERS                          *SVC OLD PSW    *CODE *INX   *TIME STAMP
00904018      IS01 MALOC  CJ00  +022     SIZE-00001068 BLOCK-00A0CF70           071D1000 84707CB4 003B 0005 B43B4E81 B6FAF101
00904048      IS01 FREEC  CJ00  +026     BLOCK-00A0CF70                         071D1000 846E6D38 003B 0007 B43B4E81 B6FC7F01
00904078      IS01 FREEC  CJ00  +026     BLOCK-00A07598                         071D1000 8470729C 003B 0007 B43B4E81 B6FCD581
009040A8      IS01 DECBC  CJ00  +0C0     TYPE-RELEASE D-00922620                071D1000 847D097A 003B 0035 B43B4E81 B6FD5C81
009040D8      IS01 DECBC  CJ00  +0C0     TYPE-RELEASE D-009226A0                071D0000 847D097A 003B 0035 B43B4E81 B6FD8581
00904108      IS01 FREEC  CJ00  +096     BLOCK-00A02208                         071D1000 846E3FB0 003B 0007 B43B4E81 B6FDAE41
00904138      IS01 FINWC  QDCB  +???     FA-FC14000D RECID-FC2A LVL-DE          071D0000 81604FF2 0026      B43B4E81 B6FEF041
00904168      IS01 RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE          071D0000 81604FF2 0032      B43B4E81 B75D8103
00904198      IS01 MALOC  QDCB  +???     SIZE-00000007 BLOCK-00A02460           071D1000 845ED068 003B 0005 B43B4E81 B75E2303
009041C8      IS01 FREEC  QDCB  +???     BLOCK-00A02460                         071D1000 845DAA70 003B 0007 B43B4E81 B75F5583
009041F8      IS01 MALOC  QDCB  +???     SIZE-00000108 BLOCK-00A02370           071D1000 845ED068 003B 0005 B43B4E81 B75FA003
00904228      IS01 GSWBC  CGSW  +026                                            071D1000 84677108 003B 002B B43B4E81 B7600A43
00904258      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B7606443
00904288      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B760B043
009042B8      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B760E643
009042E8      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B760F943
00904318      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B7613F83
00904348      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B7616A83
00904378      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B761C903
009043A8      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B761EDC3
009043D8      IS01 MALOC  QDCB  +???     SIZE-00001000 BLOCK-00A0CFD8           071D1000 845ED068 003B 0005 B43B4E81 B762EF03
00904408      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B764E683
00904438      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B7650F43
00904468      IS01 WTOPC  QDCB  +???                                            071D0000 816059AA 00F0      B43B4E81 B7658983
00904498      IS01 DLAYC  QDCB  +???                                            071D1000 8465C704 000C      B43B4E81 B7669743
009044C8 LAST IS01 DECBC  QDCB  +050     TYPE-CREATE  D-00922720                071D1000 866C85BA 003B 0035 B43B4E81 B77E5BC3
009044F8      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B2070C87
00904528      IS01 CINFC  QDCB  +???                                            071D2000 81604FF2 0060      B43B4E81 B2076A47
00904558      IS01 KEYRC  QDCB  +???                                            070D1000 84661056 0076      B43B4E81 B2078F07
00904588      IS01 CENVC  QDCB  +???                                            071D1000 8462E46C 003B 0021 B43B4E81 B20854C7
009045B8      IS01 MALOC  QDCB  +???     SIZE-0000012C BLOCK-00A024F0           071D1000 845ED068 003B 0005 B43B4E81 B208B987
009045E8      IS01 MALOC  QDCB  +???     SIZE-00000030 BLOCK-00A024B0           071D1000 845ED068 003B 0005 B43B4E81 B2091E87
00904618      IS01 MALOC  QDCB  +???     SIZE-00000015 BLOCK-00A02488           071D1000 845ED068 003B 0005 B43B4E81 B20972C7
00904648      IS01 FINWC  QDCB  +???     FA-FC140001 RECID-FC2A LVL-DE          071D0000 81604FF2 0026      B43B4E81 B20A8747
00904678      IS01 RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE          071D0000 81604FF2 0032      B43B4E81 B4B2CDC4
009046A8      IS01 CALOC  CJ00  +028     SIZE-00000270 BLOCK-00A02208           071D1000 847EA762 003B 0006 B43B4E81 B4B48984
009046D8      IS01 DECBC  CJ00  +098     TYPE-CREATE  D-00922520                071D1000 847D07D2 003B 0035 B43B4E81 B4B51784
00904708      IS01 DECBC  CJ00  +098     TYPE-CREATE  D-009225A0                071D1000 847D07D2 003B 0035 B43B4E81 B4B54444
00904738      IS01 FINWC  CJ00  +0E2     FA-000000000014582D ID-FC16 D-00922520 071D1000 847F3198 0026      B43B4E81 B4B59A84
00904768      IS01 MALOC  CJ00  +022     SIZE-00000B30 BLOCK-00A07468           071D1000 847F9FE4 003B 0005 B43B4E81 B584C787
00904798      IS01 RELCC  CJ00  +046     AD-0092C000 BKTYP-0051 LVL-DF          071D1000 847FA0F4 0032      B43B4E81 B585A247
009047C8      IS01 MALOC  CJ00  +022     SIZE-00001068 BLOCK-00A0CF70           071D1000 84707CB4 003B 0005 B43B4E81 B5864A47
009047F8      IS01 FREEC  CJ00  +026     BLOCK-00A0CF70                         071D1000 846E6D38 003B 0007 B43B4E81 B587E787
00904828      IS01 FREEC  CJ00  +026     BLOCK-00A07468                         071D1000 8470729C 003B 0007 B43B4E81 B5884187
00904858      IS01 DECBC  CJ00  +0C0     TYPE-RELEASE D-00922520                071D1000 847D097A
003B 0035 B43B4E81 B588CF07
00904888      IS01 DECBC  CJ00  +0C0     TYPE-RELEASE D-009225A0                071D0000 847D097A 003B 0035 B43B4E81 B588F807
009048B8      IS01 FREEC  CJ00  +096     BLOCK-00A02208                         071D1000 846E3FB0 003B 0007 B43B4E81 B5892A47
009048E8      IS01 FINWC  QDCB  +???     FA-FC140005 RECID-FC2A LVL-DE          071D0000 81604FF2 0026      B43B4E81 B58A7C87
00904918      IS01 RELCC  QDCB  +???     AD-0092C000 BKTYP-0051 LVL-DE          071D0000 81604FF2 0032      B43B4E81 B643FC87
00904948      IS01 CALOC  CJ00  +028     SIZE-00000270 BLOCK-00A02208           071D1000 847EA762 003B 0006 B43B4E81 B645AB47
00904978      IS01 DECBC  CJ00  +098     TYPE-CREATE  D-00922620                071D1000 847D07D2 003B 0035 B43B4E81 B6462B07
009049A8      IS01 DECBC  CJ00  +098     TYPE-CREATE  D-009226A0                071D1000 847D07D2 003B 0035 B43B4E81 B6465687
009049D8      IS01 FINWC  CJ00  +0E2     FA-0000000000145831 ID-FC16 D-00922620 071D1000 847F3198 0026      B43B4E81 B646AC87
00904A08      IS01 MALOC  CJ00  +022     SIZE-00000A00 BLOCK-00A07598           071D1000 847F9FE4 003B 0005 B43B4E81 B6F98781
00904A38      IS01 RELCC  CJ00  +046     AD-0092C000 BKTYP-0051 LVL-DF          071D1000 847FA0F4 0032      B43B4E81 B6FA5301
00904A68  L0  00904A84 009022D0           004A9D40 804AA006           062E55E0 FF00FF00           00000000 00904AA0 L1
00904A88      00904A68 065DF020           01605030 062F9E00           FF00FF00 80000000       L2  00904ABC 00904A84
00904AA8      00000000 00000000           00000000 00000000           00000000 00904AD8 L3        00904AA0 00000000
00904AC8      00000000 00000000           00000000 00000000       L4  00904AF4 00904ABC           00000000 00000000
```

Figure 45. ECB Virtual Memory (EVM) (Part 2 of 5)

```
00904AE8      00000000 00000000           00000000 00904B10 L5        00904AD8 00000000           00000000 00000000
00904B08      00000000 00000000       L6  00904B2C 00904AF4           00000000 00000000           00000000 00000000
00904B28      00000000 00904B48 L7        00904B10 00000000           00000000 00000000           00000000 00000000
00904B48  L8  00904B64 00904B2C           00000000 00000000           00000000 00000000           00000000 00904B80 L9
00904B68      00904B48 00000000           00000000 00000000           00000000 00000000      L10  00904B9C 00904B64
00904B88      00000000 00000000           00000000 00000000           00000000 00904BB8 L11       00904B80 00000000
00904BA8      00000000 00000000           00000000 00000000      L12  00904BD4 00904B9C           00000000 00000000
00904BC8      00000000 00000000           00000000 00904BF0 L13       00904BB8 00000000           00000000 00000000
00904BE8      00000000 00000000       L14 00904C0C 00904BD4           00000000 00000000           00000000 00000000
00904C08      00000000 00904C28 L15       00904BF0 00000000           00000000 00000000           00000000 00000000
00904C28  L16 00904C44 00904C0C           00000000 00000000           00000000 00000000           00000000 00904C60 L17
00904C48      00904C28 00000000           00000000 00000000           00000000 00000000      L18  00904C7C 00904C44
00904C68      00000000 00000000           00000000 00000000           00000000 00904C98 L19       00904C60 00000000
00904C88      00000000 00000000           00000000 00000000      L20  00904CB4 00904C7C           00000000 00000000
00904CA8      00000000 00000000           00000000 00904CD0 L21       00904C98 00000000           00000000 00000000
00904CC8      00000000 00000000       L22 00904CEC 00904CB4           00000000 00000000           00000000 00000000
00904CE8      00000000 00904D08 L23       00904CD0 00000000           00000000 00000000           00000000 00000000
00904D08  L24 00904D24 00904CEC           00000000 00000000           00000000 00000000           00000000 00904D40 L25
00904D28      00904D08 00000000           00000000 00000000           00000000 00000000      L26  00904D5C 00904D24
00904D48      00000000 00000000           00000000 00000000           00000000 00904D78 L27       00904D40 00000000
00904D68      00000000 00000000           00000000 00000000      L28  00904D94 00904D5C           00000000 00000000
00904D88      00000000 00000000           00000000 00904DB0 L29       00904D78 00000000           00000000 00000000
00904DA8      00000000 00000000       L30 00904DCC 00904D94           00000000 00000000           00000000 00000000
00904DC8      00000000 00904DE8 L31       00904DB0 00000000           00000000 00000000           00000000 00000000
00904DE8  L32 00904E04 00904DCC           00000000 00000000           00000000 00000000           00000000 00904E20 L33
00904E08      00904DE8 00000000           00000000 00000000           00000000 00000000      L34  00000000 00904E04
00904E28      00000000 00000000           00000000 00000000           00000000 00904E60 DBT       00000000 00000000
00904E48      00010000 00000000           00000000 00000000           00000000 00000070           00904E84 00904E3C
00904E68      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00904EA8
00904E88      00904E60 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00904EA8      00904ECC 00904E84           00000000 00000000           00000000 00000000           00000000 00000000
00904EC8      00000000 00904EF0           00904EA8 00000000           00000000 00000000           00000000 00000000
00904EE8      00000000 00000000           00904F14 00904ECC           00000000 00000000           00000000 00000000
00904F08      00000000 00000000           00000000 00904F38           00904EF0 00000000           00000000 00000000
00904F28      00000000 00000000           00000000 00000000           00904F5C 00904F14           00000000 00000000
00904F48      00000000 00000000           00000000 00000000           00000000 00904F80           00904F38 00000000
00904F68      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00904F5C
00904F88      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
00904FA8      00000000 00000000           00000000 00000000           00000000 00000000      SPT  016051F8 00000002
00904FC8      00000000 00000000           00000000 00000000       PAT 045881F8 00000000           00000000 00000000
00904FE8      00000000 00000000           00000000 00000000           00000000 00000000

*DECB FRAME
00922000 +000  D E C B  $ $ $ $      +008 00000000 009227A0      +010 00000000 00000000      +018 00000000 00000000
00922020 +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038 0092C000 00010FFF
00922040 +040 FC160100 00000000      +048 0014582D 00000000      +050 00000000 00000000      +058 00000000 00000000
00922060 +060 00000000 00000000      +068 00000000 00000000      +070 DECBDECB 00922000      +078 00000000   3E0000
00922080 +080  C J 0 0 00000000      +088 00000000 00000000      +090 FF00FF00 00000000      +098 00000000 00000000
009220A0 +0A0 00000000 00000000      +0A8 00000000 00000000      +0B0 00000000 00000000      +0B8 00000000 00010000
009220C0 +0C0 00000000 00000000      +0C8 00000000 00000000      +0D0 00000000 00000000      +0D8 00000000 00000000
009220E0 +0E0 00000000 00000000      +0E8 00000000 00000000      +0F0 DECBDECB 00922000      +0F8 00000000   000000
00922100 +100  C J 0 0 00000000      +108 00000000 00000000      +110 FF00FF00 00000000      +118 00000000 00000000
00922120 +120 00000000 00000000      +128 00000000 00000000      +130 00000000 00000000      +138 0092C000 00010FFF
00922140 +140 FC160100 00000000      +148 0014582D 00000000      +150 00000000 00000000      +158 00000000 00000000
00922160 +160 00000000 00000000      +168 00000000 00000000      +170 DECBDECB 00922000      +178 00000000   3E0000
00922180 +180  C J 0 0 00000000      +188 00000000 00000000      +190 FF00FF00 00000000      +198 00000000 00000000
009221A0 +1A0 00000000 00000000      +1A8 00000000 00000000      +1B0 00000000 00000000      +1B8 00000000 00010000
009221C0 +1C0 00000000 00000000      +1C8 00000000 00000000      +1D0 00000000 00000000      +1D8 00000000 00000000
009221E0 +1E0 00000000 00000000      +1E8 00000000 00000000      +1F0 DECBDECB 00922000      +1F8 00000000   000000
00922200 +200  C J 0 0 00000000      +208 00000000 00000000      +210 FF00FF00 00000000      +218 00000000 00000000
00922220 +220 00000000 00000000      +228 00000000 00000000      +230 00000000 00000000      +238 0092C000 00010FFF
00922240 +240 FC160100 00000000      +248 00145831 00000000      +250 00000000 00000000      +258 00000000 00000000
00922260 +260 00000000 00000000      +268 00000000 00000000      +270 DECBDECB 00922000      +278 00000000   3E0000
00922280 +280  C J 0 0 00000000      +288 00000000 00000000      +290 FF00FF00 00000000      +298 00000000 00000000
009222A0 +2A0 00000000 00000000      +2A8 00000000 00000000      +2B0 00000000 00000000      +2B8 00000000 00010000
009222C0 +2C0 00000000 00000000      +2C8 00000000 00000000      +2D0 00000000 00000000      +2D8 00000000 00000000
009222E0 +2E0 00000000 00000000      +2E8 00000000 00000000      +2F0 DECBDECB 00922000      +2F8 00000000   000000
00922300 +300  C J 0 0 00000000      +308 00000000 00000000      +310 FF00FF00 00000000      +318 00000000 00000000
```

Figure 45. ECB Virtual Memory (EVM) (Part 3 of 5)

```
00922320 +320 00000000 00000000      +328 00000000 00000000      +330 00000000 00000000      +338 0092C000 00010FFF
00922340 +340 FC160100 00000000      +348 0014582D 00000000      +350 00000000 00000000      +358 00000000 00000000
00922360 +360 00000000 00000000      +368 00000000 00000000      +370 DECBDECB 00922000      +378 00000000   3E0000
00922380 +380  C J 0 0 00000000      +388 00000000 00000000      +390 FF00FF00 00000000      +398 00000000 00000000
009223A0 +3A0 00000000 00000000      +3A8 00000000 00000000      +3B0 00000000 00000000      +3B8 00000000 00010000
009223C0 +3C0 00000000 00000000      +3C8 00000000 00000000      +3D0 00000000 00000000      +3D8 00000000 00000000
009223E0 +3E0 00000000 00000000      +3E8 00000000 00000000      +3F0 DECBDECB 00922000      +3F8 00000000   000000
00922400 +400  C J 0 0 00000000      +408 00000000 00000000      +410 FF00FF00 00000000      +418 00000000 00000000
00922420 +420 00000000 00000000      +428 00000000 00000000      +430 00000000 00000000      +438 0092C000 00010FFF
00922440 +440 FC160100 00000000      +448 00145831 00000000      +450 00000000 00000000      +458 00000000 00000000
00922460 +460 00000000 00000000      +468 00000000 00000000      +470 DECBDECB 00922000      +478 00000000   3E0000
00922480 +480  C J 0 0 00000000      +488 00000000 00000000      +490 FF00FF00 00000000      +498 00000000 00000000
009224A0 +4A0 00000000 00000000      +4A8 00000000 00000000      +4B0 00000000 00000000      +4B8 00000000 00010000
009224C0 +4C0 00000000 00000000      +4C8 00000000 00000000      +4D0 00000000 00000000      +4D8 00000000 00000000
009224E0 +4E0 00000000 00000000      +4E8 00000000 00000000      +4F0 DECBDECB 00922000      +4F8 00000000   000000
00922500 +500  C J 0 0 00000000      +508 00000000 00000000      +510 FF00FF00 00000000      +518 00000000 00000000
00922520 +520 00000000 00000000      +528 00000000 00000000      +530 00000000 00000000      +538 0092C000 00010FFF
00922540 +540 FC160100 00000000      +548 0014582D 00000000      +550 00000000 00000000      +558 00000000 00000000
00922560 +560 00000000 00000000      +568 00000000 00000000      +570 DECBDECB 00922000      +578 00000000   3E0000
00922580 +580  C J 0 0 00000000      +588 00000000 00000000      +590 FF00FF00 00000000      +598 00000000 00000000
009225A0 +5A0 00000000 00000000      +5A8 00000000 00000000      +5B0 00000000 00000000      +5B8 00000000 00010000
009225C0 +5C0 00000000 00000000      +5C8 00000000 00000000      +5D0 00000000 00000000      +5D8 00000000 00000000
009225E0 +5E0 00000000 00000000      +5E8 00000000 00000000      +5F0 DECBDECB 00922000      +5F8 00000000   000000
00922600 +600  C J 0 0 00000000      +608 00000000 00000000      +610 FF00FF00 00000000      +618 00000000 00000000
00922620 +620 00000000 00000000      +628 00000000 00000000      +630 00000000 00000000      +638 0092C000 00010FFF
00922640 +640 FC160100 00000000      +648 00145831 00000000      +650 00000000 00000000      +658 00000000 00000000
00922660 +660 00000000 00000000      +668 00000000 00000000      +670 DECBDECB 00922000      +678 00000000   3E0000
00922680 +680  C J 0 0 00000000      +688 00000000 00000000      +690 FF00FF00 00000000      +698 00000000 00000000
009226A0 +6A0 00000000 00000000      +6A8 00000000 00000000      +6B0 00000000 00000000      +6B8 00000000 00010000
009226C0 +6C0 00000000 00000000      +6C8 00000000 00000000      +6D0 00000000 00000000      +6D8 00000000 00000000
009226E0 +6E0 00000000 00000000      +6E8 00000000 00000000      +6F0 DECBDECB 00922000      +6F8 00000000   000000
00922700 +700  C J 0 0 00000000      +708 00000000 00000000      +710 FF00FF00 00000000      +718 00000000 00000000
00922720 +720  Q D C B  . C R U      +728  S A                   +730 00000000 00000000      +738 00000000 00010000
00922740 +740 00000000 00000000      +748 00000000 00000000      +750 00000000 00000000      +758 00000000 00000000
00922760 +760 00000000 00000000      +768 00000000 00000000      +770 DECBDECB 00922000      +778 00000000 80000000
00922780 +780  Q D C B 00000000      +788 00000000 00000000      +790 00000000 00000000      +798 00000000 00000000
009227A0 +7A0 00000000 00000000      +7A8 00000000 00000000      +7B0 00000000 00000000      +7B8 00000000 00010000
009227C0 +7C0 00000000 00000000      +7C8 00000000 00000000      +7D0 00000000 00000000      +7D8 00000000 00000000
009227E0 +7E0 00000000 00000000      +7E8 00000000 00000000      +7F0 DECBDECB 00922000      +7F8 00000000   000000
00922800 +800 00000000 00000000      +808 00000000 00000000      +810 00000000 00000000      +818 00000000 00000000
00922820 +820 00000000 00000000      +828 00000000 00000000      +830 00000000 00000000      +838 00000000 00010000
00922840 +840 00000000 00000000      +848 00000000 00000000      +850 00000000 00000000      +858 00000000 00000000
00922860 +860 00000000 00000000      +868 00000000 00000000      +870 DECBDECB 00922000      +878 00000000   000000
00922880 +880 00000000 00000000      +888 00000000 00000000      +890 00000000 00000000      +898 00000000 00000000
009228A0 +8A0 00000000 00000000      +8A8 00000000 00000000      +8B0 00000000 00000000      +8B8 00000000 00010000
009228C0 +8C0 00000000 00000000      +8C8 00000000 00000000      +8D0 00000000 00000000      +8D8 00000000 00000000
009228E0 +8E0 00000000 00000000      +8E8 00000000 00000000      +8F0 DECBDECB 00922000      +8F8 00000000   000000
00922900 +900 00000000 00000000      +908 00000000 00000000      +910 00000000 00000000      +918 00000000 00000000
00922920 +920 00000000 00000000      +928 00000000 00000000      +930 00000000 00000000      +938 00000000 00010000
00922940 +940 00000000 00000000      +948 00000000 00000000      +950 00000000 00000000      +958 00000000 00000000
00922960 +960 00000000 00000000      +968 00000000 00000000      +970 DECBDECB 00922000      +978 00000000   000000
00922980 +980 00000000 00000000      +988 00000000 00000000      +990 00000000 00000000      +998 00000000 00000000
009229A0 +9A0 00000000 00000000      +9A8 00000000 00000000      +9B0 00000000 00000000      +9B8 00000000 00010000
009229C0 +9C0 00000000 00000000      +9C8 00000000 00000000      +9D0 00000000 00000000      +9D8 00000000 00000000
009229E0 +9E0 00000000 00000000      +9E8 00000000 00000000      +9F0 DECBDECB 00922000      +9F8 00000000   000000
00922A00 +A00 00000000 00000000      +A08 00000000 00000000      +A10 00000000 00000000      +A18 00000000 00000000
00922A20 +A20 00000000 00000000      +A28 00000000 00000000      +A30 00000000 00000000      +A38 00000000 00010000
00922A40 +A40 00000000 00000000      +A48 00000000 00000000      +A50 00000000 00000000      +A58 00000000 00000000
00922A60 +A60 00000000 00000000      +A68 00000000 00000000      +A70 DECBDECB 00922000      +A78 00000000   000000
00922A80 +A80 00000000 00000000      +A88 00000000 00000000      +A90 00000000 00000000      +A98 00000000 00000000
00922AA0 +AA0 00000000 00000000      +AA8 00000000 00000000      +AB0 00000000 00000000      +AB8 00000000 00010000
00922AC0 +AC0 00000000 00000000      +AC8 00000000 00000000      +AD0 00000000 00000000      +AD8 00000000 00000000
00922AE0 +AE0 00000000 00000000      +AE8 00000000 00000000      +AF0 DECBDECB 00922000      +AF8 00000000   000000
00922B00 +B00 00000000 00000000      +B08 00000000 00000000      +B10 00000000 00000000      +B18 00000000 00000000
00922B20 +B20 00000000 00000000      +B28 00000000 00000000      +B30 00000000 00000000      +B38 00000000 00010000
00922B40 +B40 00000000 00000000      +B48 00000000 00000000      +B50 00000000 00000000      +B58 00000000 00000000
00922B60 +B60 00000000 00000000      +B68 00000000 00000000      +B70 DECBDECB 00922000      +B78 00000000   000000
```

Figure 45. ECB Virtual Memory (EVM) (Part 4 of 5)

```
00922B80 +B80 00000000 00000000      +B88 00000000 00000000      +B90 00000000 00000000      +B98 00000000 00000000
00922BA0 +BA0 00000000 00000000      +BA8 00000000 00000000      +BB0 00000000 00000000      +BB8 00000000 00010000
00922BC0 +BC0 00000000 00000000      +BC8 00000000 00000000      +BD0 00000000 00000000      +BD8 00000000 00000000
00922BE0 +BE0 00000000 00000000      +BE8 00000000 00000000      +BF0 DECBDECB 00922000      +BF8 00000000   000000
00922C00 +C00 00000000 00000000      +C08 00000000 00000000      +C10 00000000 00000000      +C18 00000000 00000000
00922C20 +C20 00000000 00000000      +C28 00000000 00000000      +C30 00000000 00000000      +C38 00000000 00010000
00922C40 +C40 00000000 00000000      +C48 00000000 00000000      +C50 00000000 00000000      +C58 00000000 00000000
00922C60 +C60 00000000 00000000      +C68 00000000 00000000      +C70 DECBDECB 00922000      +C78 00000000   000000
00922C80 +C80 00000000 00000000      +C88 00000000 00000000      +C90 00000000 00000000      +C98 00000000 00000000
00922CA0 +CA0 00000000 00000000      +CA8 00000000 00000000      +CB0 00000000 00000000      +CB8 00000000 00010000
00922CC0 +CC0 00000000 00000000      +CC8 00000000 00000000      +CD0 00000000 00000000      +CD8 00000000 00000000
00922CE0 +CE0 00000000 00000000      +CE8 00000000 00000000      +CF0 DECBDECB 00922000      +CF8 00000000   000000
00922D00 +D00 00000000 00000000      +D08 00000000 00000000      +D10 00000000 00000000      +D18 00000000 00000000
00922D20 +D20 00000000 00000000      +D28 00000000 00000000      +D30 00000000 00000000      +D38 00000000 00010000
00922D40 +D40 00000000 00000000      +D48 00000000 00000000      +D50 00000000 00000000      +D58 00000000 00000000
00922D60 +D60 00000000 00000000      +D68 00000000 00000000      +D70 DECBDECB 00922000      +D78 00000000   000000
00922D80 +D80 00000000 00000000      +D88 00000000 00000000      +D90 00000000 00000000      +D98 00000000 00000000
00922DA0 +DA0 00000000 00000000      +DA8 00000000 00000000      +DB0 00000000 00000000      +DB8 00000000 00010000
00922DC0 +DC0 00000000 00000000      +DC8 00000000 00000000      +DD0 00000000 00000000      +DD8 00000000 00000000
00922DE0 +DE0 00000000 00000000      +DE8 00000000 00000000      +DF0 DECBDECB 00922000      +DF8 00000000   000000
00922E00 +E00 00000000 00000000      +E08 00000000 00000000      +E10 00000000 00000000      +E18 00000000 00000000
00922E20 +E20 00000000 00000000      +E28 00000000 00000000      +E30 00000000 00000000      +E38 00000000 00010000
00922E40 +E40 00000000 00000000      +E48 00000000 00000000      +E50 00000000 00000000      +E58 00000000 00000000
00922E60 +E60 00000000 00000000      +E68 00000000 00000000      +E70 DECBDECB 00922000      +E78 00000000   000000
00922E80 +E80 00000000 00000000      +E88 00000000 00000000      +E90 00000000 00000000      +E98 00000000 00000000
00922EA0 +EA0 00000000 00000000      +EA8 00000000 00000000      +EB0 00000000 00000000      +EB8 00000000 00010000
00922EC0 +EC0 00000000 00000000      +EC8 00000000 00000000      +ED0 00000000 00000000      +ED8 00000000 00000000
00922EE0 +EE0 00000000 00000000      +EE8 00000000 00000000      +EF0 DECBDECB 00922000      +EF8 00000000   000000
00922F00 +F00 00000000 00000000      +F08 00000000 00000000      +F10 00000000 00000000      +F18 00000000 00000000
00922F20 +F20 00000000 00000000      +F28 00000000 00000000      +F30 00000000 00000000      +F38 00000000 00010000
00922F40 +F40 00000000 00000000      +F48 00000000 00000000      +F50 00000000 00000000      +F58 00000000 00000000
00922F60 +F60 00000000 00000000      +F68 00000000 00000000      +F70 DECBDECB 00922000      +F78 00000000   000000
00922F80 +F80 00000000 00000000      +F88 00000000 00000000      +F90 00000000 00000000      +F98 00000000 00000000
00922FE0 +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 00000063

*MSG BLK, LEVEL 0
00919E80 +000 00000040 002118F2      +008 00000000 00000000      +010 00130100 00 Z D E      +018  C B   A  P I - C
00919EA0 +020  R U S A 00005810      +028 58101000 58E01010      +030 0058E0E0 0058BA &      +038 145800E0 00 &10D0
00919EC0 +040 98411000 0541E070      +048  * &10D0 A0 &00 J      +050 2C98 0B0 0841B070      +058 38 &B0D0; A441B002
00919EE0 +060 8A &B0D0;
AC58B0 J      +068 2C &00 A  4 &90D0      +070 9C4110D0 98 &E0D0;      +078 A8 &B0D0; B005EF58
00919F00 +080 EA &1841 B0709341      +088 10000298  0E008 &      +090 B0D09841 E0000141      +098 B00003 & 10D09C &
00919F20 +0A0 10D0A4 & 00 A 4 &      +0A8 90D0A041 10D098 &      +0B0 E0D0A8 & 20D0AC &      +0B8 90D0B0 & B0D0B405
00919F40 +0C0 EF581A & 10582A &      +0C8 1C581010 0098 020      +0D0 08412070  * &10D0      +0D8 98411002 8A &20D0
00919F60 +0E0 9C412070 9C &10D0      +0E8 A0411000 1F &20D0      +0F0 A4412008 20 &10D0      +0F8 A858103F E0 &20D0
00919F80 +100 AC41200C 9A &10D0      +108 B0 &00 A  44110D0      +110 98 &20D0 B405EF58      +118 1A &3058 2A &24 K
00919FA0 +120 1FD09810 0098 020      +128 08 &00 A  44110D0      +130 9805EF58 B0 /1058      +138 10 /0C58 00B014 &
00919FC0 +140 00 /4847  0365807      +148 0058203F  44182 &      +150 00582080 341B0043      +158 00 /  43 20200054
00919FE0 +160 203F U19 025820       +168 00477036 80 K0720      +170 00100047  036 D07      +178 00430010 07270013
*STSA BLOCK
03052400      00000000 00000000           00000000 00000000           00000000 00000000           00000000 03052800
03052420      03053800 00000000           071D1000 00000000           00000000 00000000           00000000 00000000
03052440      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
03052600      D7D9D6C3 00000000           00000000 00000000           00000000 00008F3E           00000000 00000000
03052620      00000000 00000000           00000000 00000000           026FC400 00000100           00000101 00000000
03052640      00000006 00000002           00000000 0000FF00           00000000 00000000           00000000 00000000
03052660      00000000 00902000           023FA000 00000000           00000000 00000000           00000100 00000100
03052680      00000101 00000101           00000000 00000000           00000000 00000000           00000000 00000000
030526A0      00000000 00000000           00000000 4005012E           4005012E 00000000           00000000 00000000
030526C0      00000000 00040000           00000000 00000000           00000000 00000000           00000000 00000000
030526E0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
030527E0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
```

Figure 45. ECB Virtual Memory (EVM) (Part 5 of 5)

Table 34 explains the headings and labels that appear in this section of the main storage dump.

Table 34. Headings in the ECB Virtual Memory (EVM)

Heading	Description
*ENTRY REFERENCED VIA R9 – DUMP OF ECB VIRTUAL MEMORY FOLL OWS	This heading indicates the beginning of the dump of the failing ECB's virtual memory (EVM). In CTL dumps with no associated ECB, this section is omitted.

Table 34. Headings in the ECB Virtual Memory (EVM) (continued)

Heading	Description
*ENTRY BLK, GENERATED BY 3270 LOCAL ENTRY	This heading identifies the particular type of ECB, based on the value of its format flag. The ECB is labeled throughout with dump labels used to identify specific fields.
*ECB MACRO TRACE	The ECB macro trace table is a wraparound trace table, with room for 55 entries when register trace is off, and 23 entries when register trace is active. The register trace allows you to display registers when the macro trace is active. The enter macros cause the following values to be placed in the registers indicated: • R8 is equal to 0 for an ENTNC from CP • R10 is equal to the base of CCEB (not R10) at the time ENTxC is issued • R13 is equal to the stack register of CCEB (not R13) at the time the ENTxC is issued The header of the trace table is identified by the MTH dump label, the first fullword of which points to the next available trace entry. This pointer is not reset to the beginning of the table at ECB creation time, so as to preserve a more complete picture of system activity in the collated trace table. Each ECB trace entry has an I-stream affinity associated with it. It may be seen that this ECB switched I-streams twice before the error occurred. The remaining fields are identical to those in the collated ECB trace table.
*DECB FRAME	This heading indicates the ECB virtual address of the first data event control block (DECB) frame for this ECB.

The data and program blocks that are logically attached to the ECB are described in Table 35. The headings for the program blocks contain the name and version of the program, the name of the loadset in which the program was contained, and an indication of the allocation type. If the program is not contained in an E-type loader loadset, it is considered to be a base program and the word BASE is displayed. If the TPF system was not generated with support for tracking program versions, the program version is not displayed.

One of the following headings may also be displayed if a problem occurs while trying to dump a working storage block.

Table 35. Headings in the ECB Virtual Memory (EVM)

Heading	Description
*INVALID BLOCK TYPE	This heading is displayed when it is determined that the value of the block type on an ECB data level or DECB is not valid. An attempt is made to dump the data in the block using the address of the block as a starting point and continuing to the next 4-KB boundary. No data translation is attempted.
*INVALID BLOCK ADDRESS	This heading is displayed when the block address on an ECB data level or DECB does not point to a valid TPF block. An attempt is made to dump the data in the block, using the address of the block as a starting point and continuing to the next 4-KB boundary. No data translation is attempted.
*INVALID FORMAT FLAG	This heading appears when the value of the format flag contained in the storage block is not recognized by the postprocessor. Most often this is due to the fact that the storage reserved for the format flag in a particular storage block may occasionally be used by the TPF control program (CP) to store actual data. No data translation is attempted when this condition is encountered.

Additional Blocks Mapped

Figure 46 shows an example of additional blocks mapped.

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

ADDITIONAL BLOCKS MAPPED IN THIS EVM


*AVAIL L1 BLK
00319E80 +000 00000000 0051185A      +008 00000000 00000000      +010 00090100 00 Z T E      +018  S T +00 00000000
00319EA0 +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038 00000000 00000000
00319FE0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00130057

*AVAIL L1 BLK
00319A80 +000 0E . . .  . . . -      +008  B   Z T  E S T00      +010 00000000 00000000      +018 00000000 00000000
00319AA0 +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038 00000000 00000000
00319BE0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00130000

*AVAIL L1 BLK
00319900 +000 00000000 00211500      +008 00000000 00000000      +010 00670100 00 1 B11      +018 0000 C S  M P 0 0
00319920 +020  9 7 I    1 9 . 5      +028  3 . 0 1    C P U      +030  - B   S  S - B S      +038  S     S  S U - H
00319940 +040  P N      I S - 0      +048  1110000  C S M P      +050  0 0 9 9  I   1 9      +058  . 5 3 .  0 1   0
00319960 +060  1 0 0 0  0 - B       +068  Z T M N  T   R T      +070  A   4 2  0   B P      +078  +000000 00000000
SYSTEM ERROR NUMBER   CTL-I000004   VIRTUAL STORAGE ERROR IN OP                                      DUMP 1623  PAGE    60
00319980 +080 00000000 00000000      +088 00000000 00000000      +090 00000000 00000000      +098 00000000 00000000
00319A60 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 000031 X      +178 B00032 # E0130000

*AVAIL L1 BLK
00319780 +000 00000000 0021A5 Y      +008 00000000 FFFFFFFF      +010 00840000 0001 H11      +018      C S  M P 0 0
003197A0 +020  9 7 I    1 9 . 5      +028  3 . 0 3    C P U      +030  - B   S  S - B S      +038  S     S  S U - H
003197C0 +040  P N      I S - 0      +048  315 G O  G O 0 0      +050  0 1 I    1 9 . 5      +058  3 . 0 3    A P P
003197E0 +060  L I C A  T I O N      +068    F I X  E D   C      +070  O R E    L O A D      +078  E D15 M  O D E
00319800 +080  715 S S  U   H P      +088  N   - -    I - S      +090    0 315 19FF0000      +098 00000000 00000000
00319820 +0A0 00000000 00000000      +0A8 00000000 00000000      +0B0 00000000 00000000      +0B8 00000000 00000000
003198E0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00130000

*AVAIL L1 BLK
00319600 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00319760 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00000000

*AVAIL L1 BLK
00319480 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
003195E0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00000000

*AVAIL L1 BLK
00319300 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00319460 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00000000

*AVAIL L1 BLK
00319180 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
003192E0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00000000

*AVAIL L1 BLK
00319000 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00319160 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00000000

*AVAIL L2 BLK
00327420 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00327480 +060 00000000 00000000      +068 00000000 00010000      +070 00000000 00000000      +078 00000000 00000000
003274A0 +080 00000000 00000000      +088 00000000 00000000      +090 00000000 00000000      +098 00000000 00000000
00327780 +360  O M0000 0021A500      +368 00000000 FFFFFFFF      +370 00840000 00 1 H11      +378      C S  M P 0 0
003277A0 +380  9 7 I    1 9 . 5      +388  3 . 0 3    C P U      +390  - B   S  S - B S      +398  S     S  S U - H
003277C0 +3A0  P N      I S - 0      +3A8  315 G O  G O 0 0      +3B0  0 1 I    1 9 . 5      +3B8  3 . 0 3    A P P
003277E0 +3C0  L I C A  T I O N      +3C8    F I X  E D   C      +3D0  O R E    L O A D      +3D8  E D15 M  O D E
00327800 +3E0  715 S S  U   H P      +3E8  N   - -    I - S      +3F0    0 315 19FF0000      +3F8 00000000 00000000
00327820 +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 00000013
```

Figure 46. Additional Blocks Mapped (Part 1 of 2)

```
*AVAIL L2 BLK
00327BE0 +000 010000 B  S M P B      +008   151000 00000000      +010 18000000 00000000      +018 0000 S M  P B0100
00327C00 +020 0015C000 00000004      +028 00319E80 00000000      +030 0010 000 01159458      +038 00000000 0030 -08
00327C20 +040 00000000 00000000      +048 000005 D 00000000      +050 00319C1A 00327420      +058 00000000 00319E80
00327C40 +060 00000000 00000000      +068 00000000 0000 +00      +070 00000005 00000000      +078 00319E9B 00000000
00327C60 +080 00000000 00000000      +088 00000000 00110000      +090 00000000 00000000      +098 001D0000 00000000
00327C80 +0A0 00000000 00000000      +0A8 00000000 00000000      +0B0 00000000 00000000      +0B8 00000000 00000000
00327CA0 +0C0 00000000 00000000      +0C8 0000 +00 00000000      +0D0 00000000 00000000      +0D8 00000000 00000000
00327CC0 +0E0 00000000 00000000      +0E8 00000000 00000000      +0F0 00000000 00000000      +0F8 00000000 00000000
00327FE0 +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 00000013

*AVAIL L2 BLK
00327000 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00327400 +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 00000000

*AVAIL L4 BLK
00328000 +000 00FF007F  C V Z Z      +008  N0192DC 806C4770      +010 802E0A2C 08510A32      +018 00084110 00020AB6
00328020 +020  (E0B23C 00017FB4      +028   010000 0000 N01      +030 92DC806E 47708054      +038 0A2C1051 0A320010
00328040 +040 41100003 0AB6 (E0      +048 B23C0001 7FB4  01      +050 00000000 0A2C1851      +058 0A320018 58108068
00328060 +060 95001000 00000000      +068 0030 000 00010002      +070  0 8 / 2  1 / 9 2      +078  1 1 . 3  4 2 400
00328080 +080 00000000 00000000      +088 00000000 00000000      +090 00000000 00000000      +098 00000000 00000000
00328FE0 +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 00000013

*DATA BLK
00380E80 +000 00000000 002115 Y      +008 00000000 00000000      +010 00150100 0001 C11      +018    A Z T  E S T11
00380EA0 +020 5A J1312  * 0 +00      +028 00000000 00000000      +030 00000000 00000000      +038 00000000 00000000
00380EC0 +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
00380FE0 +160 00000000 00000000      +168 00000000 00000000      +170 00000000 00000000      +178 00000000 00130013

*DATA BLK
*BLOCK IN USE
00388000 +000 00000000 00000000      +008 00000000 00000000      +010 00000000 00000000      +018 00000000 00000000
00388BE0 +BE0  S M1A00  A C P F      +BE8 00000000 00000000      +BF0 00000000 00000000      +BF8 00000000 00000000
00388C00 +C00 00000000 00000000      +C08 00000000 00000000      +C10 00000000 00000000      +C18 00000000 00000000
00388FE0 +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 00000013
```

Figure 46. Additional Blocks Mapped (Part 2 of 2)

Table 36 explains the headings and labels that appear in this section of the main storage dump.

Table 36. Headings in the Additional Blocks Mapped

Heading	Description
*ADDITIONAL BLOCKS MAPPED IN THIS EVM	Once all data objects that are logically attached to the ECB have been dumped, the ECB virtual memory (EVM) is scanned for additional storage mapped into the ECB virtual address space. In this example the ECB avail queues were found to contain a number of blocks, some of which have already been acquired and released by the application program. Each block is dumped in case it contains residual data that could aid in debugging. In addition, 2 data blocks were found for which no logical reference appears anywhere in the ECB. The block at address X'00388000' was acquired by the control program (CP) and has not yet been released. Note that both data blocks have been allocated from the same 4 KB frame.

Table 37 explains the headings and labels that appear in this section of the main storage dump. These headings may appear in the ECB virtual memory (EVM) dump if the TPF system is running in VEQR mode.

Table 37. Headings in the Additional Blocks Mapped

Heading	Description
*BLOCK AUTHORIZED TO EVM - OWNED BY ECB AT *xxxxxxxx*	In VEQR mode it is possible that the scan of the ECB virtual memory (EVM) detects other ECBS, or 4 KB frames belonging to other ECBs. This heading identifies such blocks, and provides the address of the owning ECB if applicable.

Table 37. Headings in the Additional Blocks Mapped (continued)

Heading	Description
*COMMON BLOCK OWNED BY THIS ECB	If block check mode is active, system error processing performs a scan of all 4 KB common blocks. If a common block is found to be owned by the failing ECB it is dumped, preceded by this heading.

System Storage

Figure 47 shows an example of system storage.

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

DUMP OF SYSTEM STORAGE FOLLOWS


*CONTROL PROGRAM
00001000      07009007 92A858D0          032458F0 030C4AF0          E00058A0 F0009140          BE774770 B0200B0A
00001020      50ED0054 50FD0050          58F0BE70 58F0F048          0DEF58ED 00540B0A          58109410 48101032
00001040      482F0008 17121412          4770B052 58A0F004          07FAAF02 000050C9          000000EB B0640000
00001060      00000000 23E4D5C1          E4E3C8D6 D9C9E9C5          C440E4E2 C540D6C6          40C6C1E2 E340D3C9
00001080      D5D240D4 C1C3D9D6          00000000 00000000          00000000 00000000          07000000 00000000
000010A0      00000700 00000000          00000000 47F0C2A0          00255000 00255420          000F8000 0C070002
000010C0      000FA000 01070003          000FD000 0C070004          00000000 00000000          01347000 0135F000
000010E0      00101000 0C070004          00105000 01070004          00109000 0C070004          00000000 00000000
00001100      0010D000 00000001          0135F000 0C070008          01367000 0107000C          01373000 0C070008
00001120      00000000 00000000          47F0B9BA 0000858C CNF      0010F000 00000000          00000000 00000000
00001140      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00001220      00000000 00000000          00000000 0000AACE          58A0B270 0B0A58A0          B26C0000 58A0B26C
00001240      58F0B274 0B0F0000          58A0B26C 58F0B278          0B0F0000 58A0B26C          58F0B27C 0B0F0000
00001260      58A0B26C 58F0B280          0B0F0000 00002B90          80002B90 80002D58          80002EC8 800032D8
00001280      8000324C 0009F300          58F0B2A8 07FF0000          58F0B2AC 07FF0000          58F0B2B0 07FF0000
000012A0      58F0B2B4 07FF0000          000A3BD0 000A1958          000A4366 000A3970          58F0B2C0 07FF0000
000012C0      0000CF90 00000C40          00000000 00000000          00000000 00000000          00000000 00000000
000012E0      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00001300      00000000 00000000          0010E500 00000000 WLT   NEB 00000002 00000000      NE2 00000002 00000000
00001320      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00001500      00000000 00000000          00000000 00000000          00000000 00000000      NE3 00000000 00000000
00001520      00000002 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00001540      00000000 00000000          00000000 00000000          00000000 00000000          0059E060 0059E0E0
00001560      0059E160 00000000          00000000 00000000          00000000 00000000          58F0079C 07FF0000
00001580      47F0BF9E 47F0BFA4          47F0C202 47F0C060          47F0C2FA 47F0C306          47F0C874 47F0C8A0
000015A0      47000000 47F0C068          47F0BFD2 00000000          00150748 00000000          000C9000 013DB000
000015C0      013C2000 0010F000          00153D38 00153D30          00000000 0012C5A0          00000000 00000000
000015E0      0010E000 0010E250          00000000 0010FAE0          0012F300 000F7E44          00009870 000F56A8
   .             .        .                .        .                 .        .                 .        .
   .             .        .                .        .                 .        .                 .        .
   .             .        .                .        .                 .        .                 .        .
```

Figure 47. System Storage

Table 38 explains the headings and labels that appear in this section of the main storage dump.

Table 38. Headings in the System Storage

Heading	Description
DUMP OF SYSTEM STORAGE FOLLOWS	A dump of system storage is produced for most CTL errors and for OPR errors when ZASER LONG is set or when SYSDUMP=YES is coded on the SERRC macro. Dump labels identify pertinent locations in storage. The dump begins at location X'1000' in the system virtual memory (SVM) of the failing I-stream, and proceeds to the end of the system virtual address space. Large system tables and areas of storage for which a dump keywords have been defined are omitted, in order to limit dumps to a reasonable size.

Table 38. Headings in the System Storage (continued)

Heading	Description
*CONTROL PROGRAM	For CTL-I000004 errors the large storage areas listed opposite the *DUMP KEYWORDS heading on the cover page is included. The control program (CP) is the first such area. Each time an area normally omitted from all dumps is included in the dump for a particular system error, it is identified by a header that is specified when the dump keyword is initially defined. Areas of storage which do not have a keyword associated with them are preceded by the *SYSTEM STORAGE heading.

Dumps of ECB Working Storage in the System Virtual Memory (SVM)

Occasionally the control program (CP) may need to invoke the SERRC macro for an error for which there is an associated ECB, but at the time of the error the CP is running in the system virtual memory (SVM). When this happens, system error processing presents the same view of the ECB and associated blocks in the dump that the CP has at the time the SERRC macro was issued. This means that the addresses on the left hand side of the dump are system virtual addresses; however, the addresses imbedded in the ECB core block reference words (and elsewhere) are still ECB virtual addresses. An example is shown in Figure 48 on page 104.

To aid you in locating data blocks referenced by the ECB virtual memory (EVM) addresses, a special table is produced following the dump of the ECB, which is known as the EVA-to-SVA map table. Each 4-byte entry in this table represents a 4 KB page of the ECB private area, and contains the system virtual address of the corresponding page. Replacing the segment- and page-index portions of an ECB virtual address with the segment- and page-index portions of the address in an EVA-to-SVA map table entry yields the system virtual address of the block in question. This address can then be used to locate the data block in the dump.

ECB Working Storage Dumps in the System Virtual Memory (SVM)

Figure 48 on page 104 shows an example of ECB working storage dumps.

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

ENTRY REFERENCED VIA R9 - DUMP OF ECB VIRTUAL MEMORY FOLLOWS


*ENTRY BLK, GENERATED BY 1052/3215 ENTRY
02274000  CHW 00000000 8013070E BAD
02274008 W000 00000000 00000140      W008 01C50000 00000000      W016 01000000 00000000      W024 00000000 D4F80501
   BCD                                     E                                                                 M 8
02274028 W032 00000000 00001387      W040 00000000 00000000      W048 00000000 E2D4D7C2      W056 010000C2 80B00000
   BCD                                                                    S M P B                    B
02274048 W064 00000000 00000000      W072 00008400 04000000      W080 E3C5E2E3 00000000      W088 00000000 05961458
   BCD                                                                  T E S T
02274068 W096 00000000 00000000      WSW1 00000000 00000000
02274078  FA0 00000000 00000000       FA1 00000000 D4F80005       FA2 00000000 00000000       FA3 00000000 00000000
02274098  FA4 00000000 00000000       FA5 00000000 00000000       FA6 00000000 00000000       FA7 00000000 00000000
022740B8  FA8 00000000 00000000       FA9 00000000 00000000       FAA 00000000 00000000       FAB 00000000 00000000
022740D8  FAC 00000000 00000000       FAD 00000000 00000000       FAE 00000000 00000000       FAF 00000000 00000000
022740F8  FAP 00000000 00000000
02274100  CR0 00931BE0 0031041F       CR1 00925000 00010FFF       CR2 00925000 00510FFF       CR3 007C6000 00510FFF
02274120  CR4 00000000 00010000       CR5 00000000 00010000       CR6 00000000 00010000       CR7 00000000 00010000
02274140  CR8 00918C00 0001017D       CR9 00000000 00010000       CRA 00000000 00010000       CRB 00000000 00010000
02274160  CRC 00000000 00010000       CRD 00000000 00010000       CRE 00000000 00010000       CRF 00000000 00010000
02274180  CRP 0065A910 00010000
02274188  FX0 00000000 00000000       FX1 00000000 00000000       FX2 00000000 00000000       FX3 00000000 00000000
022741A8  FX4 00000000 00000000       FX5 00000000 00000000       FX6 00000000 00000000       FX7 00000000 00000000
022741C8  FX8 00000000 00000000       FX9 00000000 00000000       FXA 00000000 00000000       FXB 00000000 00000000
022741E8  FXC 00000000 00000000       FXD 00000000 00000000       FXE 00000000 00000000       FXF 00000000 00000000
02274208  SUD 00000000 00000000           00000000 00000000       SUP 0000FF00 FF000000           00000000 00213E60
02274228      B441CC69 49C41606           00000000 00000000           00000001 912AA843           00000003 00000001
02274248      00000001 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02274268      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02274288      00906E3C 00000000           00000000 00000000       GLA 001CA000 001DA000 GLY
022742A0  R14 804ADC10 804AAE86 R15
022742A8  R0  00000004 00A01DB0 R1    R2  00A01DD0 00000004 R3    R4  0000001F 00000000 R5    R6  00A01F88 00000000 R7
022742C8  R8  0065A910
022742CC  CPL 009042D0 00906A68 FPL
022742D4  CXR 00000008 C3E5C1F1       ISN 00010000 00008400       STP 00000000 000062F0 CNF   CSC 00000000 00000000 CSP
022742F4  EHT 00000000 000FDEE8 TRT   TCA 00000000 00000000 STK   VCT 00320000 00906E3C DET       01000000 00000000
02274314      00000000 00000000           00FF0000 00000000       CRS 00000000 000000C2 CPX
0227432C  SBI FFFFFFFF FF00FF00 DBI   SSU FF000000
02274338  PSW 071D0000 8065B048       CTL 00000000 39000000       REC 00010000 00000000           00000040 0001925A TRC
02274358      0080C67B 6CBF0000       TTA 00000000 00000000 AUT
02274368  URA 00000000 00000000 URB   UR0 00000000 00000000 UR1   UR2 00000000 00000000 UR3   UR4 00000000 00000000 UR5
02274388  UR6 00000000 00000000 UR7
02274390 X000 01008400 8000E3C5      X008 E2E3C3E5 E9E90004      X016 00000100 00000000      X024 00918E80 00918EA1
   BCD                    T E            S T C V  Z Z
022743B0 X032 00918E80 05961458      X040 00000880 0000000B      X048 001C3FF8 00000000      X056 00000000 00000000
   BCD                                                                  8
022743D0 X064 001C3FF8 05961458      X072 00000000 C3E5C1F3      X080 00000000 00000000      X088 804AACE0 00000007
   BCD              8                             C V A 3
022743F0 X096 00000000 00000000      XSW0 02000000 00000000           00000000 00000000           00000000 000000B5
02274410      FFFFFFFF C8200000           00000000 8065B380           01C9D8E9 E9F0C100           A01DB000 00000000
   BCD                H                                               I Q Z  Z 0 A
02274430      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02274530      00000000 00000000           00000000 00000000
02274540 USA0 00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 1 of 6)

```
02275000  AV1 029C96A0 02A3F730 AV2   AV4 00000000 02DC800F STD   SGT 02DC8000 02DCC800 PGT   NVA 000000C4 02274000 SVA
02275020  EVA 00904000 02A3F700 FRM   EHT 02A68FD4 02DC8004 GLB   GLX 02DC8188 000D0000 FG2   GMT 02DC9800 02DC9804 GMH
02275040  TFC 000100A0 10000A00           00200001 00A01000 HFC   HTR 00A01000 00000000 ASC   EVC 00000000 00000000 RET
02275060  TOD B441CC67 B6CBF0C7           00000001 00000000           04586028 00000000       GSP 02DC8800 02DC881C GSH
02275080  ISO 01600010 04000007           00010007 02DCC800           00010000 00000000           00000000 00000000
022750A0      00000000 00000000           00000000 00000000           00000000 02DC8600 PCB   TCB 00000000 00000000 FCH
022750C0  BCH 00000000 02DC8400 TSA   STA 01600000 02DC9800 EGS   TYP 00000000 00000000       TMC 00000000 00000000
022750E0      00000000 00000000       DEC 0092C000 00000000           00000000 00000000           00000000 00000000
02275100      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02275120      00000000 00000000       AVM 00000000 00A01FC0           00000000 00000000 PER       00000000 00000000
02275140      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02275B00  USR 00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02275C00  SVT 00000000 00000000           00000000 00000000           82274000 82275000           82276000 00000000
02275C20      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02275C60      029C9600 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02275C80      00000000 00000000           00000000 00000000           00000000 029CE300           00000000 00000000
02275CA0      00000000 00000000           00000000 00000000           0299A200 00000000           00000000 00000000
02275CC0      00000000 02A3F700           00000000 00000000           00000000 00000000           00000000 00000000
02275CE0      00000000 00000000           00000000 00000000           00000000 00000000           00000000 00000000
02276000  MTH 009066A8 00000030           00906018 00906A38           02276018 02276A38

*ECB MACRO TRACE  *MACRO *PROG *LOCATION   *PARAMETERS                          *SVC OLD PSW    *CODE *INX   *TIME STAMP
02276018      IS01 ESFAC  QZZ0  +43A                                          071D0000 8065AD4C 003B 0014 B441CC67 B88FAB43
02276048      IS01 ESFAC  QZZ0  +442                                          071D0000 8065AD54 003B 0014 B441CC67 B892D443
02276078      IS01 ESFAC  QZZ0  +44C                                          071D0000 8065AD5E 003B 0014 B441CC67 B895BC03
022760A8      IS01 GETCC  QZZ0  +456     AD-00925000 TYPE-L4   LVL-D2         071D0000 8065AD68 002C      B441CC67 B898A643
022760D8      IS01 GETCC  QZZ0  +45A     AD-007C6000 TYPE-L4   LVL-D3     COMM 071D0000 8065AD6C 002C      B441CC67 B898D603
02276108      IS01 CALOC  QZZ0  +46C     SIZE-00000190 BLOCK-00A01DB0         071D0000 8065AD7E 003B 0006 B441CC67 B899A503
02276138      IS01 DECBC  QZZ0  +484     TYPE-CREATE  D-0092C020             071D2000 8065AD96 003B 0035 B441CC67 B89B6903
02276168      IS01 DECBC  QZZ0  +484     TYPE-CREATE  D-0092C0A0             071D0000 8065AD96 003B 0035 B441CC67 B89E6703
02276198      IS01 DECBC  QZZ0  +484     TYPE-CREATE  D-0092C120             071D0000 8065AD96 003B 0035 B441CC67 B89FE603
022761C8      IS01 DECBC  QZZ0  +484     TYPE-CREATE  D-0092C1A0             071D0000 8065AD96 003B 0035 B441CC67 B8A159C3
022761F8      IS01 DECBC  QZZ0  +484     TYPE-CREATE  D-0092C220             071D0000 8065AD96 003B 0035 B441CC67 B8A2C783
02276228      IS01 WTOPC  QZZ0  +512                                          071D0000 8065AE24 00F0      B441CC67 B8A443C3
02276258      IS01 DECBC  QZZ0  +526     TYPE-RELEASE D-0092C1A0             071D0000 8065AE38 003B 0035 B441CC67 B8A89C43
02276288      IS01 DECBC  QZZ0  +532     TYPE-CREATE  D-0092C2A0             071D0000 8065AE44 003B 0035 B441CC67 B8AA4BC3
022762B8      IS01 DECBC  QZZ0  +544     TYPE-RELEASE D-0092C2A0             071D0000 8065AE56 003B 0035 B441CC67 B8ABD003
022762E8      IS01 DECBC  QZZ0  +554     TYPE-RELEASE D-0092C020             071D0000 8065AE66 003B 0035 B441CC67 B8AD4B83
02276318      IS01 DECBC  QZZ0  +56A     TYPE-RELEASE D-0092C220             071D3000 8065AE7C 003B 0035 B441CC67 B8AED183
02276348      IS01 DECBC  QZZ0  +57A     TYPE-CREATE  D-0092C320             071D3000 8065AE8C 003B 0035 B441CC67 B8B05A03
02276378      IS01 DECBC  QZZ0  +586     TYPE-SWAPBLK D-0092C320 LVL-D0      071D3000 8065AE98 003B 0035 B441CC67 B8B1DE03
022763A8      IS01 DECBC  QZZ0  +596     TYPE-SWAPBLK D-0092C320 LVL-D0      071D3000 8065AEA8 003B 0035 B441CC67 B8B35583
022763D8      IS01 DECBC  QZZ0  +5A6     TYPE-LOCATE  D-0092CFA0             071D3000 8065AEB8 003B 0035 B441CC67 B8B4C1C3
02276408      IS01 WTOPC  QZZ0  +69A                                          071D0000 8065AFAC 00F0      B441CC68 BF80F6C4
02276438      IS01 DECBC  QZZ0  +5BC     TYPE-LOCATE  D-0092C320             071D0000 8065AECE 003B 0035 B441CC68 BF927A84
02276468      IS01 GETFC  QZZ0  +6A4                                          071D0000 8065AFB6 0034      B441CC69 4900C706
02276498      IS01 WTOPC  QZZ0  +712                                          071D0000 8065B024 00F0      B441CC69 49179B86
022764C8      IS01 FILEC  QZZ0  +72E     FA-0000000000046C4D ID-D6D4 D-0092C320 071D0000 8065B040 0016      B441CC69 491E0186
022764F8      IS01 FIWHC  QZZ0  +734     FA-0000000000046C4D ID-D6D4 D-0092C320 071D0000 8065B046 0028      B441CC69 493C8F46
02276528      IS01 UNFRC  QZZ0  +73E     FA-0000000000046C4D ID-D6D4 D-0092C320 071D0000 8065B050 0050      B441CC69 49C57106
02276558      IS01 DECBC  QZZ0  +77C     TYPE-SWAPBLK D-0092C320 LVL-D0      071D0000 8065B08E 003B 0035 B441CC69 49C7E786
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 2 of 6)

```
02276588      IS01 WTOPC  QZZ0  +820                                          071D0000 8065B132 00F0      B441CC69 49C99986
022765B8      IS01 WTOPC  QZZ0  +8B8                                          071D0000 8065B1CA 00F0      B441CC69 49D03E06
022765E8      IS01 WTOPC  QZZ0  +9CA                                          071D0000 8065B2DC 00F0      B441CC69 49D64686
02276618      IS01 WTOPC  QZZ0  +AC2                                          071D0000 8065B3D4 00F0      B441CC69 49DAE606
02276648      IS01 FREEC  QZZ0  +AC6     BLOCK-00A01DB0                       071D0000 8065B3D8 003B 0007 B441CC69 49DE7E46
02276678 LAST IS01 DECBC  QZZ0  +ACE     TYPE-LOCATE  D-00000000             071D0000 8065B3E0 003B 0035 B441CC69 49E04946
022766A8      IS01 GETCC  CVAA  +44C     AD-00918C00 TYPE-L1   LVL-D8        071D0000 804AB00E 002C      B441CC67 B6EE2847
022766D8      IS01 FINWC  CVAA  +218     FA-F402312D RECID-00EB LVL-D1       071D2000 804AADDA 0026      B441CC67 B6EED047
02276708      IS01 RELCC  CVAA  +250     AD-00925000 BKTYP-0051 LVL-D1       071D0000 804AAE12 0032      B441CC67 B7946841
02276738      IS01 GETPC  CVAA  +278                                          071D2000 804AAE3A 006C      B441CC67 B794B441
02276768      IS01 FINWC  CVAA  +218     FA-F4043831 RECID-00FF LVL-D1       071D2000 804AADDA 0026      B441CC67 B7952281
02276798      IS01 RELCC  CVAA  +290     AD-00925000 BKTYP-0051 LVL-D1       071D0000 804AAE52 0032      B441CC67 B868E083
022767C8      IS01 ENTRC  CVAA  +2BE     ENTER TO CVAU                                804AAE82 FF12      B441CC67 B8696743
022767F8      IS01 RELCC  CVAU  +362     AD-00918C00 BKTYP-0021 LVL-D8       071D0000 804ADC04 0032      B441CC67 B86A3403
02276828      IS01 BACKC  CVAU  +36C     RETURN TO CVAA                               804ADC10 FF02      B441CC67 B86A5D43
02276858      IS01 PROGC  CVAA  +2D4                                          071D1000 804AAE96 003B 0000 B441CC67 B86A6E03
02276888      IS01 ENTDC  CVAA  +2F6     ENTER TO CVZZ                                804AAEBA FF0E      B441CC67 B86C2C83
022768B8      IS01 ENTNC  CVZZ  +2AA     ENTER TO QZZ0                                8065A2DE FF10      B441CC67 B8743083
022768E8      IS01 CALOC  QZZ0  +016     SIZE-0000002D BLOCK-00A01F88         071D1000 8065A928 003B 0006 B441CC67 B875A703
02276918      IS01 WTOPC  QZZ0  +0C0                                          071D0000 8065A9D2 00F0      B441CC67 B878BB83
02276948      IS01 WTOPC  QZZ0  +124                                          071D0000 8065AA36 00F0      B441CC67 B87D3A43
02276978      IS01 WTOPC  QZZ0  +1CC                                          071D0000 8065AADE 00F0      B441CC67 B8812F43
022769A8      IS01 WTOPC  QZZ0  +230                                          071D0000 8065AB42 00F0      B441CC67 B884DE03
022769D8      IS01 CALOC  QZZ0  +23C     SIZE-0000002A BLOCK-00A01F50         071D0000 8065AB4E 003B 0006 B441CC67 B8888443
02276A08      IS01 WTOPC  QZZ0  +416                                          071D0000 8065AD28 00F0      B441CC67 B88AB343
02276A38      IS01 CSONC  QZZ0  +41A                                          071D0000 8065AD2C 00CA      B441CC67 B88E6D43
02276A68  L0  00906A84 009042D0           004AABC0 804AAE86          062E55E0 FF00FF00           00000000 00906AA0 L1
02276A88      00906A68 06401020           01605030 062F9E00          FF00FF00 80000000       L2  00906ABC 00906A84
02276AA8      0650F020 01605388           062F95C0 FF00FF00          80000000 00906AD8 L3        00906AA0 00000000
02276AC8      00000000 00000000           00000000 00000000       L4 00906AF4 00906ABC           00000000 00000000
02276AE8      00000000 00000000           00000000 00906B10 L5       00906AD8 00000000           00000000 00000000
02276B08      00000000 00000000       L6  00906B2C 00906AF4          00000000 00000000           00000000 00000000
02276B28      00000000 00906B48 L7        00906B10 00000000          00000000 00000000           00000000 00000000
02276B48  L8  00906B64 00906B2C           00000000 00000000          00000000 00000000           00000000 00906B80 L9
02276B68      00906B48 00000000           00000000 00000000          00000000 00000000       L10 00906B9C 00906B64
02276B88      00000000 00000000           00000000 00000000          00000000 00906BB8 L11       00906B80 00000000
02276BA8      00000000 00000000           00000000 00000000       L12 00906BD4 00906B9C          00000000 00000000
02276BC8      00000000 00000000           00000000 00906BF0 L13      00906BB8 00000000           00000000 00000000
02276BE8      00000000 00000000       L14 00906C0C 00906BD4          00000000 00000000           00000000 00000000
02276C08      00000000 00906C28 L15       00906BF0 00000000          00000000 00000000           00000000 00000000
02276C28  L16 00906C44 00906C0C           00000000 00000000          00000000 00000000           00000000 00906C60 L17
02276C48      00906C28 00000000           00000000 00000000          00000000 00000000       L18 00906C7C 00906C44
02276C68      00000000 00000000           00000000 00000000          00000000 00906C98 L19       00906C60 00000000
02276C88      00000000 00000000           00000000 00000000       L20 00906CB4 00906C7C          00000000 00000000
02276CA8      00000000 00000000           00000000 00906CD0 L21      00906C98 00000000           00000000 00000000
02276CC8      00000000 00000000       L22 00906CEC 00906CB4          00000000 00000000           00000000 00000000
02276CE8      00000000 00906D08 L23       00906CD0 00000000          00000000 00000000           00000000 00000000
02276D08  L24 00906D24 00906CEC           00000000 00000000          00000000 00000000           00000000 00906D40 L25
02276D28      00906D08 00000000           00000000 00000000          00000000 00000000       L26 00906D5C 00906D24
02276D48      00000000 00000000           00000000 00000000          00000000 00906D78 L27       00906D40 00000000
02276D68      00000000 00000000           00000000 00000000       L28 00906D94 00906D5C          00000000 00000000
02276D88      00000000 00000000           00000000 00906DB0 L29      00906D78 00000000           00000000 00000000
02276DA8      00000000 00000000       L30 00906DCC 00906D94          00000000 00000000           00000000 00000000
02276DC8      00000000 00906DE8 L31       00906DB0 00000000          00000000 00000000           00000000 00000000
02276DE8  L32 00906E04 00906DCC           00000000 00000000          00000000 00000000           00000000 00906E20 L33
02276E08      00906DE8 00000000           00000000 00000000          00000000 00000000       L34 00000000 00906E04
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 3 of 6)

```
02276E28      00000000 00000000          00000000 00000000          00000000 00906E60 DBT       00000000 00918E80
02276E48      0021017D 00000000          00000000 00000000          00000000 80000000           00906E84 00906E3C
02276E68      00000000 00000000          00000000 00000000          00000000 00000000           00000000 00906EA8
02276E88      00906E60 00000000          00000000 00000000          00000000 00000000           00000000 00000000
02276EA8      00906ECC 00906E84          00000000 00000000          00000000 00000000           00000000 00000000
02276EC8      00000000 00906EF0          00906EA8 00000000          00000000 00000000           00000000 00000000
02276EE8      00000000 00000000          00906F14 00906ECC          00000000 00000000           00000000 00000000
02276F08      00000000 00000000          00000000 00906F38          00906EF0 00000000           00000000 00000000
02276F28      00000000 00000000          00000000 00000000          00906F5C 00906F14           00000000 00000000
02276F48      00000000 00000000          00000000 00000000          00000000 00906F80           00906F38 00000000
02276F68      00000000 00000000          00000000 00000000          00000000 00000000           00000000 00906F5C
02276F88      00000000 00000000          00000000 00000000          00000000 00000000           00000000 00000000
02276FA8      00000000 00000000          00000000 00000000          00000000 00000000       SPT 01604EC0 00000002
02276FC8      00000000 00000000          00000000 00000000       PAT 045880D8 00000000           00000000 00000000
02276FE8      00000000 00000000          00000000 00000000          00000000 00000000

*EVA-TO-SVA MAP TABLE
00900000-00907000      00000000 00000000          00000000 00000000          02274000 02275000          02276000 00000000
00908000-0090F000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00918000-0091F000      01808000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00920000-00927000      00000000 00000000          00000000 00000000          00000000 01855000          00000000 00000000
00928000-0092F000      00000000 00000000          00000000 00000000          01514000 00000000          00000000 00000000
00930000-00937000      00000000 01F69000          00000000 00000000          00000000 00000000          00000000 00000000
00938000-0093F000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
009F8000-009FF000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000

*GETMAIN MAP TABLE
00A00000-00A07000      00000000 00CFD000          00000000 00000000          00000000 00000000          00000000 00000000
00A08000-00A0F000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
00AF8000-00AFF000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000

*ISOC STACK MAP TABLE
01600000-01607000      01852000 01922000          0163C000 01AC9000          017F1000 00D0B000          00CF0000 00000000
01608000-0160F000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000
016F8000-016FF000      00000000 00000000          00000000 00000000          00000000 00000000          00000000 00000000

*DECB FRAME
*BLOCK AUTHORIZED TO EVM
01514000 +000  D E C B  $ $ $ $      +008 00000000 0092 CA0      +010 00000000 00000000      +018 00000000 00000000
01514020 +020 00000000 00000000      +028 00000000 00000000      +030 00000000 00000000      +038 00000000 00010000
01514040 +040 00000000 00000000      +048 00000000 00000000      +050 00000000 00000000      +058 00000000 00000000
01514060 +060 00000000 00000000      +068 00000000 00000000      +070 DECBDECB 0092C000      +078 00000000   000000
01514080 +080  Q Z Z 0 00000000      +088 00000000 00000000      +090 00000000 00000000      +098 00000000 00000000
015140A0 +0A0 00000000 00000000      +0A8 00000000 00000000      +0B0 00000000 00000000      +0B8 00000000 00010000
015140C0 +0C0 00000000 00000000      +0C8 00000000 00000000      +0D0 00000000 00000000      +0D8 00000000 00000000
015140E0 +0E0 00000000 00000000      +0E8 00000000 00000000      +0F0 DECBDECB 0092C000      +0F8 00000000 80000000
01514100 +100  Q Z Z 0 00000000      +108 00000000 00000000      +110 00000000 00000000      +118 00000000 00000000
01514120 +120 00000000 00000000      +128 00000000 00000000      +130 00000000 00000000      +138 00000000 00010000
01514140 +140 00000000 00000000      +148 00000000 00000000      +150 00000000 00000000      +158 00000000 00000000
01514160 +160 00000000 00000000      +168 00000000 00000000      +170 DECBDECB 0092C000      +178 00000000 80000000
01514180 +180  Q Z Z 0 00000000      +188 00000000 00000000      +190 00000000 00000000      +198 00000000 00000000
015141A0 +1A0 00000000 00000000      +1A8 00000000 00000000      +1B0 00000000 00000000      +1B8 00000000 00010000
015141C0 +1C0 00000000 00000000      +1C8 00000000 00000000      +1D0 00000000 00000000      +1D8 00000000 00000000
015141E0 +1E0 00000000 00000000      +1E8 00000000 00000000      +1F0 DECBDECB 0092C000      +1F8 00000000   000000
01514200 +200  Q Z Z 0 00000000      +208 00000000 00000000      +210 00000000 00000000      +218 00000000 00000000
01514220 +220 00000000 00000000      +228 00000000 00000000      +230 00000000 00000000      +238 00000000 00010000
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 4 of 6)

```
01514240 +240 00000000 00000000      +248 00000000 00000000      +250 00000000 00000000      +258 00000000 00000000
01514260 +260 00000000 00000000      +268 00000000 00000000      +270 DECBDECB 0092C000      +278 00000000   000000
01514280 +280  Q Z Z 0 00000000      +288 00000000 00000000      +290 00000000 00000000      +298 00000000 00000000
015142A0 +2A0  D E C B  1            +2A8                        +2B0 00000000 00000000      +2B8 00000000 00010000
015142C0 +2C0 00000000 00000000      +2C8 00000000 00000000      +2D0 00000000 00000000      +2D8  D E C B 00000000
015142E0 +2E0 00000000 00000000      +2E8 00000000 00000000      +2F0 DECBDECB 0092C000      +2F8 00000000   000000
01514300 +300  Q Z Z 0 00000000      +308 00000000 00000000      +310 00000000 00000000      +318 00000000 00000000
01514320 +320  D E C B  2            +328                        +330 00000000 00000000      +338 00918E80 000101 '
01514340 +340  O M0000 00000000      +348 00046C ( 00000000      +350 00000000 00000000      +358 00000000 00000000
01514360 +360 00000000 00000000      +368 00000000 00000000      +370 DECBDECB 0092C000      +378 00000000 803E0000
01514380 +380  Q Z Z 0 00000000      +388 00000000 00000000      +390 00000000 00000000      +398 00000000 00000000
015143A0 +3A0 00000000 00000000      +3A8 00000000 00000000      +3B0 00000000 00000000      +3B8 00000000 00010000
015143C0 +3C0 00000000 00000000      +3C8 00000000 00000000      +3D0 00000000 00000000      +3D8 00000000 00000000
015143E0 +3E0 00000000 00000000      +3E8 00000000 00000000      +3F0 DECBDECB 0092C000      +3F8 00000000   000000
01514400 +400 00000000 00000000      +408 00000000 00000000      +410 00000000 00000000      +418 00000000 00000000
01514420 +420 00000000 00000000      +428 00000000 00000000      +430 00000000 00000000      +438 00000000 00010000
01514440 +440 00000000 00000000      +448 00000000 00000000      +450 00000000 00000000      +458 00000000 00000000
01514460 +460 00000000 00000000      +468 00000000 00000000      +470 DECBDECB 0092C000      +478 00000000   000000
01514480 +480 00000000 00000000      +488 00000000 00000000      +490 00000000 00000000      +498 00000000 00000000
015144A0 +4A0 00000000 00000000      +4A8 00000000 00000000      +4B0 00000000 00000000      +4B8 00000000 00010000
015144C0 +4C0 00000000 00000000      +4C8 00000000 00000000      +4D0 00000000 00000000      +4D8 00000000 00000000
015144E0 +4E0 00000000 00000000      +4E8 00000000 00000000      +4F0 DECBDECB 0092C000      +4F8 00000000   000000
01514500 +500 00000000 00000000      +508 00000000 00000000      +510 00000000 00000000      +518 00000000 00000000
01514520 +520 00000000 00000000      +528 00000000 00000000      +530 00000000 00000000      +538 00000000 00010000
01514540 +540 00000000 00000000      +548 00000000 00000000      +550 00000000 00000000      +558 00000000 00000000
01514560 +560 00000000 00000000      +568 00000000 00000000      +570 DECBDECB 0092C000      +578 00000000   000000
01514580 +580 00000000 00000000      +588 00000000 00000000      +590 00000000 00000000      +598 00000000 00000000
015145A0 +5A0 00000000 00000000      +5A8 00000000 00000000      +5B0 00000000 00000000      +5B8 00000000 00010000
015145C0 +5C0 00000000 00000000      +5C8 00000000 00000000      +5D0 00000000 00000000      +5D8 00000000 00000000
015145E0 +5E0 00000000 00000000      +5E8 00000000 00000000      +5F0 DECBDECB 0092C000      +5F8 00000000   000000
01514600 +600 00000000 00000000      +608 00000000 00000000      +610 00000000 00000000      +618 00000000 00000000
01514620 +620 00000000 00000000      +628 00000000 00000000      +630 00000000 00000000      +638 00000000 00010000
01514640 +640 00000000 00000000      +648 00000000 00000000      +650 00000000 00000000      +658 00000000 00000000
01514660 +660 00000000 00000000      +668 00000000 00000000      +670 DECBDECB 0092C000      +678 00000000   000000
01514680 +680 00000000 00000000      +688 00000000 00000000      +690 00000000 00000000      +698 00000000 00000000
015146A0 +6A0 00000000 00000000      +6A8 00000000 00000000      +6B0 00000000 00000000      +6B8 00000000 00010000
015146C0 +6C0 00000000 00000000      +6C8 00000000 00000000      +6D0 00000000 00000000      +6D8 00000000 00000000
015146E0 +6E0 00000000 00000000      +6E8 00000000 00000000      +6F0 DECBDECB 0092C000      +6F8 00000000   000000
01514700 +700 00000000 00000000      +708 00000000 00000000      +710 00000000 00000000      +718 00000000 00000000
01514720 +720 00000000 00000000      +728 00000000 00000000      +730 00000000 00000000      +738 00000000 00010000
01514740 +740 00000000 00000000      +748 00000000 00000000      +750 00000000 00000000      +758 00000000 00000000
01514760 +760 00000000 00000000      +768 00000000 00000000      +770 DECBDECB 0092C000      +778 00000000   000000
01514780 +780 00000000 00000000      +788 00000000 00000000      +790 00000000 00000000      +798 00000000 00000000
015147A0 +7A0 00000000 00000000      +7A8 00000000 00000000      +7B0 00000000 00000000      +7B8 00000000 00010000
015147C0 +7C0 00000000 00000000      +7C8 00000000 00000000      +7D0 00000000 00000000      +7D8 00000000 00000000
015147E0 +7E0 00000000 00000000      +7E8 00000000 00000000      +7F0 DECBDECB 0092C000      +7F8 00000000   000000
01514800 +800 00000000 00000000      +808 00000000 00000000      +810 00000000 00000000      +818 00000000 00000000
01514820 +820 00000000 00000000      +828 00000000 00000000      +830 00000000 00000000      +838 00000000 00010000
01514840 +840 00000000 00000000      +848 00000000 00000000      +850 00000000 00000000      +858 00000000 00000000
01514860 +860 00000000 00000000      +868 00000000 00000000      +870 DECBDECB 0092C000      +878 00000000   000000
01514880 +880 00000000 00000000      +888 00000000 00000000      +890 00000000 00000000      +898 00000000 00000000
015148A0 +8A0 00000000 00000000      +8A8 00000000 00000000      +8B0 00000000 00000000      +8B8 00000000 00010000
015148C0 +8C0 00000000 00000000      +8C8 00000000 00000000      +8D0 00000000 00000000      +8D8 00000000 00000000
015148E0 +8E0 00000000 00000000      +8E8 00000000 00000000      +8F0 DECBDECB 0092C000      +8F8 00000000   000000
01514900 +900 00000000 00000000      +908 00000000 00000000      +910 00000000 00000000      +918 00000000 00000000
01514920 +920 00000000 00000000      +928 00000000 00000000      +930 00000000 00000000      +938 00000000 00010000
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 5 of 6)

```
01514940 +940 00000000 00000000      +948 00000000 00000000      +950 00000000 00000000      +958 00000000 00000000
01514960 +960 00000000 00000000      +968 00000000 00000000      +970 DECBDECB 0092C000      +978 00000000   000000
01514980 +980 00000000 00000000      +988 00000000 00000000      +990 00000000 00000000      +998 00000000 00000000
015149A0 +9A0 00000000 00000000      +9A8 00000000 00000000      +9B0 00000000 00000000      +9B8 00000000 00010000
015149C0 +9C0 00000000 00000000      +9C8 00000000 00000000      +9D0 00000000 00000000      +9D8 00000000 00000000
015149E0 +9E0 00000000 00000000      +9E8 00000000 00000000      +9F0 DECBDECB 0092C000      +9F8 00000000   000000
01514A00 +A00 00000000 00000000      +A08 00000000 00000000      +A10 00000000 00000000      +A18 00000000 00000000
01514A20 +A20 00000000 00000000      +A28 00000000 00000000      +A30 00000000 00000000      +A38 00000000 00010000
01514A40 +A40 00000000 00000000      +A48 00000000 00000000      +A50 00000000 00000000      +A58 00000000 00000000
01514A60 +A60 00000000 00000000      +A68 00000000 00000000      +A70 DECBDECB 0092C000      +A78 00000000   000000
01514A80 +A80 00000000 00000000      +A88 00000000 00000000      +A90 00000000 00000000      +A98 00000000 00000000
01514AA0 +AA0 00000000 00000000      +AA8 00000000 00000000      +AB0 00000000 00000000      +AB8 00000000 00010000
01514AC0 +AC0 00000000 00000000      +AC8 00000000 00000000      +AD0 00000000 00000000      +AD8 00000000 00000000
01514AE0 +AE0 00000000 00000000      +AE8 00000000 00000000      +AF0 DECBDECB 0092C000      +AF8 00000000   000000
01514B00 +B00 00000000 00000000      +B08 00000000 00000000      +B10 00000000 00000000      +B18 00000000 00000000
01514B20 +B20 00000000 00000000      +B28 00000000 00000000      +B30 00000000 00000000      +B38 00000000 00010000
01514B40 +B40 00000000 00000000      +B48 00000000 00000000      +B50 00000000 00000000      +B58 00000000 00000000
01514B60 +B60 00000000 00000000      +B68 00000000 00000000      +B70 DECBDECB 0092C000      +B78 00000000   000000
01514B80 +B80 00000000 00000000      +B88 00000000 00000000      +B90 00000000 00000000      +B98 00000000 00000000
01514BA0 +BA0 00000000 00000000      +BA8 00000000 00000000      +BB0 00000000 00000000      +BB8 00000000 00010000
01514BC0 +BC0 00000000 00000000      +BC8 00000000 00000000      +BD0 00000000 00000000      +BD8 00000000 00000000
01514BE0 +BE0 00000000 00000000      +BE8 00000000 00000000      +BF0 DECBDECB 0092C000      +BF8 00000000   000000
01514C00 +C00 00000000 00000000      +C08 00000000 00000000      +C10 00000000 00000000      +C18 00000000 00000000
01514C20 +C20 00000000 00000000      +C28 00000000 00000000      +C30 00000000 00000000      +C38 00000000 00010000
01514C40 +C40 00000000 00000000      +C48 00000000 00000000      +C50 00000000 00000000      +C58 00000000 00000000
01514C60 +C60 00000000 00000000      +C68 00000000 00000000      +C70 DECBDECB 0092C000      +C78 00000000   000000
01514C80 +C80 00000000 00000000      +C88 00000000 00000000      +C90 00000000 00000000      +C98 00000000 00000000
01514CA0 +CA0 00000000 00000000      +CA8 00000000 00000000      +CB0 00000000 00000000      +CB8 00000000 00010000
01514CC0 +CC0 00000000 00000000      +CC8 00000000 00000000      +CD0 00000000 00000000      +CD8 00000000 00000000
01514CE0 +CE0 00000000 00000000      +CE8 00000000 00000000      +CF0 DECBDECB 0092C000      +CF8 00000000   000000
01514D00 +D00 00000000 00000000      +D08 00000000 00000000      +D10 00000000 00000000      +D18 00000000 00000000
01514D20 +D20 00000000 00000000      +D28 00000000 00000000      +D30 00000000 00000000      +D38 00000000 00010000
01514D40 +D40 00000000 00000000      +D48 00000000 00000000      +D50 00000000 00000000      +D58 00000000 00000000
01514D60 +D60 00000000 00000000      +D68 00000000 00000000      +D70 DECBDECB 0092C000      +D78 00000000   000000
01514D80 +D80 00000000 00000000      +D88 00000000 00000000      +D90 00000000 00000000      +D98 00000000 00000000
01514DA0 +DA0 00000000 00000000      +DA8 00000000 00000000      +DB0 00000000 00000000      +DB8 00000000 00010000
01514DC0 +DC0 00000000 00000000      +DC8 00000000 00000000      +DD0 00000000 00000000      +DD8 00000000 00000000
01514DE0 +DE0 00000000 00000000      +DE8 00000000 00000000      +DF0 DECBDECB 0092C000      +DF8 00000000   000000
01514E00 +E00 00000000 00000000      +E08 00000000 00000000      +E10 00000000 00000000      +E18 00000000 00000000
01514E20 +E20 00000000 00000000      +E28 00000000 00000000      +E30 00000000 00000000      +E38 00000000 00010000
01514E40 +E40 00000000 00000000      +E48 00000000 00000000      +E50 00000000 00000000      +E58 00000000 00000000
01514E60 +E60 00000000 00000000      +E68 00000000 00000000      +E70 DECBDECB 0092C000      +E78 00000000   000000
01514E80 +E80 00000000 00000000      +E88 00000000 00000000      +E90 00000000 00000000      +E98 00000000 00000000
01514EA0 +EA0 00000000 00000000      +EA8 00000000 00000000      +EB0 00000000 00000000      +EB8 00000000 00010000
01514EC0 +EC0 00000000 00000000      +EC8 00000000 00000000      +ED0 00000000 00000000      +ED8 00000000 00000000
01514EE0 +EE0 00000000 00000000      +EE8 00000000 00000000      +EF0 DECBDECB 0092C000      +EF8 00000000   000000
01514F00 +F00 00000000 00000000      +F08 00000000 00000000      +F10 00000000 00000000      +F18 00000000 00000000
01514F20 +F20 00000000 00000000      +F28 00000000 00000000      +F30 00000000 00000000      +F38 00000000 00010000
01514F40 +F40 00000000 00000000      +F48 00000000 00000000      +F50 00000000 00000000      +F58 00000000 00000000
01514F60 +F60 00000000 00000000      +F68 00000000 00000000      +F70 DECBDECB 0092C000      +F78 00000000   000000
01514F80 +F80 00000000 00000000      +F88 00000000 00000000      +F90 00000000 00000000      +F98 00000000 00000000
01514FE0 +FE0 00000000 00000000      +FE8 00000000 00000000      +FF0 00000000 00000000      +FF8 00000000 00000063

*DATA BLK, LEVEL 0
01F69BE0 +000  O M0000  Q Z Z 0      +008 00000000 00000000      +010 13131313 13131313      +018 13131313 13131313
01F69C00 +020 13131313 13131313      +028 13131313 13131313      +030 13131313 13131313      +038 13131313 13131313
01F69FE0 +400 13131313 13131313      +408 13131313 13131313      +410 13131313 13131313      +418 13131313 13131313
```

Figure 48. Dump of ECB Working Storage in the System Virtual Memory (SVM) (Part 6 of 6)

Table 39 explains the headings and labels that are displayed in this section of the main storage dump.

Table 39. Headings in the Dump of ECB Working Storage in the System Virtual Memory (SVM)

Heading	Description
*ENTRY BLK, GENERATED BY 1052/3215 LOCAL ENTRY	This heading identifies the particular type of ECB, based on the value of its format flag. The ECB is labeled throughout with dump labels used to identify specific fields.

Table 39. Headings in the Dump of ECB Working Storage in the System Virtual Memory (SVM) (continued)

Heading	Description
*ECB MACRO TRACE	The ECB macro trace table is a wraparound trace table with room for 55 entries when register trace is off and 23 entries when register trace is active. The register trace allows you to display registers when the macro trace is active. The enter macros cause the following values to be placed in the registers indicated: • R8 is equal to 0 for an ENTNC from CP • R10 is equal to the base of CCEB (not R10) at the time ENTxC is issued • R13 is equal to the stack register of CCEB (not R13) at the time the ENTxC is issued The header of the trace table is identified by the MTH dump label, the first fullword of which points to the next available trace entry. This pointer is not reset to the beginning of the table at ECB creation time, so as to preserve a more complete picture of system activity in the collated trace table. Each ECB trace entry has an I-stream affinity associated with it. It may be seen that this ECB switched I-streams twice before the error occurred. The remaining fields are identical to those in the collated ECB trace table.
*EVA-TO-SVA MAP TABLE	In Figure 48 on page 104, the ECB virtual address shown in the CBRW on level 0 of the ECB is X'00319C00'. Examining the EVA-to-SVA map table, you will find that the entry corresponding to the ECB virtual address X'00319000' contains the system virtual address X'00301000'. When combined with the byte-index portion of the ECB virtual address, this yields the system virtual address X'00301C00'. Scanning the attached data blocks following the ECB in the dump, we see that the data block on level 0 begins with X'00301C00'. This technique is especially useful in a dump of system storage that includes the storage pools of ECBs and 4 KB frames. Whenever the pool of ECBs is dumped, an EVA-to-SVA map table is provided for each ECB. The tables can be used to locate attached blocks embedded in individual 4 KB frames within the pool of 4 KB frames.
*GETMAIN MAP TABLE	Not Applicable.
*ISOC STACK MAP TABLE	Not Applicable.
*DECB FRAME	This heading indicates the system virtual address of the first data event control block (DECB) frame for this ECB.
*DATA BLK, LEVEL 0	Not Applicable.

Link Map Data for C Load Modules

Formatted link map data is appended directly after the C load module. This data contains the addresses of the object files and C function addresses plus the offsets of the C functions into their respective object files.

Figure 49 on page 111 shows an example of formatted link map data appended after the C load module. If the compiler time stamp and version information are not available in the C load module, that data cannot be dumped or formatted.

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

*FR SHARED PROG CDM0J1 LOADSET-LINKMAP
02885020 +000 0000FFFF  C D M 0      +008 00000000 02887AA0       +010 02887AA8 02885078      +018 00000000 00000000
02885040 +020 028887C8 00038000      +028 00000000 028876B8       +030 00000000 00000000      +038 00000000 00000000
02885060 +040 00000000 00000000      +048 00000000 00000000       +050 C3C4D4C1 C9D5D1F1      +058 00F0F026 01C3C5C5
02885080 +060 00000200 00000240      +068 47F0F001 183F58F0       +070 C31C184E 05EF0000      +078 000047F0 303A90E6
028850A0 +080 D00C58E0 D04C4100      +088 E2005500 C3144720       +090 F01458F0 C28090F0      +098 E0489210 E00050D0
   .            .        .                .        .                .        .
   .            .        .                .        .                .        .
   .            .        .                .        .                .        .
02888800 +7E0 00000000 00000000      +7E8 02885070 02888848       +7F0 028852F0 02888850      +7F8 02885788 02888858
02888820 +800 028858D0 02888860      +808 02886B38 0288888C       +810 02886CD8 02888894      +818 028871D0 0288889C
02888840 +820 028876A8 00000000      +828 00000001 02885078       +830 00000001 028852F8      +838 00000001 02885790
02888860 +840 0000000A 028858D8      +848 02885D50 02886048       +850 02886270 028863B0      +858 02886490 02886548
02888880 +860 02886600 028866E8      +868 028867A8 00000001       +870 02886B40 00000001      +878 02886CE0 00000001
028888A0 +880 028871D8

    ********* ASSOCIATED LINK MAP DATA

              02885078 OBJECT FILE = CDMAINJ1 COMPILED ON 1997/02/07 AT 19.31.20 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 02885078 CDM0

              028852F8 OBJECT FILE = CDMPRSJ1 COMPILED ON 1997/02/07 AT 19.31.40 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 028852F8 zdmap_parse

              02885790 OBJECT FILE = CDMHLPJ1 COMPILED ON 1997/02/07 AT 19.32.33 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 02885790 zdmap_help

              028858D8 OBJECT FILE = CDMDSPJ1 COMPILED ON 1997/02/10 AT 16.08.26 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 028858D8 zdmap_display
    +00000478 02885D50 process_object
    +00000770 02886048 process_function
    +00000998 02886270 display_line
    +00000AD8 028863B0 display_header
    +00000BB8 02886490 display_end
    +00000C70 02886548 cleanup
    +00000D28 02886600 free_object
    +00000E10 028866E8 free_function
    +00000ED0 028867A8 break_func_name

              02886B40 OBJECT FILE = CDMER1J1 COMPILED ON 1997/02/07 AT 19.32.53 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 02886B40 zdmap_parse_error_handler

              02886CE0 OBJECT FILE = CDMER2J1 COMPILED ON 1997/02/07 AT 19.33.41 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 02886CE0 zdmap_retrieve_error_handler

              028871D8 OBJECT FILE = CDMER3J1 COMPILED ON 1997/02/07 AT 19.33.43 V01 R02 M00

     OFFSET   ADDRESS  FUNCTION NAME
    +00000000 028871D8 zdmap_process_error_handler
```

Figure 49. Formatted Link Map Data in a Dump

The following information is appended to the C load module if no link map data is available or if the data cannot be processed:

- If the C load module has no link map data:

  `THERE IS NO LINK MAP DATA FOR THIS C LOAD MODULE.`

- If the link map address is not valid:

```
THE LINK MAP ADDRESS IS NOT WITHIN THE C LOAD MODULE ADDRESS RANGE.
NO LINK MAP DATA IS AVAILABLE.
```

- If the link map data cannot be recognized:

```
THIS C LOAD MODULE CONTAINS LINK MAP DATA THAT CANNOT BE RECOGNIZED.
```

Main Storage Dump Labels

Table 40 lists the labels found in main storage dumps to identify the storage locations of various TPF tables and work areas. The table columns have the following meanings:

Column Heading	Description
Label	Identifies the label in the dump.
References	Identifies the corresponding DSECT, CINFC label, and control program (CP) label for the dump where relevant.
Scope	Indicates whether the label is unique for a system (S), a subsystem (SS), a subsystem user (SSU), or an I-stream (IS). For example, if a dump label is unique for a SS, there is one dump label for each SS. Therefore, SS is printed in the Scope column for that dump label.
Description	Explains the dump output marked by the label.

Table 40. Labels Found in Main Storage Dumps

Label	References	Scope	Description
ACN	CMMACNUM CPMACNUM (CHSZ)	S	Activation number-the value of the activation number assigned by OPZERO to new ECBs.
AET	IDSAET (DSECT) CMMAET	S	Asynchronous event table-contains the asynchronous event queues and token information for the record cache subsystem (RCS) asynchronous event facility.
ALS	CW0CC (DSECT)	S	SNA CCW area table.
ANT	AN0NT (DSECT) CMMANT	S	Application name table (SNA and ALC terminals) contains the names of all application programs that a user can log (a maximum of 256 entries). There are 3 areas: • Application name • RCAT • Message counter.
BBT	CMMBBT	S	Buffer block table- contains the addresses of all 4 KB working storage blocks that are currently in use as part of a buffer for a tape mounted in blocked mode.
BIT	None	S	Bit map used by SNA dynamic load.
BSA	BS0AT (DSECT) CMMBSAT	S	BSC Multipoint station address table- contains each unique BSC station address to be used by a TPF system.
BVT	CMMBVT	S	Branch vector table- first byte is branch vector of current operation. Last 3 bytes are base address of the CCW area.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
CBT	IDSCCT (DSECT)	S	Common block control table- Identifies the base of the common block control table, which is used to manage the common blocks that exist within the TPF system. There is an entry for each common block within the system.
CCA		S	
CCB	ICCB (DSECT)	S	Conversation control block table used for TPF/APPC conversations.
CCC	CXLCCW (CAPT)	S	CCP CCW table- contains queues and error counts for I/O operations on communications lines.
CCT	CK6KE (DSECT) CMMCCT	S	Communications configuration table- the interface point for file-resident programs accessing CCP data and tables. The data describes the network.
CDB	DCTCDB (DSECT) MGTCDB (DCTMGT)	S	Connection definition block- contains control information. Used to define a logical connection between a pair of MPIF users generally located in separate systems.
CFC	ICFCT/c$cfct (DSECT/C Structure) CMMCFC	S	Coupling facility control table.
CFL	ICFLT/c$cflt.h (DSECT/C Structure) CMMCFL	S	Coupling facility lock table.
CFS	ICFST/c$cfst (DSECT/C Structure) ICFCTCFST (ICFCT/c$cfct)	S	Coupling facility status table.
CFT	ICFTT/c$cftt.h (DSECT/C Structure) ICFCTCFTT (ICFCT/c$cfct)	S	Coupling facility trace table.
CIO	CPMIOCD (CAPT)	S	Start of CIO CODE Area.
CLK	SC0TM (DSECT) CMMCLKCM	S	System local standard time-system clock fields.
CLT	CPSCLST (CAPT)	S	CP low core trace blocks.
CLV	DCTLVL (DSECT)	S	Core level controls- used by CCP to regulate traffic. Defines the minimum and maximum number of various control levels in the TPF system.
CLB		S	CLAW table and control block.
CLW		S	CLAW header area.
CMB		S	Common blocks- the beginning address of the common blocks in main storage.
CNF	CP0CNF (CAPT)	S	Start of CINFC Tables.
CNV		SS	Pointer to the global environment list.
CPT	CPMZZZ (CAPT)	S	CPSE internal trace.
CPY	CPMZZZ (CAPT)	S	File fallback recovery work area.
CRH	IDSCRP/c$idscrp.h (DSECT/ C Structure) CMMCRP31	S	Core-resident program area for programs allocated to run in 31-bit mode.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
CRL	IDSCRP/`c$idscrp.h` (DSECT/C Structure) CMMCRP24	S	Core-resident program area for programs allocated to run in 24-bit mode.
CRS	CR0AT (DSECT) CMCRAS CYCRAS (CAPT)	S	CRAS status table- contains the current configuration and status of all computer room agent set (CRAS) devices defined in the TPF system.
CRT	CMMCRT ICRCT (DSECT)	S	TPF transaction services control table.
CST	CMMCST	S	SON I/O control unit status table.
CTL1	DCTCTL (DSECT)	IS	CIO common control area- control information queues, save and work areas used by CIO and PIO. The 4 KB block preceding CIO code.
CTT		S	CCP trace control table.
CWA	DCTCWA (DSECT) MGTCCWA (DCTMGT)	S	MPIF CCW Area- provides the CCW area and interrupt cross reference table needed to service the channel-to-channel (CTC) support
CWP		S	CCW pointer table.
CYA	CMCYAA	S	CYAA Work Area-used to save file pool maintenance and initialization information.
DAA		S	File pool management area for device A (DEVA)
DBC	CMDFDC	SS	TPFDF data collection table.
DBK			
DBL	CMMDFL	SS	TPFDF fast-link table.
DBT	CMMDBT	SS	TPFDF central database table.
DCL	DCTDCL (DSECT)	IS	Dispatch Control List for the Ready list. The DCL is the third-level record in the CLH data structure and is used to control the CPU lists.
DCC	DCTDCR (DSECT)	IS	Dispatch Control Record for the cross list. There is 1 entry for each dispatch list defined. The DCR is the second-level record in the CLH data structure and is used to control the CPU lists.
DCD	DCTDCR (DSECT)	IS	Dispatch Control Record for the Defer list. There is 1 entry for each dispatch list defined. The DCR is the second-level record in the CLH data structure and is used to control the CPU lists.
DCI	DCTDCR (DSECT)	IS	Dispatch Control Record for the Input list. There is 1 entry for each dispatch list defined. The DCR is the second-level record in the CLH data structure and is used to control the CPU lists.
DCR	DCTDCR (DSECT)	IS	Dispatch Control Record for the Ready list. There is 1 entry for each dispatch list defined. The DCR is the second-level record in the CLH data structure and is used to control the CPU lists.
DC2	DCTDCL (DSECT)	IS	Dispatch Control List for the Input list. The DCL is the third-level record in the CLH data structure and is used to control the CPU lists.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
DC3	DCTDCL (DSECT)	IS	Dispatch Control List for the Defer list. The DCL is the third-level record in the CLH data structure and is used to control the CPU lists.
DC4	DCTDCL (DSECT)	IS	Dispatch Control List for the cross list. The DCL is the third-level record in the CLH data structure and is used to control the CPU lists.
DEB	CMMDFLD	SS	TPFDF fast-link directory.
DFD	IFLDDF (DSECT) CMMDFLD	SS	TPFDF fast-link directory.
DK1	CPMGL1 (CAPT)	S	Driver Keypoint 1.
DK2	CPMGL2 (CAPT)	S	Driver keypoint 2.
DNT	DCTDNT (DSECT) MGTDNT (DCTMGT)	S	Directory update notification table- a list of users to be notified when a specific directory update occurs.
DSC	IDSDSC (DSECT) CMMDSCA	SS	General data set control area-contains control program information used by general data set (GDS) support.
DSD	CPMJES (CAPT)	S	DISK SENSE DATA – 6 byte field used by disk error routines to read in sense data.
EAT	IDSEAT/c$idseat.h (DSECT/C Structure) CMMEAT	S	ECB activation table – tracks the number of ECBs that exist at each activation level. Used by the E-type loader to determine when resources can be returned to the system.
ECB		S	ECB blocks- the beginning address of the ECBs in main storage.
ECT	IDSECT (DSECT)	S	ECB control table – identifies the base of the ECB control table, which is used to manage the ECBs that exist within the TPF system. There is an entry for each ECB within the system.
ELD	CMMELDR	SS	E-type loader indicators.
EPO	CMMELPOL	S	E-type loader policing and long running job detection counters.
FCA	DCTFCA (DSECT) CMMFCA	S	File capture/restore low core area- filled in by the BXAR program (capture/restore). It contains exception recording indicators, BXPx ENTER expansions and the address of the CAP/RST keypoint.
FCT	IDSFCT (DSECT)	S	Frame control table- Identifies the base of the frame control table, which is used to manage the frames that exist within the TPF system. There is an entry for each frame within the system.
FC0	FC0TB (DSECT)	SS	Beginning of the FACE table.
FHD	IDSRHT (DSECT)	S	Hold table overflow area – the overflow area for hold table processing. See HPT.
FRM		S	Frame blocks – Identifies the beginning address of the frames in main storage.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
FRT	FR0RT (DSECT) CMMFRTBL	S	Fast recovery table- contains validity information plus starting and ending address of FRT records.
FS0	DCTMFS (DSECT) CMMFS0	SS	Module file status table section 0 – contains module status information (duped/ non-duped, online/offline, symbolic module number of dupe).
FS1	DCTMFS (DSECT) CMMFS1	SS	Module file status table section 1 – contains physical device address, queues, CCWs, and so on.
FS2	DCTMFS (DSECT) CMMFS2	SS	Module file status table section 2- contains data collection counters, VSNs, and count of ECBs on queue.
GFN	DCTGFN (DSECT) MGTGFND (DCTMGT)	S	Global function name directory table – contains the name of every MPIF user identified as residing in the MPIF complex.
GKP	CMMCGKP	S	Global file storage keypoint update wait switch.
GMT	CMMGMT	SS	Greenwich mean time.
GS1	GL0BA (DSECT) CPMGA1S (LOCORE expanded in CAPT)	S	I-stream shared global area 1.
GS2	CPMGA2S (LOCORE expanded in CAPT)	S	I-stream shared global area 2.
GS3	GL0BY (DSECT) CPMGA3S (LOCORE expanded in CAPT)	S	I-stream shared global area 3.
GT11	CPMGAT (LOCORE expanded in CAPT)	IS	Global attribute table 1.
GT31	CPMGAT (LOCORE expanded in CAPT)	IS	Global attribute table 3.
GU1	GL0BA (DSECT) CPMGA1U (LOCORE expanded in CAPT)	S	I-stream unique global area 1
GU2	CPMGA2U (LOCORE expanded in CAPT)	S	I-stream unique global area 2
GU3	GL0BY (DSECT) CPMGA3U (LOCORE expanded in CAPT)	S	I-stream unique global area 3
GX1I	CPMGA1X (LOCORE expanded in CAPT)	IS	Extended global area 1
GX2I	CPMGA2X (LOCORE expanded in CAPT)	IS	Extended global area 2
GX3I	CPMGA3X (LOCORE expanded in CAPT)	IS	Extended global area 3
HCT	IHCTCB (DSECT) CMMHCT	S	Hotcon table – contains a list of available hot conversations or hot connections (hotcons) used by the TPF Application Requester (TPFAR) feature.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
HMT	IHPMT (DSECT)	S	HPR message table (HPRMT) – contains a copy of each output message sent on a rapid transport protocol (RTP) connection. Output messages are removed from the HPRMT after they are acknowledged by the remote RTP endpoint. See *TPF ACF/SNA Data Communications Reference* for more information about the HPRMT.
HPT	IDSRHT (DSECT)	S	Record hold table – contains the file address, ECB address, overflow count and waiting queue (IOB address) of data records in hold status.
HSA	IHPSA (DSECT)	S	HPR session address table (HPRSAT) – contains the session addresses and the address of the resource vector table (RVT) entry for each LU-LU session established over a rapid transport protocol(RTP) connection. See *TPF ACF/SNA Data Communications Reference* for more information about the HPRSAT.
ICA	IDSICD (DSECT) CMMICD	S	ISO-C Data Area, maps the CIS0DATA area (AOLA) in CCISOC.
ICD	DCTICD (DSECT)	S	IPC connection definition block- one assigned for each connection of IPC to other processor in loosely coupled complex. Only used in a shared DASD processor support (SDPS) environment.
ICW	IPCCW (DSECT)	S	Internet Protocol (IP) channel command word (CCW) area table.
IEF	IDSEVN (DSECT)	S	INTERNAL EVENT Table – used by DEQC, ENQC, EVNTC, EVNWC and POSTC Macros. See the CP CSECT CCCIEF.
IGT	DCTIGT (DSECT) CMMIGT	S	IPC global table – contains information relaxant to communication between processors in a loosely coupled complex.
ILQ	CX#EV (DSECT) CMMKLC	S	ILQ – AI I/P link control block queue-queueing information for SLC (AI).
IMT	IPMTE (DSECT)	S	Internet Protocol (IP) message table.
IOA		S	I/O block available list.
IOB		S	IOB available list.
IPA	`i$tcpc.h`	S	Internet Protocol (IP) address processor shared table.
IPF	ISTAK (DSECT) IPFILEREC	S	Internet Protocol (IP) configuration record (containing the IP address table and the IP device table).
IPN	CK2SN (DSECT) CK2PNTA	S	Internet Protocol (IP) local port number table.
IPR	IPRTE (DSECT)	S	Base address of the Internet Protocol routing table (IPRT).
IPT	ISTAK (DSECT) IPTRACE	S	Internet Protocol (IP) trace table.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
IST	DCTIST (DSECT) CMMIST	S	I-stream STATUS table – contains data used by the cross I-stream services routine.
ITD	INDTR (DSECT)	S	TCP/IP individual Internet Protocol (IP) trace definition table.
ITL	CM1ITL CPMAC2 (CAPT)	S	INTERNAL LINE Table- contains an index value for each communications line-used to access various CCP tables.
ITT	INDTR (DSECT)	S	TCP/IP individual IP trace table.
KCM	CMMCLKCM	S	Clock support communications byte.
KPE	CK6KE (DSECT) (SCCT) CMPCCC	S	Keypoint record E – contains the system communications configuration table. The start of CTKE minus 16.
KPM	MK0CK (DSECT) CMMCTKM	S	Keypoint record M – used for MDBF system configuration definitions. Used by the IPL program each time a new subsystem configuration is generated.
KPX	CX0CK (DSECT) CMMCTKX	SS	Keypoint record X – contains control information (file address, residency allocation of all keypoints in the system, also core image restart chain address).
KP2	CK2SN (DSECT) CMMCTK2	S	SNA keypoint – contains information in the TPF system about SNA configuration (a parameter list, main storage allocation table, and system constants). Start address of CCSNA5.
KP9	CY1KR (DSECT) CMMKP9A	SS	Keypoint record 9 – SON POOL keypoint. Contains SON file pool keypoint tables, pool configuration masks, and miscellaneous counters.
K6H		SS	Keypoint record 6 Header.
LCP			
LDV	LDEVBK (DSECT)	S	Logical device blocks – CIO device control block, one defined per subchannel.
LGD	CMMLGD	S	The field that defines the difference between GMT and Local Standard Time.
LMB	ICRCT (DSECT)	S	Log manager buffer areas.
LMP	CPLKMP CMLKMPT	S	Start of the CP link map-contains CSECT name, starting address and length of CSECT.
LNS	CMMVLN CYMVLN (CAPT)	S	Line status table- format depends on line type and direction flow for full duplex lines.
LOD	CMMLODIC CPLODTAB (CICS)	S	LODIC shutdown table.
LRT	CK7KE (CTKD)	S	Synchronous link routing table- used to route type A messages to or from a high level communications network.
LST	CMMLST	SS	24-HOUR EBCDIC LOCAL TIME CLOCK.
LTT	CMMLTT	SS	Lock trace table.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
LUB	ISLUBQ (DSECT)	S	LU message blocking table-used by LUs that support FTPI.
MAC	CMMEXT CMMEXTAP	S	Macro decoder table- contains the address of each CP macro service routine - the address of the main I-stream is in CMMEXT, the address of the application I-stream is in CMMEXTAP.
MAP		S	Symbolic device address map.
MAT	DCTMAT (DSECT) CMMMAT	S	Message assembly table – used by the system message processor (SMP) to maintain pointers to long chained messages.
MCA	CPEXTAP	S	Application macro decoder table.
MCP	IMCPT (DSECT)	S	Message channel path table.
MCR	CPCTLRG (CAPT)	S	CTL register save Area – saved by CPSE.
MGT	DCTMGT (DSECT) CMMPIF	S	MPIF global table – contains parameters common to all MPIF programs (addresses of tables, routines, indicators).
MHD	CA4CT (DSECT) CM4MHD	SS	CRET minute table – contains the time of activation. ENTER expansion, name of activating program, and so on. One for each subsystem.
MIB	ISNMP/i$snmp.h(DSECT/C Structure)	S	Simple Network Management Protocol (SNMP) Management Information Base (MIB) table.
MP1	CMMCPMP1	S	System local standard time.
MSB	MGTBUFS (DCTMGT)	S	MPIF contiguous buffer area.
MSC	IMSCT/c$msct(DSECT/C Structure) ICFCTMSCT (ICFCT/c$cfct) CTLMSCT (DCTCTL)	S	Message subchannel table.
MSR	CB8HD (DSECT) MGTAIT (DCTMGT)	S	3088 address range index table.
MTR	DCTTRC (DSECT) MGTTRCA (DCTMGT)	S	MPIF trace table – contains information describing the destination, path used, and parameters associated with the MPIF user and connection.
NAT	NA0AT (DSECT)	S	Network address table – contains the network addresses of all PU 5 resources along with the associated RIDs.
NCS	CMMNCS CSNANCST	S	Network command status table – every ZNETW command creates an entry. It contains the address range of the resources affected, time stamp the message was received and an indicator of the requested function.
NEB	CM1NEBA CA1NEBA (CAPT)	S	Count of active ECBs (system) – does not include those on the defer list.
NE2	CM2NEBA CA2NEBA (CAPT)	S	Count of active ECBs (subsystem).

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
NE3	CM3NEBA CA3NEBA (CAPT)	S	Count of active ECBs (I-stream).
OCR	IOSCR (DSECT)	S	Open Systems Adapter (OSA) configuration record.
OFC	CMMFPO CPMOFC (CAPT)	S	Working storage size indicator.
OLQ		S	Synchronous link output link control block queues.
OPE	CK1KE (DSECT) CMMOPE	S	System error options – contains option indicators for the system error program (CPSE).
OPG	CK1KE (DSECT) CMMOPG	S	GFS options – contains option indicators for get file storage (GFS) (cell location, logging, tagging, and so on).
OPS	CK1KE (DSECT) CMMOPS	S	CP option indicator – contains option indicators for the CP such as, long message transmission (LMT) status and get file storage (GFS) status.
OSA	IOSAB (DSECT)	S	OSA control block table.
OSR		S	OSA read buffer table.
PAN	DCTPAN (DSECT) MGTPAN (DCTMGT)	S	MPIF PATH ACTIVATED notification table – a list of user exits to be invoked when a path is activated.
PAT	DCTPST (DSECT)	SS	Program sharing table – for file-resident programs: contains core address, indicators about status, and core block type. PAT is 4 × the number of SS (before tag).
PBT	IDSPBT (DSECT) CMMPBT	S	Physical block table – Identifies the base of the physical block table, which is used to manage the physical block types that exist within the TPF system. The table is used to answer queries about the physical blocks, and to get and release physical blocks.
PCD	IPCID (DSECT)	S	PCID table – contains an entry for each APPN LU-LU session that is being activated.
PDT	DCTPDT (DSECT) MGTPDT (DCTMGT)	S	MPIF path definition table – defines each path available to the resident system for connecting to other systems in the complex.
PHS		SS	PAT hash table.
PID	PI1DT (DSECT) CMMPID CPMPIDT (CAPT)	S	Processor ID table – maintains status of multiple processors in a TPF complex (CPU ID, CPUID ordinal number, IPL indicators, path ID, and so on).
PIU	IPTBL (DSECT) CMMPIUC CINFC_CMMPIUC	S	Path information unit (PIU) trace table – contains a copy of the data transferred between the TPF system and remote SNA resources. See *TPF ACF/SNA Data Communications Reference* for more information about the PIU trace table.
PKP	CMPSTP CPMPKSTP (CAPT)	S	PKST table pointer – contains the addresses of the PKSTs (communications control unit keypoint status records).

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
PKS	CPTIC (DSECT) CMPKST CPMPTCCT (CAPT)	S	Communications control unit keypoint status Record- contains the status of all communications control units in the system.
PMT	CMMPMGT	S	SON pool management table.
PNT		S	Socket port number tables.
PSD		S	IRB/TAPE sense data-IRB sense and status save Area (returned by PIO). Tag CCDPIOSV in CEDT of CCCPSE.
PST	CMMPST CYPOLL (CAPT)	S	POLL status table
PSV	IXPSV (DSECT)	S	Process selection vector (PSV) name table
RCB	IDSSST (DSECT) CMMSSTB	S	RCS subsystem status table-base – contains RCS subsystem entry information for subsystems whose IDs hash directly into the primary (base) entry area.
RCH	IDSSST (DSECT) CMMSSTH	S	RCS subsystem status table-header-contains control information for the memory-resident SSST data structure.
RCO	IDSSST (DSECT) CMMSSTO	S	RCS subsystem status table-overflow-contains RCS subsystem entry information for subsystems whose ID hash values conflict with existing table entries in the base area. Access is sequential in this area.
RFN	DCTRFN (DSECT) MGTRFND (DCTMGT)	S	MPIF resident function name directory – contains the name of each MPIFuser that has identified itself to the TPF processor.
RHC	IRNHCT (DSECT)	S	Resource name hash control table (RNHCT) – contains information about the resource vector table (RVT) available list, RVT termination list, resource name hash prime table (RNHPT), and resource name hash entry table (RNHET).
RHE	IRNHET (DSECT)	S	Resource name hash entry table (RNHET) – contains pointers to the entries in the resource vector table (RVT).
RHP	IRNHPT (DSECT)	S	Resource name hash prime table (RNHPT) – contains hashing buckets for the resource name hash entry table (RNHET).
RIT	DCTRIT (DSECT) CMMRIT	SS	Record identifier attribute table (RIAT) – contains the file ID, exception recording, logging, and restoring status, record attribute descriptors, user exit information, VFA candidacy, locking status, and record caching candidacy.
RTP	IRTPB (DSECT)	S	RTP control block (RTPCB) table – contains information about each rapid transport protocol(RTP) connection established for the TPF processor. See *TPF ACF/SNA Data Communications Reference* for more information about the RTPCB table.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
RV1	RV1VT (DSECT)	S	Resource vector table part 1 – contains the definition and status of the nodes.
RV2	RV2VT (DSECT)	S	Resource vector table part 2 – contains dynamic information and status of the nodes.
SAT	SA0AT (DSECT)	S	Subarea address table – contains status and other information related to each subarea defined in the network (PU5).
SBA		S	OSA storage block address table.
SCI	CMMSCYI	S	Subsystem cycle indicator.
SCT	CONKC (MACRO) CMMSCT	SS	System configuration table – contains system configuration information required by TPF programs (file address of CTKA, number of SEL CHNLs, and so on)
SC1	ISCB (DSECT)	S	Session control block table part 1 – extension of the RVT1 used by TPF/APPC sessions.
SC2	ISCB (DSECT)	S	Session control block table part 2 – extension of the RVT2 used by TPF/APPC sessions.
SDD	ISDDCB (DSECT) CMMSDD (215)	SS	SQL database directory – contains the list of SQL relational databases to which the TPF Application Requester (TPFAR) feature can connect.
SEA	MRLNQ (DSECT)	S	SON entry address table – contains pointers for the addresses of error routines CJIX and CJIY.
SEL		S	
SG0	SI0GT (DSECT) CMSIGT0 CPMSIGT0 (CAPT)	S	System interprocessor global table – used to synchronize global fields/records among I-streams where the user is active.
SHC	ISHCT (DSECT)	SS	System heap control table (SHCT) – contains an entry for each 4 KB of system heap virtual address space. Each entry contains information returned from a GSYSC macro call.
SHD	CA4CT (DSECT) CM4SHD CA4BAS (CAPT)	S	CRET second table – contains the time to be activated, ENTER expansion, activating program name, and so on. One per system.
SHM	IDSHM (DSECT)	S	Shared memory table.
SIB	None	S	SNA I/O buffers – assigned for NCP read operations and channel-to-channel (CTC) read and write operations.
SIT	None	S	Session identifier table – contains entry for each ALS and the SIDs for each session through an ALS.
SKCI		IS	CROSC stack.
SKEI	DCTSTK (DSECT) DCTPFX (PFXESAVE)	IS	External interrupt save stack.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
SKFI	DCTSTK (DSECT) DCTPFX (PFXFSAVE)	IS	Fast-link save stack.
SKII	DCTSTK (DSECT) DCTPFX (PFXNSAVE)	IS	I/O interrupt save stack.
SKMI	DCTSTK (DSECT) DCTPFX (PFXMSAVV)	IS	PFXMSAVV contains the virtual address of the machine check interruption stack.
SKNI			
SKPI	DCTSTK (DSECT) DCTPFX (PFXPSAVE)	IS	Program check save stack.
SKRI	DCTSTK (DSECT) DCTPFX (PFXRSAVE)	IS	Restart save stack.
SKSI	DCTSTK (DSECT) DCTPFX (PFXSSAVE)	IS	SVC save stack.
SKZI	DCTSTK (DSECT) DCTPFX (PFXZSAVE)	IS	CPSE I/O save stack.
SLN	SLSTL (DSECT) CMSLST CSYLST (CAPT)	S	Symbolic line status table – prime control table for the non-SNA communications network. It contains line control indicators, transmission routing and data control information.
SLS		S	OSA storage list structure table.
SMT		S	
SNA	CK2SN (DSECT) CMMSNA	S	SNA parameter list area within keypoint record 2.
SNC	SN0CT (DSECT) CMMSNCT	S	BSC station name conversion table – each entry defines how many connections exist to a station connection address and a unique 4-character station name.
SNF	CMMSNF	S	Sequential number fields (SNF) table – used to correlate responses to TPF requests sent over a CDRM-CDRM session.
SNL		S	SNA local element bit map.
SNP		S	SNP.
SNS	CK2SN	S	Sense code table.
SNT		S	
SOK	ISOCK (DSECT)	S	Socket block table.
SPM	CY5GT (DSECT) CMMPMGT	S	SON pool management global table – contains constants, counters, and addresses necessary for the functioning of SON pool management.
SPT	IEQCE3 (DSECT) CE3SPTR	S	ISO-C stack pointer.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
SPY	CMPACNFL	S	SON copy pointer – used during a copy function to force all files to be behind the copy pointer (field CJFCPTR in CJIO).
SRT	SR0RT (DSECT)	S	System recovery table – logs input and output messages in the system. Performs SNA message recovery.
SSA	MS0AT (DSECT) CMMMDBF CMMSSAT CPMSSAT (CAPT)	S	Subsystem attribute table – contains subsystem configuration data for an MDBF system. Field CPMSSAT in CAPT.
SSC	DCTSCT (DSECT) MGTSSCT (DCTMGT)	S	System-to-system control table – used to manage logical queues/paths for a specific processor connected to other processors using MPIF.
SSL	`i$ssl.h` (C Structure)	S	SSL control table – contains statistical information about Secure Sockets Layer (SSL) usage and shared SSL sessions.
SSU	MS0UT (DSECT) CMMMASU CMMSSUT CPMSSUT (CAPT)	S	Subsystem users table – contains information about each subsystem in an MDBF system (user ID, mnemonic, and so on).
STI	CMMSTI CPMCLEX (CAPT)	S	System state indicators – tells current state of system (cycling to/from state, and so on).
SWB		S	SWB blocks – The beginning address of the SWBs that are in main storage.
SWT	IDSSCT (DSECT)	S	SWB control table – Identifies the base of the SWB control table, which is used to manage the SWBs that exist within the TPF system. There is an entry for each SWB within the system.
TAP	ITAPST (DSECT)	S	TPF Advanced Peer-to-Peer Networking (TPF/APPN) processor shared table – contains information about all APPN links that are active in the loosely coupled complex.
TAT		S	Tape assignment table.
TBF	DCTDUC (CEFS) CADTBF (CAPT)	S	3480 deferred unit check recovery area – used by tape CSECT for 3480 deferred unit check processing.
TCT	`i$tcpc.h` (C Structure)	S	Transmission Control Protocol/Internet Protocol (TCP/IP) configuration table.
TCX	CMMTCX	S	Tape control unit cross-reference table – (COSY) contains TSTB rechain flag, item count and CU/CHNL addresses (fields defined in TAPEQ).
TGT		S	Tape group definition table.
TIS	CMOTB2 CYTITB (CAPT)	S	Terminal interchange status table – contains error counters, poll status flag, and so on. Layout different for BSC, HS, and LS.

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
TMA	ICRCT (DSECT)	S	Transaction manager (TM) transaction anchor table (TANC).
TMS	CMMTMSLC ITSNT (DSECT)	S	Time slice name table.
TPI	ITPICB (DSECT)	S	Transaction program instance control block.
TSD	CPMBSD (CCTAPE)	S	Tape sense data save area – CPMBSD in CCTAPE.
TSW	CPMTASSA (CAPT)	S	Subsystem TAS switch.
TS1	ITSTB (DSECT) CMMTS1 CADTS1 (CAPT)	S	Tape status table section 1 – contains device addresses, symbolic tape names, tape status, VSNs.
TS2	ITSTB (DSECT) CMMTS2 CADTS2 (CAPT)	S	Tape status table section 2 – contains queue pointers, counters, and error recovery flags.
TS3	ITSTB (DSECT) CMMTS3 CADTS3 (CAPT)	S	Tape status table section 3 – contains the buffer control block and the main storage copy of the tape label.
UCL	DCTUCL (DSECT) CMMUCL CPMAUCL (CAPT)	S	User exit control list – a list of defined user exit points. Contains the address and status of each user exit routine associated with an exit point.
URT	UR1ST (DSECT) CMMURT CPMUBASE (CAPT)	S	UNIT record status table – contains information showing the real-time status of the unit record devices in the system.
US1		SSU	Subsystem user I-stream shared global area 1.
US2		SSU	Subsystem user I-stream shared global area 2.
US3		SSU	Subsystem user I-stream shared global area 3.
UT1		SSU, IS	Subsystem user global attribute table 1.
UT3		SSU, IS	Subsystem user global attribute table 3.
UU1		SSU	Subsystem user I-stream unique global area 1.
UU2		SSU	Subsystem user I-stream unique global area 2.
UU3		SSU	Subsystem user I-stream unique global area 3.
UX1		SSU, IS	Subsystem user extended global area 1.
UX2		SSU, IS	Subsystem user extended global area 2.
UX3		SSU, IS	Subsystem user extended global area 3.
UST	IUSAT (DSECT)	S	User storage allocation table – contains addresses for user-defined tables and storage areas.
VCT	VF0AC (DSECT) CMMVFAC	S	VFA control area-control information (CPUID, Number of subsystems at IPL, table of SSU names).

Table 40. Labels Found in Main Storage Dumps (continued)

Label	References	Scope	Description
VST	CMMFAL	S	Pointer to start of VFA – contains the address of field CK9FAL in CTKB.
WGT	WG0TA (DSECT) CMMWGT	S	WGTA table – an entry for every LIT or LEID in the system. It contains the ordinal number of the associated AAA/RCB, RID (if a SNA terminal) and the application (APPL) the terminal is logged to.
WLT	CM2WLT CA2WLT (CAPT)	S	Tape ECB wait list – tape ECB is placed on list when an unusual condition occurs.
WRT	IDSELT/`c$idselt.h` (DSECT/C Structure) CMMWRT	SS	E-type loader working record table – contains the ordinal numbers of E-type loader fixed-file records that are available for use by a processor.
XID	IOTBL (DSECT) CSNAIOTB	S	SNA I/O trace table.
XPT	IDSXTP/`c$idsxtp.h` (DSECT/C Structure) CMMXPAT	S	Extra program allocation table (PAT) slot area-extra PAT slots that are used by the E-type loader.
XSW	CMMXSW CPMXSW (CCCCP1) LINEQ	S	CCP poll control switch – low core and input inhibited switches. Switch masks are defined in LINEQ.
X25		S	X25 keypoint.
37K		S	3705 keypoint, device Q(S).

Dump Labels for the Entry Control Block (ECB)

The following dump labels identify fields in the ECB.

Table 41. Dump Labels for the Entry Control Block (ECB)

Label	Description
AUT	ALASC auto – storage block save area.
AVM	Maximum times to avoid application time-out plus one.
AV1	Base of the ECB available queue for 128-/381-byte blocks.
AV2	Base of the ECB available queue for 1055-byte blocks.
AV4	Base of the ECB available queue for 4095-byte blocks.
BAD	Post-interrupt branch address.
BCH	SVM address of the previous ECB in a chain of thread ECBs in a process.
CHW	ECB chain word.
CNF	Address of the BSS CINFC table.
CPL	ECB virtual address of current program nesting level.
CPX	CCP status word. Last byte is the CPU ID of the CPU that caused the ECB.
CRP	Core block reference word of active program block.

Table 41. Dump Labels for the Entry Control Block (ECB) (continued)

Label	Description
CRS	CROSC save area.
CR0–CRF	Core block reference words.
CSC	Pointer to static storage control block for C language support.
CSP	Pointer to C static storage.
CTL	Count of I/O requests outstanding.
CXR	Enter expansion to be executed at CXFRC post-interrupt time.
DBI	The MDBF database identifier.
DBT	The detached block table.
DEC	ECB virtual address of the first data event control block (DECB) frame for this ECB.
DET	Address of current detached block table entry.
EGS	EVM address of the last page table for the ISO-C stack of this ECB.
EHT	ECB hold table pointer.
EVA	The EVA of the ECB.
FAP	File address reference word of active program block.
FA0–FAF	File address reference words.
FCH	SVM address of the next ECB in the chain of thread ECBs in a process.
FLG	ECB format flag.
FPL	ECB virtual address of first program nesting level.
FRM	Anchor for the ECB frame chain.
FX0–FXF	File address reference word extensions. Used for general data set (GDS) support.
GLA	Pointer to the global 1 directory for this ECB's I-stream and subsystem user.
GLY	Pointer to the global 3 directory for this ECB's I-stream and subsystem user.
GLB	System virtual address of the start of the ECB segment table entries for the I-stream unique primary globals.
GLX	System virtual address of the start of the ECB segment table entries for the high storage primary globals.
GMH	System virtual address of the high water mark page table entry for contiguous GETMAIN storage.
GMT	System virtual address of ECB page table for contiguous GETMAIN storage.
HFC	Heap management – head of frame chain.
ISN	I-stream affinity of the ECB.
L0–L34	ECB program nesting levels.
MTH	ECB macro trace table header.
NVA	The offset into the ECB page table of the next available ECB virtual address in the ECB private area.
PSW	Contents of the current PSW the last time a TPF SVC or fastlink macro was issued. This PSW is used to resume application program execution when the macro service routine completes its processing.

Table 41. Dump Labels for the Entry Control Block (ECB) (continued)

Label	Description
PAT	Current PAT slot address.
PGT	The system virtual address of the ECB's page table.
R8	Address of the currently active program.
R14–R7	Contents of the general registers R14–R7 at the time of the last SVC of fastlink macro call.
RET	Number of frames removed from this ECB since ECB creation.
SBI	MDBF program base identifier.
SGT	The system virtual address of the ECB's segment table.
SON	SON decode field.
SSU	MDBF subsystem user identifier.
STA	EVM address of the start of the ISO-C stack area of this thread.
STD	The segment table designation for the ECB's virtual address space.
STK	Pointer to the current C language stack frame.
STP	Address of the TPFAR STP block.
SUD	Detail data level indicators – one byte for each level.
SUP	Program level error indicator.
SVA	The SVA of the ECB.
SVT	The system virtual address table for this ECB.
TCA	Base of the C language task communication area.
TCB	EVM address of the thread control block (TCB) of this ECB.
TFC	Heap management – total frames in use.
TMC	Transaction manager control record (TMCR).
TOD	Contents of TOD clock at time of ECB creation.
TRC	Pointer to the OPZERO program that created this ECB, or name of the program that created this ECB through CRET/CREM/CRED/CREX.
TRT	Pointer to the C language translate table.
TSA	The address of the segment table save area (STSA) block for this ECB.
TTA	Test tools area.
TYP	The ECB type, where: • 00–normal ECB • 01–master thread ECB • 02–nonmaster thread ECB.
URA–UR7	User save area for general registers R14–R7.
USA0–U104	User work area (if present).
USR	Area in ECB page 2 reserved for TPF users.
VCT	Count of trips to VFA.
WSW1	General purpose switches.
W000–W096	ECB work area.
XSW0	Secondary work area switches.
X000–X096	Secondary work area.

Dump Labels for the ISO-C Control Structures

The dump labels that follow identify fields in the entry control block (ECB). See the IDSDSA data macro.

Only those registers used by the compiler, and not necessarily all registers, are valid in the stack label information. For the information in the registers to be valid, the registers must have been saved by the compiler. See the compiled listing of the program in question for a list of the registers saved. This area is of marginal use for debugging because of this specialized preservation of register information.

Table 42. Dump Labels for ISO-C Control Structures

Label	Description
ALA	Array of LIBVEC addresses (AOLA) pointer.
BOS	C language beginning of stack pointer.
CED	C language enclave level data control block (CEDB) pointer.
CID	Base of the C language implementation data (CID) area.
EOS	C language end of stack pointer.
EPT	C language address of entry point function.
ISL	Initial stack allocation length.
LWS	Local working storage pointer.
TBA	TOBEY area pointer.
TCA	Base of the C language task communication area (TCA).
TIA	Implementation-defined task area pointer.
TPS	C language temporary program storage area pointer.
WSC	Base of the C language writable static control area.
WSP	C language current writable static storage pointer.

Dump Labels for the ISO-C Stack Area

Table 43. Dump Labels for the ISO-C Stack Area

Label	Description
BKP	Backward pointer (old stack frame) for C language.
GSH	High-water mark for stack page table.
GSP	Stack page table.
ISO	Task communication area (TCA).
NAB	Next available byte (next stack frame).
R12	Last register saved in the DSA (STM instruction) for C language.
R14	First register saved in the DSA (STM instruction) for C language.
SPT	Current stack pointer.

Processing Overview

Figure 50 provides an overview of DOF utility processing. The figure shows both the system error processing for online processing and offline postprocessing.

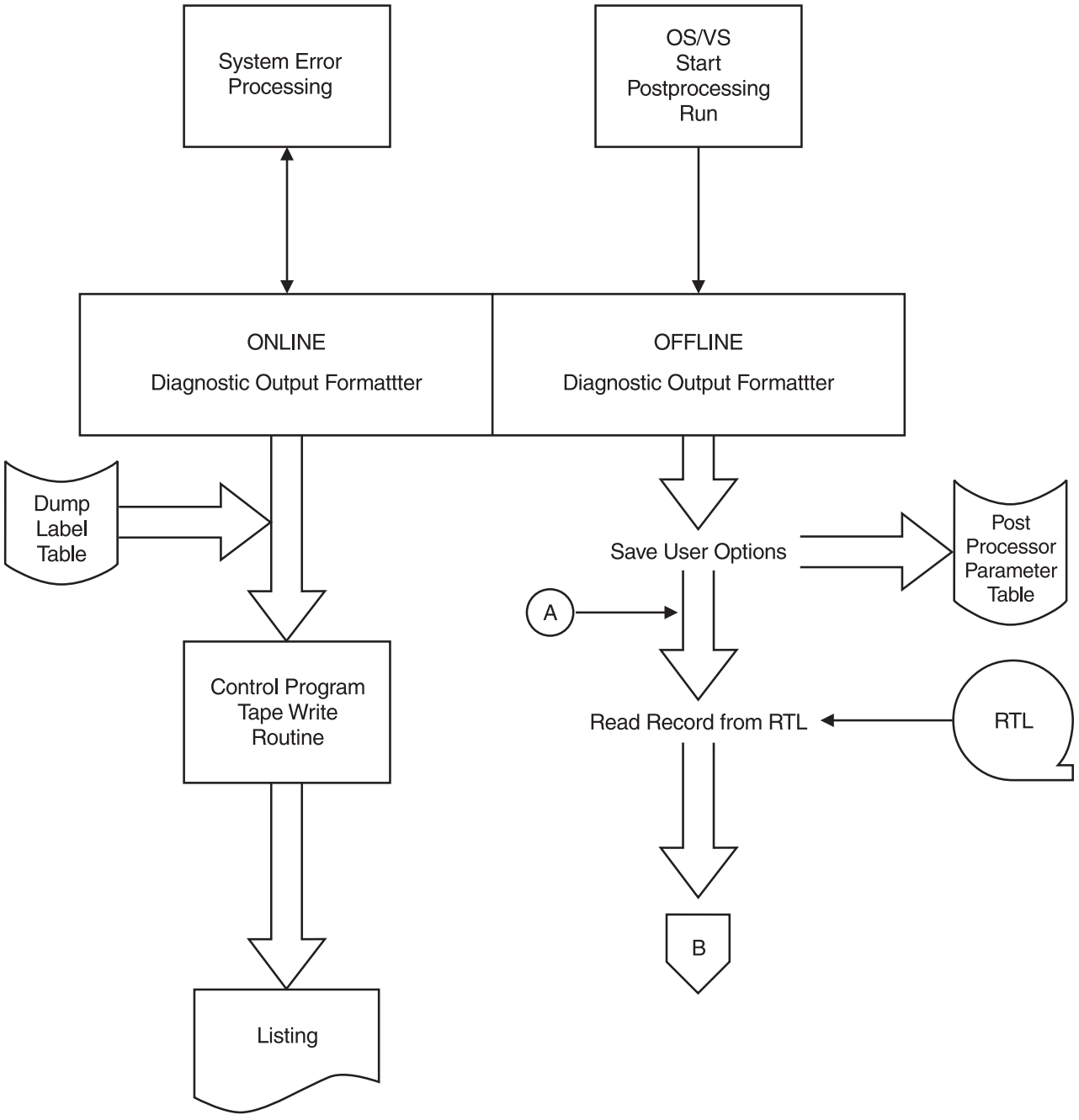

Figure 50. Diagnostic Output Formatter Processing Overview (Part 1 of 2)

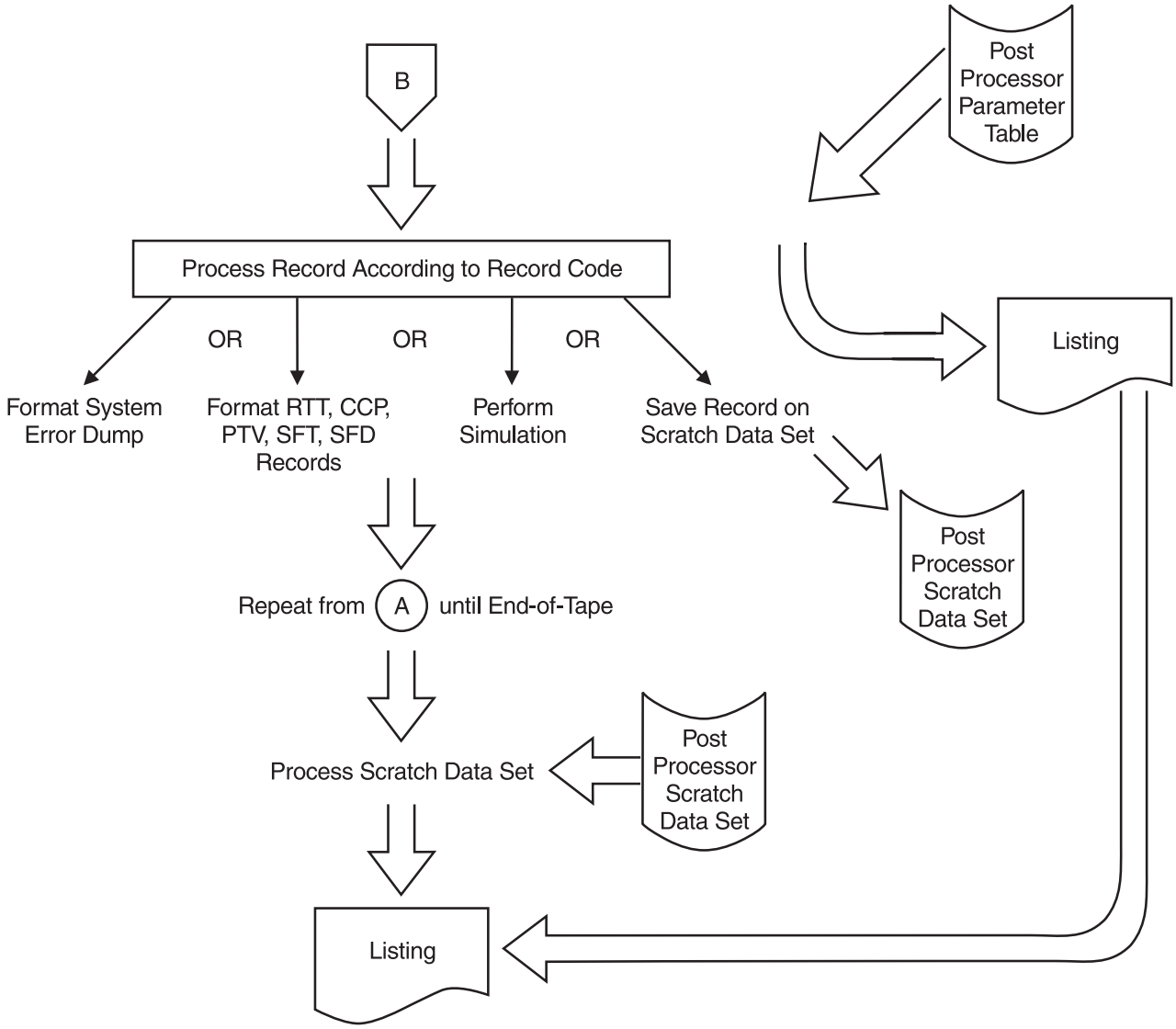

Figure 50. Diagnostic Output Formatter Processing Overview (Part 2 of 2)

C Function Trace

Use C function trace to trace ISO-C programs that have been compiled using the TEST compiler option of one of the IBM C/370 family of compilers supported by the TPF 4.1 system. C function trace provides relevant information to help analyze C program problems, therefore reducing the development cycle of these C programs.

C function trace uses *breakpoints* generated by the compiler when a program is compiled using the TEST compiler option. Breakpoints are grouped into types, for example, program entry, program exit, and others. The default values of these breakpoints are no-operation instructions.

If the C function trace of program entry breakpoints and program exit breakpoints is enabled by specifying the CDEBUG parameter on the ZSTRC command or by specifying the ENTRY and EXIT parameters on the ENATC macro, the program entry breakpoints and the program exit breakpoints are modified so that the C function trace is run.

If the C function trace of breakpoints other than program entry breakpoints and program exit breakpoints is enabled by specifying the CDEBUG and XHOOKS parameters on the ZSTRC command or by specifying the OTHERS parameter on the ENATC macro, the other breakpoints are modified so that the C function trace is run for these breakpoints.

The trace environment is established when:

- An ECB issues an active breakpoint for the first time
- The trace table storage is allocated.
- C function trace starts.

The major elements of the C function trace environment are:

- Trace options
- Trace table and user trace area sizes
- Trace breakpoint statements and breakpoint related data in the task communication area (TCA).

See “Controlling Traces” on page 134 for more information about how these elements are set up and controlled.

C function trace provides trace information for each ECB, for:

- Internal C functions
- Dynamic link modules (DLMs)
- C library functions.

When C function trace is active, each ECB that invokes C functions has its own C function trace table. Once the trace is started, subsequent attempts to alter or reset the trace environment have no effect on the currently running ECB. However, you can stop and start C function trace at any time using the ENATC macro.

Summary of Command and Macro Options for C Function Trace

C function trace allows you to trace the current ECB or another ECB. Table 44 provides an overview of the options of the ZSTRC command, the SETTC macro, and the ENATC macro you can use to control the output of C function trace.

Table 44. Manipulation of C Function Trace and Trace Suboptions

Option	Trace all ECBs	Trace an Individual ECB
Program entry breakpoint	ZSTRC ALTER CDEBUG	ENATC ENTRY=ON
Program exit breakpoint	ZSTRC ALTER CDEBUG	ENATC EXIT=ON
Other breakpoints	ZSTRC ALTER CDEBUG XHOOKS	ENATC OTHERS=ON
STACK option	ZSTRC ALTER CDEBUG STACK	SETTC STACK=ON
STATIC option	ZSTRC ALTER CDEBUG STATIC	SETTC STATIC=ON

See *TPF Operations* for more information about commands and *TPF System Macros* for more information about system macros.

Controlling Traces

You can control C function trace by issuing appropriate commands or macros in user exits or in the real-time assembler code.

You can control various aspects of C function trace such as:

- Selecting programs to be traced at compile time
- Setting system wide trace options
- Controlling specific ECBs
- Selectively activating loadsets.

Selecting Programs to Be Traced at Compile Time

At compile time, select the ISO-C programs to be traced. Compile these programs using the TEST compiler option, any desired suboptions of the TEST compiler option, and the desired compiler optimization level, dependent on the features of the compiler being used. Load each of the compiled and linked programs into the system to become part of the base system.

Notes:

1. The BLOCK, LINE, and PATH suboptions of the TEST compiler option are supported by C function trace, dependent on the features of the compiler being used.
2. If you compile using the TEST compiler option and using an optimization level other than OPT(0) or NOOPTIMIZE, the compiler generates program entry breakpoints and program exit breakpoints only.

Setting System Wide Trace Options

Use the ZSTRC command to control tracing on a system wide level.

Using the ZSTRC Command

Use the ZSTRC command to:

- Enable or disable the C function trace for all ECBs

- Enable or disable tracing of C function stack and static areas at program entries and program exits for all ECBs.

Start C function trace and set trace options using these parameters.

CDEBUG
The CDEBUG parameter enables C function trace for ISO-C programs compiled using the TEST compiler option.

The NOCDEBUG parameter disables the C function trace for ISO-C programs compiled using the TEST compiler option.

STACK
Places up to 68 bytes of the C function stack data in the trace table when the CDEBUG parameter is active. The information is displayed in the dump.

If you specify NOSTACK, only the address of the stack area is placed in the trace table when the CDEBUG parameter is active.

If the CDEBUG parameter is not active, STACK or NOSTACK will not trace any information.

STATIC
Places up to 68 bytes of the C function static data in the trace table when the CDEBUG parameter is active. The information is displayed in the dump.

If you specify NOSTATIC, only the address of the static area is placed in the trace table when the CDEBUG parameter is active.

If the CDEBUG parameter is not active, STATIC or NOSTATIC will not trace any information.

XHOOKS
Enables the C function trace of breakpoints other than program entry breakpoints and program exit breakpoints, for ISO-C programs compiled using the suboptions of the TEST compiler option.

NOXHOOKS disables the C function trace of breakpoints other than program entry breakpoints and program exit breakpoints, for ISO-C programs compiled using the suboptions of the TEST compiler option.

If the CDEBUG parameter is not active, XHOOKS will not trace any information.

The tracing of breakpoints other than program entry breakpoints and program exit breakpoints can enhance your ability to debug problems in your C code.

See *TPF Operations* for more information about the ZSTRC command.

Controlling Specific ECBs

Use the SETTC and ENATC macros to set trace options and start tracing either the current ECB or other ECBs in the TPF 4.1 system. The C function trace environment is unique for each ECB.

Using the SETTC Macro

Issue the SETTC macro in the CDEB user exit to:

- Enable or disable tracing of C program static and stack areas for specific ECBs
- Alter the size of the C function trace table
- Define a size for a C function trace user area.

Note: Once a C function trace table has been allocated for an ECB, the SETTC macro has no effect, and an error indication code is returned in register 15.

Using the ENATC Macro

Issue the ENATC macro to:

- Enable or disable the C function trace at program entries and program exits to C programs for a specific ECB
- Enable or disable other traces for a specific ECB.

You can issue the ENATC macro at any time.

Using User Exits

The C function trace user exits provide additional flexibility for controlling tracing for specific ECBs. For example, you can issue the SETTC and ENATC macros from user exits to set options and start tracing a specific ECB. The C function trace user exits are as follows.

User Exit	Description
CDEB	Trace environment customization
CEXP	User trace area initialization
CTRC	Trace user data.

See “Customizing Trace Information” on page 151 for more information about these user exits.

Activating Loadsets with the E-Type Loader

Use the ZOLDR ACTIVATE command to activate a loadset that contains C programs that have been compiled using the TEST compiler option.

E-type loader functions allow you to control the C function trace in production and test environments. By using the E-type loader you can set up C function trace for specific load modules or specific ECB origins (for example, terminals, lines, and user IDs).

Tracing in a Production Environment

To limit trace data to specific ISO-C programs, use the ZOLDR ACTIVATE command to activate a loadset that contains ISO-C programs compiled using the TEST compiler option.

Tracing by ECB Origin

To control which ECB has access to a specific load module, use the ZOLDR ACTIVATE command with the SEL parameter. This will selectively activate a loadset that contains ISO-C C programs that were compiled using the TEST compiler option. Selective activation allows you to control which ECB origins have access to the specific load module that contains the ISO-C C program. See *TPF Operations* for more information about the ZOLDR ACTIVATE command.

System Error Processing

On the first page of a system dump is the line, `SYSTEM TRACES ACTIVE`, which indicates whether C function trace is active and which trace options are specified. The CDEBUG, STACK, STATIC, and XHOOKS parameters are displayed on this line if they have been specified previously by the ZSTRC command. System dumps

are similar to the sample dumps shown starting with "Trace Output without Stack and Static Data" on page 138 except that the sample dumps do not have the `SYSTEM TRACES ACTIVE` line.

Selecting ECBs for Formatting and Dumping C Function Trace Entries

Because of performance considerations, only certain ECBs are formatted and dumped. The criteria for formatting and dumping ECBs follows:

- C function trace entries for the *active* ECB in EVA are formatted using system error processing and included in the dump.
- C function trace entries for the *active* ECB when the dump took place in the SVM are ***not*** formatted and are not included in the dump.
- C function trace entries for *non-active* ECBs are ***not*** formatted and are not included in the dump.

C Function Trace User Area

The C function trace user area (for each ECB) is ***not*** dumped if the area exists. The C function trace user area is allocated using the USERSIZE parameter on the SETTC macro. You can allocate space and put whatever trace data you desire into that area. The C function trace table header contains the address and length of the user area. It is your responsibility to format and print the C function trace user area if desired.

C Function Trace Table

The C function trace table is allocated automatically the first time you use it.

Other Breakpoints

When breakpoints other than program entry breakpoints and program exit breakpoints are compiled into the C programs and turned on by entering **ZSTRC ALTER CDEBUG XHOOKS** or by issuing the ENATC macro with the OTHERS=ON parameter specified, and when the assembler breakpoint instruction is invoked, the control is transferred to the CTRY copy member of the CCISOC module. CTRY builds a trace table entry in the C function trace table for breakpoints other than program entry breakpoints and program exit breakpoints.

The CID fields that follow here are updated to reflect relevant information of the current trace entry for tracing of program entry breakpoints, program exit breakpoints, and other breakpoints.

CID Field	Description
ICID_NLEV	Nesting level
ICID_FUN	Current function name address
ICID_FUNL	Current function name length
ICID_LMSZ	Current load module size
ICID_STKA	Caller's C automatic stack frame address
ICID_STKL	Caller's C automatic stack frame length
ICID_STCA	Current load module C static area address
ICID_STCL	Current load module C static area length.

Format and Description of C Function Trace

This section shows the format and content of typical C function trace output. The types of output formats are:

- Standard output. The standard C function trace data or output is a single entry without stack and static data.
- Extended output. For each additional STACK or STATIC option that you specify, 68 bytes of data will be displayed. The extended C function trace output formats are created when you specify the STACK, STATIC, or both options using the ZSTRC command or the SETTC macro.

  Based on the options you specify, you receive the following output:

  – C function trace table output with stack and static data
  – C function trace table output with stack and without static data
  – C function trace table output without stack and with static data.

Samples of each of these different types of output follow. After the first sample trace, an explanation of the alpha pointers (A) through (J) is provided along with other relevant information.

There is an example of the output you receive if the calling or return parameters are not addressable.

There also is an example of output with breakpoint entries other than program entry breakpoints and program exit breakpoints. This example contains a user library version of the `printf` library function that was compiled using the suboptions of the TEST compiler option, and therefore trace entries for `printf` display in the output. Although the `vsprintf` and `puts` library functions were compiled without using the TEST compiler option, they display in the output as

- HOOK..CALLRET from `vsprintf`
- HOOK..CALLRET from `puts`

because the `printf` library function contains CALLRET breakpoints. The C function trace entry for the CALLRET breakpoint displays the name of the function that was called rather than the name of the calling function. This feature is useful for tracing calls to functions that were not compiled using the TEST compiler option.

Trace Output without Stack and Static Data

Figure 51 on page 139 shows the trace output without stack and static data.

```
*ISOC C FUNCTION TRACE ENTRIES
    TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

 (A)     (B)    (C)   (D)                (E)                          (F)              (G)
*LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS          **STACK  **STATIC *TIME STAMP
 QPM0 00000494 0001 QPM0
                                00000007 00000077 0032C368 00000006 02001B80 02403010 AAF49435 504FAA41
 ???? 8175D646 0002 memset
                                02001D71 00000082 00000007 01939580 02001D58 02403010 AAF49435 E429C543
 ???? 8175D67C 0002 --Return-   from   memset
                                02001D71                            02001E28 02403010 AAF49436 5C6BD844
 QPM1 00000494 0002 QPM1
                                80000000 0000FF00 4047F00F 8A47F00F 02001B80 02405010 AAF49436 F098B842
 ???? 8175F5DE 0003 printf
                                02405020 FFFFFFFF 43202000 00000000 02001D58 02405010 AAF49437 68F45C40
 ???? 8175F68C 0003 --Return-   from   printf
                                00000011                            02001E18 02405010 AAF49438 BD72C344
 QPM1 0000423E 0003 funcfloat_routine
                                02001D98 FFFFFFFF 43202000 00000000 02001D58 02405010 AAF49439 4F28A843
 QPM1 0000426C 0003 --Return-   from   funcfloat_routine
                                02001BD0                            02001E18 02405010 AAF49439 E0BFB943
 ???? 8175F5DE 0003 printf
                                02405032 41100000 00000000 00000000 02001D58 02405010 AAF4943A 7488F441
 ???? 8175F68C 0003 --Return-   from   printf
                                0000001F                            02001E18 02405010 AAF4943B 1D7A7244
 QPM1 000016A6 0003 func1
                                00000001 02405010 00000000 00000000 02001D58 02405010 AAF4943B 99FC1C43
 QPM1 0000178C 0003 --Return-   from   func1
                                0000000A                            02001E18 02405010 AAF4943C 29FF9540
 QPM1 00001BF8 0003 functio3
                                00000000 00000005 00000000 00000000 02001D58 02405010 AAF4943C BC99C842
 QPM1 000016A6 0004 func1
                                00000001 02405010 10F0F04B 02001DA0 02001E18 02405010 AAF4943D 4D4DC140
 QPM1 0000178C 0004 --Return-   from   func1
                                0000000A                            02001ED0 02405010 AAF4943D F7CD4D40
 QPM1 000019CE 0004 func2
                                00000002 02405010 10F0F04B 02001DA0 02001E18 02405010 AAF4943E 713A6C40
 QPM1 00001A6C 0004 --Return-   from   func2
                                00000014                            02001ED0 02405010 AAF4943F 047E6A42
 QPM1 000024C6 0004 function_name_which_is_truncated_because_it_contains_several_characte+
                                00000005 02405010 10F0F04B 02001DA0 02001E18 02405010 AAF4943F 95FE0340
 QPM1 000016A6 0005 func1
                                00000001 02405010 10404040 02001E58 02001ED0 02405010 AAF49440 124CF741
 QPM1 0000178C 0005 --Return-   from   func1
                                0000000A                            02001F88 02405010 AAF49441 3572BE40
 QPM1 000019CE 0005 func2
                                00000002 02405010 10404040 02001E58 02001ED0 02405010 AAF49444 27973144
 QPM1 00001A6C 0005 --Return-   from   func2
                                00000014                            02001F88 02405010 AAF4944D 2FB03941
*QPM1 0000268C 0004 --Return-   from   function_name_which_is_truncated__because_it_conta+
                                00000000                            02001E18 02405010 AAF4943F 95FE0340
```

Figure 51. Trace Output without Static or Stack Data

Description of Alpha Pointers

Table 45 on page 140 provides a description of the alpha pointers.

Table 45. Descriptions of Alpha Pointers

Alpha Pointer	Description
(A)	LODM — The load module field stores the current load module where the breakpoint resides. ???? — Indicates the breakpoint involved is inside a library program that resides in a separate load module, not the current load module. * (an asterisk) — Precedes the last trace entry in the trace table. Since the trace table wraps around when full, the last entry may appear anywhere within the trace entries. See “Trace Output with Stack and Static Data” on page 142 for an example of wrapping.
(B)	OFFSET, which is the offset field is 8 characters long. It is the offset of the breakpoint in the load module. When the load module name is indicated as ????, the offset field is the address of the next sequential instruction in the load module where the library program resides.
(C)	The nesting counter (LV), which is the level of nesting counter for calls to and returns from those C functions compiled using the TEST compiler option. The LV starts with zero when the C program trace starts and is reset to zero when the trace is stopped and started again. Each time a program entry breakpoint is processed the LV is incremented by one. Each time a program exit breakpoint is processed the LV is decremented by one if the current LV value is non-zero. For all other breakpoints, the LV is neither incremented nor decremented. See “Interpreting Trace Nesting Levels” on page 142 for more information.
(D)	FUNCTION — The function name field depends on the particular breakpoint: • For a program entry breakpoint trace table entry, the function name field in the table holds a function name up to 70 characters long. If a function name is longer than 70 characters, the function name is truncated to 69 characters with a plus sign (+) appended at the end to indicate the truncation. See “Trace Output without Stack and Static Data” on page 138 for an example. • For all trace table entries except for a program entry breakpoint trace table entry, the function name field in the table holds a function name up to 51 characters long. If a function name is longer than 51 characters, the function name is truncated to 50 characters with a plus sign (+) appended at the end to indicate the truncation. See “Trace Output without Stack and Static Data” on page 138 for an example.

Table 45. Descriptions of Alpha Pointers (continued)

Alpha Pointer	Description
(D) – Continued	FUNCTION The function name field depends on the particular breakpoint: • For a program exit breakpoint trace table entry, the function name follows the character string: --Return- from • For breakpoints other than program entry breakpoints and program exit breakpoints, the function name follows a breakpoint name as generated by the compiler, possibly combined with the current (calling) or called function; for example, HOOK..TRUEIF func1 • For the HOOK..CALLBGN breakpoint, the function name of the current (calling) function follows the character string: HOOK..CALLBGN by For the HOOK..CALLRET breakpoint, the function name of the called function follows the character string: HOOK..CALLRET from **Note:** The HOOK..CALLRET breakpoint trace table entry contains the called function name and ***not*** the current function name in order to provide you with additional debugging information. Otherwise, if the called function had been compiled without breakpoints, it would be difficult to determine what function had been called because the called function would not generate a trace table entry. See “Trace Output with Other Breakpoint Entries” on page 149 for trace output that contains trace table entries for breakpoints other than program entry breakpoints and program exit breakpoints.
(D) – Continued	???????? Indicates that C function trace code cannot determine the function name. If you use `#define` to map the name, the remapped name appears in the function field. If you use `#pragma` map to map the name, the original source file name appears in the function field.
(E)	CALLING/RETURN PARAMETERS For a program entry breakpoint trace table entry and a HOOK..CALLBGN trace table entry, the calling parameter field indicates up to four parameter values. For a program exit breakpoint trace table entry and a HOOK..CALLRET trace table entry, the return parameter field indicates the return value. For all other trace table entries, no parameter values and no return value are displayed because these values are meaningless for all other breakpoints. ???????? Indicates that C function trace code cannot determine if parameter values have been passed. C function trace code issues a load real address (LRA) instruction. On a non-zero condition code from the LRA instruction, C function trace will place question marks (????????) in the field. See “Question Marks (????????) in the CALLING/RETURN PARAMETERS Field” on page 150 for trace output that shows the usage of ????????.

Table 45. Descriptions of Alpha Pointers (continued)

Alpha Pointer	Description

(F)	STACK STATIC — Specifies that up to 68 bytes of the stack or static user data is displayed in the output.
	NOSTACK — Specifies that the address of the stack data is displayed, but no stack data is displayed. In the trace table output, NOSTACK is shown as:
	`>NOSTACK >>STACK@ xxxxxxxx`
	where: *xxxxxxxx* is the stack address.
	NOSTATIC — Specifies that the address of the static area is displayed, but no static data is displayed. In the actual trace table output, NOSTATIC is shown as:
	`NOSTATIC >STATIC@ yyyyyyyy`
	where: *yyyyyyyy* is the static address.

(G)	TIME STAMP, which is the value from a STCK instruction issued at the time of creation of the trace table entry.
(H)	TABLE ADDRESS/LENGTH, which is the table address and length fields contain the address and length of the C function trace table.
(I)	USER AREA ADDRESS/LENGTH. which is the user area address and length fields contain the address and length of the optional C function trace user area.
(J)	STK@=/STAT@= where the the STK@= field contains the address of the stack data when stack data is displayed. The STK@= field is not displayed when stack data is not displayed. The STAT@= field contains the address of the static data when static data is displayed. The STAT@= field is not displayed when static data is not displayed.

Interpreting Trace Nesting Levels

When interpreting the data in the nesting field (LV), you should consider:

- Whether or not all C programs in all load modules are compiled using the TEST compiler option.

  In some cases only some and not all of the C programs are compiled using the TEST compiler options. The LV field represents the *relative* nesting level of the calls to and returns from the C programs being traced. The nesting level counter does not represent the exact calling sequence of the actual program flow.
- Whether or not trace has been enabled, disabled, started, and stopped multiple times and at what point the trace is enabled, started, or restarted.

  Because C program trace can be enabled, disabled, started, stopped, and restarted at any point of the program when an ECB is running, the nesting level counter may not reflect the actual entries and exits.

Trace Output with Stack and Static Data

Figure 52 on page 143 shows the trace output with stack and static data.

```
*ISOC C FUNCTION TRACE ENTRIES
    TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

 (A)     (B)    (C)   (D)                (E)                      (F)                                  (G)
*LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS         **STACK  **STATIC                  *TIME STAMP
 QPM1 000016A6 0005 func1                                                                  STK@=02001ED0 STAT@=02405010 (J)
                              00000001 02405010 10000000 02001E58 >>STACK> 00000000 00000000 00000000 AAF48B6B 86F7A142
                                                                  00000000 00000000 00000000 86A49583
                                                                  F440A2A3 81839200 00000004 00000005
                                                                  00000006 00000000 00000000 00000001
                                                                  02405010
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
 QPM1 0000178C 0005 --Return-   from   func1                                               STK@=02001F88 STAT@=02405010
                              0000000A                            >>STACK> 00307000 800F55FA 00001000 AAF48B6C 21A37C43
                                                                  00000000 02001BD0 020020D0 86A49583
                                                                  F140A2A3 81839200 00000001 00000002
                                                                  00000003 00000004 00000000
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
 QPM1 000019CE 0005 func2                                                                  STK@=02001ED0 STAT@=02405010
                              00000002 02405010 10000000 02001E58 >>STACK> 00000000 00000000 00000000 AAF48B6C 956F9D41
                                                                  00000000 00000000 00000000 86A49583
                                                                  F440A2A3 81839200 0000000A 00000005
                                                                  00000006 00000000 00000000 00000002
                                                                  02405010
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
*QPM1 00001A6C 0005 --Return-   from   func2                                               STK@=02001F88 STAT@=02405010
                              00000014                            >>STACK> 00307000 800F55FA 00001000 AAF48B6D 0DC76043
                                                                  00000000 02001BD0 020020D0 86A49583
                                                                  F240A2A3 81839200 00000002 00000003
                                                                  00000003
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
 ???? 8175F5DE 0003 printf                                                                 STK@=02001D58 STAT@=02405010
                              02405020 A3A44841 42520000 00000000 >>STACK> 00000000 00000000 00000000 AAF48B62 0A441C41
                                                                  00000000 00000000 00000000 00000004
                                                                  82828282 C2C2C2C2 C2C2C2C2 00000000
                                                                  86CFC11D 02405020 A3A44841 42520000
                                                                  00000000 00000000
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
 ???? 8175F68C 0003 --Return-   from   printf                                              STK@=02001E18 STAT@=02405010
                              00000011                            >>STACK> 00000000 02001D70 00000000 AAF48B63 11C8BA44
                                                                  8172600C 02001D88 02001EA4 00000000
                                                                  00000000 00000011 00000001 A385A2A3
                                                                  89958740 D8D7D4F1 C14B4B4B 1500F97F
                                                                  4F24C272 02001E48
                                                                  >>STATIC 000B0000 00000001 00000000
                                                                  00000005 A385A2A3 89958740 D8D7D4F1
                                                                  C14B4B4B 150086A4 95838693 9681A3F6
                                                                  409985A3 A49995A2 7E6C864B 4B4B1500
                                                                  00000000 D8D7D2F3
```

Figure 52. Trace Output with Stack and Static Data (Part 1 of 3)

```
QPM1 0000423E 0003 funcfloat_routine                                        STK@=02001D58 STAT@=02405010
                           02001D98 A3A44841 42520000 00000000 >>STACK> 00000000 00000000 00000000 AAF48B63 A9625C44
                                                               00000000 00000000 00000000 00000004
                                                               82828282 C2C2C2C2 C2C2C2C2 00000000
                                                               86CFC11D 02001D98 A3A44841 42520000
                                                               00000000 00000000
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 0000426C 0003 --Return-   from   funcfloat_routine                      STK@=02001E18 STAT@=02405010
                           02001BD0                            >>STACK> 00000000 02001D70 00000000 AAF48B64 1FEFCF43
                                                               8172600C 02001D88 02001EA4 86A49583
                                                               86939681 A3F640A2 A3818392 0085A2A3
                                                               89958740
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
???? 8175F5DE 0003 printf                                                   STK@=02001D58 STAT@=02405010
                           02405032 41100000 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF48B64 AEE85A42
                                                               00000000 00000000 00000000 00000004
                                                               82828282 C2C2C2C2 C2C2C2C2 41100000
                                                               86CFC11D 02405032 41100000 00000000
                                                               00000000 41100000
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
???? 8175F68C 0003 --Return-   from   printf                               STK@=02001E18 STAT@=02405010
                           0000001F                            >>STACK> 00000000 02001D70 00000000 AAF48B65 45D7D841
                                                               8172600C 02001D88 02001EA4 00000000
                                                               00000000 0000001F A3818392 86A49583
                                                               86939681 A3F64099 85A3A499 95A27EF1
                                                               4BF0F0F0 F0F0F04B
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 000016A6 0003 func1                                                    STK@=02001D58 STAT@=02405010
                           00000001 02405010 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF48B65 BC791B40
                                                               00000000 00000000 00000000 00000004
                                                               82828282 C2C2C2C2 C2C2C2C2 41100000
                                                               86CFC11D 00000001 02405010 00000000
                                                               00000000 41100000
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 0000178C 0003 --Return-   from   func1                                 STK@=02001E18 STAT@=02405010
                           0000000A                            >>STACK> 00000000 02001D70 00000000 AAF48B66 5355DF44
                                                               8172600C 02001D88 02001EA4 86A49583
                                                               F140A2A3 81839200 00000001 00000002
                                                               00000003 00000004 85A3A499
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
```

Figure 52. Trace Output with Stack and Static Data (Part 2 of 3)

```
QPM1 00001BF8 0003 functio3                                                    STK@=02001D58 STAT@=02405010
                           00000000 00000005 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF48B66 C850EE41
                                                               00000000 00000000 00000000 00000004
                                                               82828282 C2C2C2C2 C2C2C2C2 41100000
                                                               86CFC11D 00000000 00000005 00000000
                                                               00000000 41100000
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 000016A6 0004 func1                                                       STK@=02001E18 STAT@=02405010
                           00000001 02405010 10F0F04B 02001DA0 >>STACK> 00000000 02001D70 00000000 AAF48B67 4A263D42
                                                               8172600C 02001D88 02001EA4 86A49583
                                                               A38996F3 40A2A381 83920001 00000003
                                                               00000004 00000005 00000000 00000001
                                                               02405010
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 0000178C 0004 --Return-   from   func1                                    STK@=02001ED0 STAT@=02405010
                           0000000A                            >>STACK> 00000000 00000000 00000000 AAF48B67 D4429241
                                                               00000000 00000000 00000000 86A49583
                                                               F140A2A3 81839200 00000001 00000002
                                                               00000003 00000004 00000000
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 000019CE 0004 func2                                                       STK@=02001E18 STAT@=02405010
                           00000002 02405010 10F0F04B 02001DA0 >>STACK> 00000000 02001D70 00000000 AAF48B68 6597F144
                                                               8172600C 02001D88 02001EA4 86A49583
                                                               A38996F3 40A2A381 83920001 0000000A
                                                               00000004 00000005 00000000 00000002
                                                               02405010
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 00001A6C 0004 --Return-   from   func2                                    STK@=02001ED0 STAT@=02405010
                           00000014                            >>STACK> 00000000 00000000 00000000 AAF48B6A 7EBCFB44
                                                               00000000 00000000 00000000 86A49583
                                                               F240A2A3 81839200 00000002 00000003
                                                               00000003
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
QPM1 000024C6 0004 function_name_which_is_truncated_because_it_contains_several_characte+ STK@=02001E18 STAT@=02405010
                           00000005 02405010 10F0F04B 02001DA0 >>STACK> 00000000 02001D70 00000000 AAF48B6A F66A2E43
                                                               8172600C 02001D88 02001EA4 86A49583
                                                               A38996F3 40A2A381 83920001 00000003
                                                               00000004 00000005 00000001 00000005
                                                               02405010
                                                               >>STATIC 000B0000 00000001 00000000
                                                               00000005 A385A2A3 89958740 D8D7D4F1
                                                               C14B4B4B 150086A4 95838693 9681A3F6
                                                               409985A3 A49995A2 7E6C864B 4B4B1500
                                                               00000000 D8D7D2F3
```

Figure 52. Trace Output with Stack and Static Data (Part 3 of 3)

Trace Output with Stack and without Static Data

Figure 53 on page 146 shows the trace output with stack and without static data.

```
*ISOC C FUNCTION TRACE ENTRIES
    TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

 (A)     (B)    (C)   (D)                (E)                        (F)                                  (G)
*LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS          **STACK  **STATIC                  *TIME STAMP
 QPM0 00000494 0001 QPM0                                                                                STK@=02001B80 (J)
                               00000007 00000077 0032C368 00000006 >>STACK> 00000000 00000000 00000000 AAF49353 3D6E7F44
                                                                   000007FC 0032C390 00140010 00010000
                                                                   00000002 0032C390 0032C3A8 D8D7D4F0
                                                                   FFFFFF00 000000B0 0193AFA8 00000000
                                                                   00000000 00000000
                                                                   NOSTATIC >STATIC@ 02403010
 ???? 8175D646 0002 memset                                                                              STK@=02001D58
                               02001D71 00000082 00000007 01939580 >>STACK> 00000000 00000000 00000000 AAF49353 B55DA940
                                                                   00000000 00000000 00000000 C1C2C2C2
                                                                   C2C2C2C2 C2C2C2C2 C2C2C2C2 01724F68
                                                                   02001E18 02001E10 02001D70 8177C3E8
                                                                   00000000 02001D71
                                                                   NOSTATIC >STATIC@ 02403010
 ???? 8175D67C 0002 --Return-   from   memset                                                            STK@=02001E28
                               02001D71                            >>STACK> 01727758 02001EA4 02001E10 AAF49354 177DB642
                                                                   02001D70 02001D98 00000001
                                                                   NOSTATIC >STATIC@ 02403010
 QPM1 00000494 0002 QPM1                                                                                STK@=02001B80
                               80000000 0000FF00 4047F00F 8A47F00F >>STACK> 01939930 02403010 02001E30 AAF49354 910AB240
                                                                   00000007 02404000 81939A04 02403010
                                                                   02403010 00000000 00303000 D8D7D4F1
                                                                   FFFFFF00 0000014C 0193FAF8 00000000
                                                                   00000000 00000000
                                                                   NOSTATIC >STATIC@ 02405010
 ???? 8175F5DE 0003 printf                                                                              STK@=02001D58
                               02405020 FFFFFFFF 43213000 00000000 >>STACK> 00000000 00000000 00000000 AAF49354 F2CA9041
                                                                   00000000 00000000 00000000 00000004
                                                                   82828282 C2C2C2C2 C2C2C2C2 00000000
                                                                   B404B725 02405020 FFFFFFFF 43213000
                                                                   00000000 00000000
                                                                   NOSTATIC >STATIC@ 02405010
 ???? 8175F68C 0003 --Return-   from   printf                                                            STK@=02001E18
                               00000011                            >>STACK> D5000000 02001D70 40404040 AAF49359 0BFDF443
                                                                   8172600C 02001D88 02001EA4 00000000
                                                                   00000000 00000011 00000001 A385A2A3
                                                                   89958740 D8D7D4F1 C14B4B4B 1500F97F
                                                                   D48101E1 02001E48
                                                                   NOSTATIC >STATIC@ 02405010
 QPM1 0000423E 0003 funcfloat_routine                                                                   STK@=02001D58
                               02001D98 FFFFFFFF 43213000 00000000 >>STACK> 00000000 00000000 00000000 AAF49366 10C21940
                                                                   00000000 00000000 00000000 00000004
                                                                   82828282 C2C2C2C2 C2C2C2C2 00000000
                                                                   B404B725 02001D98 FFFFFFFF 43213000
                                                                   00000000 00000000
                                                                   NOSTATIC >STATIC@ 02405010
 QPM1 0000426C 0003 --Return-   from   funcfloat_routine                                                 STK@=02001E18
                               02001BD0                            >>STACK> D5000000 02001D70 40404040 AAF49368 FCF53643
                                                                   8172600C 02001D88 02001EA4 86A49583
                                                                   86939681 A3F640A2 A3818392 0085A2A3
                                                                   89958740
                                                                   NOSTATIC >STATIC@ 02405010
 ???? 8175F5DE 0003 printf                                                                              STK@=02001D58
                               02405032 41100000 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF4936B C6861743
                                                                   00000000 00000000 00000000 00000004
                                                                   82828282 C2C2C2C2 C2C2C2C2 41100000
                                                                   B404B725 02405032 41100000 00000000
                                                                   00000000 41100000
                                                                   NOSTATIC >STATIC@ 02405010
```

Figure 53. Trace Output with Stack and without Static Data (Part 1 of 2)

```
???? 8175F68C 0003 --Return-   from   printf                                                    STK@=02001E18
                           0000001F                          >>STACK> D5000000 02001D70 40404040 AAF4936E 71579640
                                                             8172600C 02001D88 02001EA4 00000000
                                                             00000000 0000001F A3818392 86A49583
                                                             86939681 A3F64099 85A3A499 95A27EF1
                                                             4BF0F0F0 F0F0F04B
                                                             NOSTATIC >STATIC@ 02405010
QPM1 000016A6 0003 func1                                                                       STK@=02001D58
                           00000001 02405010 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF4936F 1AE13442
                                                             00000000 00000000 00000000 00000004
                                                             82828282 C2C2C2C2 C2C2C2C2 41100000
                                                             B404B725 00000001 02405010 00000000
                                                             00000000 41100000
                                                             NOSTATIC >STATIC@ 02405010
QPM1 0000178C 0003 --Return-   from   func1                                                     STK@=02001E18
                           0000000A                          >>STACK> D5000000 02001D70 40404040 AAF4936F AD62A741
                                                             8172600C 02001D88 02001EA4 86A49583
                                                             F140A2A3 81839200 00000001 00000002
                                                             00000003 00000004 85A3A499
                                                             NOSTATIC >STATIC@ 02405010
QPM1 00001BF8 0003 functio3                                                                    STK@=02001D58
                           00000000 00000005 00000000 00000000 >>STACK> 00000000 00000000 00000000 AAF49370 721EF843
                                                             00000000 00000000 00000000 00000004
                                                             82828282 C2C2C2C2 C2C2C2C2 41100000
                                                             B404B725 00000000 00000005 00000000
                                                             00000000 41100000
                                                             NOSTATIC >STATIC@ 02405010
QPM1 000016A6 0004 func1                                                                       STK@=02001E18
                           00000001 02405010 10F0F04B 02001DA0 >>STACK> D5000000 02001D70 40404040 AAF49372 D0153743
                                                             8172600C 02001D88 02001EA4 86A49583
                                                             A38996F3 40A2A381 83920001 00000003
                                                             00000004 00000005 00000000 00000001
                                                             02405010
                                                             NOSTATIC >STATIC@ 02405010
QPM1 0000178C 0004 --Return-   from   func1                                                     STK@=02001ED0
                           0000000A                          >>STACK> F0404B4B 4B4B4B4B 4B4B1519 AAF49375 31617C43
                                                             FF326E80 0021017D 00000000 86A49583
                                                             F140A2A3 81839200 00000001 00000002
                                                             00000003 00000004 00000000
                                                             NOSTATIC >STATIC@ 02405010
QPM1 000019CE 0004 func2                                                                       STK@=02001E18
                           00000002 02405010 10F0F04B 02001DA0 >>STACK> D5000000 02001D70 40404040 AAF49376 25E93741
                                                             8172600C 02001D88 02001EA4 86A49583
                                                             A38996F3 40A2A381 83920001 0000000A
                                                             00000004 00000005 00000000 00000002
                                                             02405010
                                                             NOSTATIC >STATIC@ 02405010
*QPM1 00001A6C 0004 --Return-   from   func2                                                    STK@=02001ED0
                           00000014                          >>STACK> F0404B4B 4B4B4B4B 4B4B1519 AAF49376 E7447E40
                                                             FF326E80 0021017D 00000000 86A49583
                                                             F240A2A3 81839200 00000002 00000003
                                                             00000003
                                                             NOSTATIC >STATIC@ 02405010
```

Figure 53. Trace Output with Stack and without Static Data (Part 2 of 2)

Trace Output with Static and without Stack Data

Figure 54 on page 148 shows the trace output with static and without stack data.

```
* ISOC C FUNCTION TRACE ENTRIES
     TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
 USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

  (A)     (B)    (C)   (D)                (E)                        (F)                                  (G)
 *LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS          **STACK  **STATIC                  *TIME STAMP
  QPM0 00000494 0001 QPM0                                                                              STAT@=02403010 (J)
                                00000007 00000077 0032F368 00000006 >NOSTACK >>STACK@ 02001B80          AAF493EE A500BA40
                                                                    >>STATIC C2C2C2C2 C2C2C2C2 C2C2C2C2
                                                                    C2C2C200 D8D7D5F2 00D8D7D5 F240C5D5
                                                                    E3C4C340 86818993 A4998540 60409596
                                                                    409985A3 A4999540 A39640D8 D7D4F04B
                                                                    40150000 D8D7D2F2
  ???? 8175D646 0002 memset                                                                            STAT@=02403010
                                02001D71 00000082 00000007 01939580 >NOSTACK >>STACK@ 02001D58          AAF493EF 25BA0A40
                                                                    >>STATIC C2C2C2C2 C2C2C2C2 C2C2C2C2
                                                                    C2C2C200 D8D7D5F2 00D8D7D5 F240C5D5
                                                                    E3C4C340 86818993 A4998540 60409596
                                                                    409985A3 A4999540 A39640D8 D7D4F04B
                                                                    40150000 D8D7D2F2
  ???? 8175D67C 0002 --Return-   from   memset                                                         STAT@=02403010
                                02001D71                            >NOSTACK >>STACK@ 02001E28          AAF493EF 9A9F5740
                                                                    >>STATIC C2C2C2C2 C2C2C2C2 C2C2C2C2
                                                                    C2C2C200 D8D7D5F2 00D8D7D5 F240C5D5
                                                                    E3C4C340 86818993 A4998540 60409596
                                                                    409985A3 A4999540 A39640D8 D7D4F04B
                                                                    40150000 D8D7D2F2
  QPM1 00000494 0002 QPM1                                                                              STAT@=02405010
                                80000000 0000FF00 4047F00F 8A47F00F >NOSTACK >>STACK@ 02001B80          AAF493F0 12B4D044
                                                                    >>STATIC 000B0000 00000001 00000000
                                                                    00000005 A385A2A3 89958740 D8D7D4F1
                                                                    C14B4B4B 150086A4 95838693 9681A3F6
                                                                    409985A3 A49995A2 7E6C864B 4B4B1500
                                                                    00000000 D8D7D2F3
  ???? 8175F5DE 0003 printf                                                                            STAT@=02405010
                                02405020 FFFFFFFF 43297000 00000000 >NOSTACK >>STACK@ 02001D58          AAF493F0 92FD2740
                                                                    >>STATIC 000B0000 00000001 00000000
                                                                    00000005 A385A2A3 89958740 D8D7D4F1
                                                                    C14B4B4B 150086A4 95838693 9681A3F6
                                                                    409985A3 A49995A2 7E6C864B 4B4B1500
                                                                    00000000 D8D7D2F3
  ???? 8175F68C 0003 --Return-   from   printf
                                                                                                       STAT@=02405010
                                00000011                            >NOSTACK >>STACK@ 02001E18          AAF493F2 0153CC44
                                                                    >>STATIC 000B0000 00000001 00000000
                                                                    00000005 A385A2A3 89958740 D8D7D4F1
                                                                    C14B4B4B 150086A4 95838693 9681A3F6
                                                                    409985A3 A49995A2 7E6C864B 4B4B1500
                                                                    00000000 D8D7D2F3
  QPM1 0000423E 0003 funcfloat_routine                                                                 STAT@=02405010
                                02001D98 FFFFFFFF 43297000 00000000 >NOSTACK >>STACK@ 02001D58          AAF493F2 8BC27342
                                                                    >>STATIC 000B0000 00000001 00000000
                                                                    00000005 A385A2A3 89958740 D8D7D4F1
                                                                    C14B4B4B 150086A4 95838693 9681A3F6
                                                                    409985A3 A49995A2 7E6C864B 4B4B1500
                                                                    00000000 D8D7D2F3
  QPM1 0000426C 0003 --Return-   from   funcfloat_routine                                              STAT@=02405010
                                02001BD0                            >NOSTACK >>STACK@ 02001E18          AAF493F3 04B63042
                                                                    >>STATIC 000B0000 00000001 00000000
                                                                    00000005 A385A2A3 89958740 D8D7D4F1
                                                                    C14B4B4B 150086A4 95838693 9681A3F6
                                                                    409985A3 A49995A2 7E6C864B 4B4B1500
                                                                    00000000 D8D7D2F3
```

Figure 54. Trace Output with Static and without Stack Data (Part 1 of 2)

```
???? 8175F5DE 0003 printf                                                                    STAT@=02405010
                          02405032 41100000 00000000 00000000 >NOSTACK >>STACK@ 02001D58          AAF493F3 8214F140
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
???? 8175F68C 0003 --Return-   from   printf                                                  STAT@=02405010
                          0000001F                            >NOSTACK >>STACK@ 02001E18          AAF493F4 BCA50D40
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 000016A6 0003 func1                                                                     STAT@=02405010
                          00000001 02405010 00000000 00000000 >NOSTACK >>STACK@ 02001D58          AAF493F5 38351E43
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 0000178C 0003 --Return-   from   func1                                                  STAT@=02405010
                          0000000A                            >NOSTACK >>STACK@ 02001E18          AAF493F5 AECA5F41
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 00001BF8 0003 functio3                                                                  STAT@=02405010
                          00000000 00000005 00000000 00000000 >NOSTACK >>STACK@ 02001D58          AAF493F6 2A26DC41
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 000016A6 0004 func1                                                                     STAT@=02405010
                          00000001 02405010 10F0F04B 02001DA0 >NOSTACK >>STACK@ 02001E18          AAF493F6 BD5B9E42
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 0000178C 0004 --Return-   from   func1                                                  STAT@=02405010
                          0000000A                            >NOSTACK >>STACK@ 02001ED0          AAF493F7 3496CA42
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
QPM1 000019CE 0004 func2                                                                     STAT@=02405010
                          00000002 02405010 10F0F04B 02001DA0 >NOSTACK >>STACK@ 02001E18          AAF493F7 AE3CE843
                                                              >>STATIC 000B0000 00000001 00000000
                                                              00000005 A385A2A3 89958740 D8D7D4F1
                                                              C14B4B4B 150086A4 95838693 9681A3F6
                                                              409985A3 A49995A2 7E6C864B 4B4B1500
                                                              00000000 D8D7D2F3
*QPM1 00001A6C 0004 --Return-   from   func2                                                 STAT@=02405010
                          00000014                            >NOSTACK >>STACK@ 02001ED0          AAF493F8 27F6DC40
                                                              >>STATIC 000B0000 00000001 00000000
```

Figure 54. Trace Output with Static and without Stack Data (Part 2 of 2)

Trace Output with Other Breakpoint Entries

Figure 55 on page 150 shows the trace output with other breakpoint entries.

```
*ISOC C FUNCTION TRACE ENTRIES
     TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
 USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

  (A)     (B)    (C)   (D)                (E)                          (F)             (G)
*LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS          **STACK  **STATIC *TIME STAMP
 QZZ0 000000CE 0001 qzz0
                               D8E9E9F0 D8E9E9F1 D8E9E9F1 D8E9E9F2 02A01D78 02E02018 ADBD6BDE F3441842
 QZZ0 0000011C 0001 HOOK..CALLBGN  by  qzz0
                               02E021A0 00000001 00401394 00000002 02A01D78 02E02018 ADBD6BDE F3490B42
 ???? 815BA3AE 0002 printf
                               02E021A0 00000001 00401394 00000002 02A01E38 02E02018 ADBD6BDE F34EBF42
 ???? 815BA3EE 0002 HOOK..CALLBGN  by  printf
                               02A01E60 02E021A0 02A01E50 00000000 02A01E38 02E02018 ADBD6BDE F3545142
 ???? 815BA3F4 0002 HOOK..CALLRET from vsprintf
                               0000002B                            02A01E38 02E02018 ADBD6BDE F3653F42
 ???? 815BA424 0002 HOOK..CALLBGN  by  printf
                               02A01E60 02E021A0 02A01E50 00000000 02A01E38 02E02018 ADBD6BDE F36A6742
 ???? 815BA42A 0002 HOOK..CALLRET from puts
                               0000002C                            02A01E38 02E02018 ADBD6BDE FC5F4442
 ???? 815BA44E 0002 HOOK..POST         printf
                                                                   02A01E38 02E02018 ADBD6BDE FC62EB42
 ???? 815BA456 0002 HOOK..GOTO         printf
                                                                   02A01E38 02E02018 ADBD6BDE FC64A642
 ???? 815BA464 0002 --Return-   from   printf
                               0000002B                            02A01E38 02E02018 ADBD6BDE FC666342
 QZZ0 00000122 0001 HOOK..CALLRET from printf
                               0000002B                            02A01D78 02E02018 ADBD6BDE FC689542
 QZZ0 00000168 0001 HOOK..DO           qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FC73B342
 QZZ0 00000168 0001 HOOK..DO           qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FC77B842
 QZZ0 00000198 0001 HOOK..POST         qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCA83D42
 QZZ0 000001AE 0001 HOOK..ENTRY  BLOCK qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCAA6842
 QZZ0 000001C2 0001 HOOK..EXIT  BLOCK  qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCAE5B42
 QZZ0 000001AE 0001 HOOK..ENTRY  BLOCK qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCB08A42
 QZZ0 000001C2 0001 HOOK..EXIT  BLOCK  qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCB48442
*QZZ0 000001E8 0001 HOOK..POST         qzz0
                                                                   02A01D78 02E02018 ADBD6BDE FCE94542
```

Figure 55. Trace Output with Other Breakpoint Entries

Question Marks (????????) in the CALLING/RETURN PARAMETERS Field

When the calling or return parameters are not addressable, as determined by the load real address (LRA) instruction, they are indicated by *????????*. C function trace cannot determine the parameters, if any.

Figure 56 on page 151 shows the calling or return parameters that are not addressable.

```
*ISOC C FUNCTION TRACE ENTRIES
    TABLE ADDRESS = 02E03000   LENGTH = 00001000 (H)
USER AREA ADDRESS = 02F03000   LENGTH = 00001000 (I)

 (A)     (B)    (C)   (D)                (E)                       (F)                                  (G)
*LODM **OFFSET **LV *FUNCTION *CALLING/RETURN PARAMETERS          **STACK  **STATIC                  *TIME STAMP
 QPM0 00000494 0001 QPM0                                                                  STK@=02001B80 STAT@=02403010 (J)
                            00000007 00000077 00328368 00000006 >>STACK> 00000000 00000000 00000000 AAE1723F 8391D442
                                                                000007FC FC0032A3 A3900014 14001000
                                                                00010000 00000000 00020032 32A39000
                                                                0032A3A8 A8000000 00000000 00000000
                                                                00000000 00000000
                                                                >>STATIC C2C2C2C2 C2C2C2C2 C2C2C2C2
                                                                C2C2C200 D8D7D5F2 00D8D7D5 F240C5D5
                                                                E3C4C340 86818993 A4998540 60409596
                                                                409985A3 A4999540 A39640D8 D7D4F04B
                                                                40150000 D8D7D2F2
*QPM1 00000494 0002 QPM1                                                                  STK@=02001B80 STAT@=02405010
                            ???????? ???????? ???????? ???????? >>STACK> 00D159A0 00000000 01401E30 AAE1723F F88FC042
                                                                00000007 07018030 300080D1 D15A7400
                                                                00D16D60 60000000 00000030 30100000
                                                                00000004 04D8D7D4 D4F1FFFF FFFF0000
                                                                00000090 9000D1B4
                                                                >>STATIC 00000000 00000005 A385A2A3
                                                                89958740 40D8D7D4 D4F1C14B 4B4B4B15
                                                                150086A4 A4958386 86939681 81A3F640
                                                                409985A3 A3A49995 95A27E6C 6C864B4B
                                                                4B4B1500 00000000
```

Figure 56. Calling or Return Parameters That Are not Addressable

Customizing Trace Information

The C function trace environment can be customized by using user exits.

User Exits

CP user exits are available to give C users flexibility to customize the C function trace environment, initialize storage after the user trace area storage is allocated, and insert additional trace data into the user data area. The C function trace user exits are as follows.

User Exit	Description
CDEB	Trace environment customization. This user exit is called only once for each ECB, when C function trace is entered for the first time. At this user exit, you can customize the C function trace environment. For example, you can use the ENATC macro to activate or deactivate C function trace breakpoints and use the SETTC macro to set C function trace table size, user area size, and additional trace options (such as STACK and STATIC).
CEXP	User trace area initialization. This user exit is called only once for each ECB, when a C function trace is entered for the first time. At this user exit you can initialize the contents of any C function trace user area. You must have previously issued a SETTC macro to specify a C function trace user area size so that storage can be allocated prior to calling this user exit. For example, you can issue the SETTC macro in the CDEB user exit; if activated, the CDEB user exit is called before the CEXP user exit is called.
CTRC	Trace user data. This user exit is used each time a C function trace breakpoint is encountered. At this user exit you can insert additional user trace data for the current trace entry.

The address of the user area is stored in the ICID_UTAA field in the CID. The user area must have been allocated previously by issuing the SETTC macro.

Note: It is your responsibility to handle data formatting of the C function trace user area at dump time. The routine AFECTRU in CFMCC is called to format the user trace area when the breakpoints are traced. This routine simply branches back to the caller. If you have created a C function trace user area and want the C function trace user area output to be contained in a dump, you must provide code for the AFECTRU routine to format and print the C function trace user area.

See the *TPF System Installation Support Reference* for a complete description of these user exits.

C Function Trace Use of the SETOC Macro

ISO-C programs that were compiled using the TEST compiler option of one of the IBM C/370 family of compilers supported by the TPF 4.1 system, and that are traced using the C function trace facility, may time out because of multiple trace hooks in the code.

In the TPF 4.1 system, you can compile trace hooks into C or C++ code to perform statement tracing, path tracing, and other types of tracing. Compiling many modules using C function trace and then tracing all of these hooks may cause a CTL-10 because the application may spend a large amount of time tracing.

Using the SETOC macro, the TPF 4.1 system allows an entry control block (ECB) to increase the amount of time it takes to time-out between loss of control. Using the ZCTKA command and the SETOC macro together, an ECB can avoid application time-out a number of times before finally timing out.

The following is an example of avoiding time-out when using C function trace.

1. Enter **ZCTKA ALTER AVOIDT**-*n* where *n* is a value of 0 through 32766. It is important to specify the minimum value you need because a large value can degrade system performance and cause lockout problems.
2. IPL the TPF 4.1 system to update keypoint A (CTKA).
3. Enter **ZSTRC ALTER CDEBUG XHOOKS** to activate C function trace of all trace hooks.

   C function trace receives control the first time an execute (EX) trace hook instruction is issued.

   C function trace obtains the amount of times to avoid time-out and changes this ECB value using the SETOC macro.

Additional Information:

- See *TPF Operations* for more information about the ZCTKA ALTER and ZSTRC ALTER commands.
- See *TPF System Macros* for more information about the SETOC macro.

TPF MQSeries Local Queue Manager Support

TPF MQSeries Trace and Postprocessing

TPF MQSeries local queue manager support provides a communications trace and a function trace.

Communications Trace

Communications trace provides a detailed trace of the data transferred between a TPF MQSeries channel and a remote MQSeries system. Communications trace provides the following information about each message:

- General information, including:
  - Channel name
  - Channel type
  - Time stamp
  - Message type.
- The unformatted transmission segment header (TSH)
- The unformatted message segment header (MSH)
- The transmission queue header (XQH), including:
  - The destination remote queue manager name
  - The remote queue name
  - The message queuing message descriptor (MQMD) associated with the message.
- The message data, which can be up to 80 bytes long.

Function Trace

Function trace allows you to trace the entry to or exit from any function in the MQSeries system. You can set tracing for:

- One or more queues
- One or more channels
- A combination of queues and channels
- All functions in the MQSeries system
- ZMQSC commands.

Function trace provides the following information:

- The name of the function on entry
- The return code on exit
- Additional information, as appropriate, for the particular function.

You can direct the trace output to an RTL tape, the console, both tape and console, or neither tape nor console. The trace data is also attached to the entry control block (ECB) and will be included in any system error dumps that are issued.

Note: Use the console for trace output only on test systems; sending trace output to the console can degrade system performance on a production system.

Using TPF MQSeries Trace

TPF MQSeries tracing is controlled with the ZMQSC TRACE command. Use the following procedure:

1. Enter the ZMQSC TRACE command to set the trace parameters that you want for your MQSeries system. See *TPF Operations* for more information about the specific parameters to specify.
2. Enter **ZMQSC TRACE START** to start tracing according to the parameters you set in step 1. Tracing will continue until you stop tracing even if the channel is stopped and started, or if the queue manager is stopped and started.
3. Enter **ZMQSC TRACE STOP** to stop tracing.
4. If you directed the output to tape, see "Postprocessing the Trace Data" for information about how to postprocess the trace data.

Enter the ZMQSC TRACE command with the DISPLAY parameter specified whenever you want to display the current trace status. See *TPF Operations* for more information about the ZMQSC TRACE command.

All tracing parameters and start and stop status are maintained through an IPL.

Postprocessing the Trace Data

If you direct the trace data to an RTL tape, you must postprocess the data offline. The following is an example of the job control language (JCL) that you can use to generate the TPF MQSeries trace report.

```
//MQTR     EXEC PGRM=CMQMPP,PARM='parm'
//STEPLIB  DD DISP=SHR,DSN=linklibname
//PRINT    DD SYSOUT=A,DCB=(LRECL=133,BLKSIZE=3990,RECFM=FBA)
//RTL      DD DSN=RTL,UNIT=TAPE,DISP=OLD,VOL=SER=tapenum,
//            DCB=(LRECL=4184,BLKSIZE=32760,RECFM=VB)
//* RECFM=VB FOR TAPES CREATED IN BLOCKED FORMAT
/*
```

Where:

parm
is one of the following:

C processes the communications trace data.

F processes the function trace data.

If you do not specify C or F, both communications trace data and function trace data are processed.

linklibname
is your link library name.

tapenum
is the tape number.

Communications Trace Example

Consider the following command entries:

```
ZMQSC TRACE CHL-TPF.MVS.MQD2.TPFMGR ON
ZMQSC TRACE TAPE-ON
ZMQSC TRACE START
```

The following is an example of the communications trace data.

```
MQ COMMS TRACE BEGIN....
-----------------------------------------------------
CHL = TPF.MVS.MQD2.TPFMGR   CHLTYPE = RCVR
1998-09-04      12.35.35
Message Type For This message is INITIAL_DATA.
Data Contains:
E3E2C840 00000084 01010200 00000000 00000000 00000311 01F40000 C9C44040
03070038 0000000A 00007FFE 00002800 3B9AC9FF E3D7C64B D4E5E24B D4D8C4F2
4BE3D7C6 D4C7D940 040001F4 D4D8C4F2 40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040 40404040 40404040 40404040 0000012C
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
-----------------------------------------------------
CHL = TPF.MVS.MQD2.TPFMGR   CHLTYPE = RCVR
1998-09-04      12.35.35
Message Type For This message is INITIAL_DATA.
Data Contains:
E3E2C840 0000004C 01010100 00000000 00000000 00000311 00250000 C9C44040
03070000 0000000A 00007FFE 00002800 3B9AC9FF E3D7C64B D4E5E24B D4D8C4F2
4BE3D7C6 D4C7D940 00000025 D4D8C4F2 40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040 40404040 40404040 40404040 0000012C
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
```

Figure 57. TPF MQSeries Communications Trace Output Example (Part 1 of 2)

```
-------------------------------------------------------
CHL = TPF.MVS.MQD2.TPFMGR   CHLTYPE = RCVR
1998-09-04      12.35.35
Message Type For This message is STATUS_DATA.
Data Contains:
E3E2C840 0000001C 01050000 00000000 00000000 00000222 01F40000 C9C44040
03070000 0000000A 00007FFE 00002800 3B9AC9FF E3D7C64B D4E5E24B D4D8C4F2
4BE3D7C6 D4C7D940 000001F4 D4D8C4F2 40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040
-------------------------------------------------------
CHL = TPF.MVS.MQD2.TPFMGR   CHLTYPE = RCVR
1998-09-04      12.35.35
Message Type For This message is MESSAGE_DATA.
TSH:
E3E2C840 00001564 01043000 B1013AA8 E081FE03 00000311 00250000
MSH:
D4E2C840 00000BB9 00001388 00000000 00001534
Only one message segment in this message.
MQXQH:
E7D8C840 00000001 D6D9C1D5 C7C54BD8 E4C5E4C5 40404040 40404040 40404040
40404040 40404040 40404040 40404040 40404040 40404040 E3D7C6D4 C7D94040
40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
40404040 40404040 D4C44040 00000001 00000000 00000008 FFFFFFFF 00000000
00000311 00000025 40404040 40404040 00000000 00000001 C3E2D840 D4D8C4F2
40404040 40404040 B10077C5 BD792403 00000000 00000000 00000000 00000000
00000000 00000000 00000000 40404040 40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040 40404040 40404040 40404040 D4D8C4F2
40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
40404040 40404040 40404040 E6E7C9C1 D6404040 40404040 1AF1F0F1 F3C4C2C4
C36BF3F9 E4F0F66B D5D6D5E8 6BC2D6E7 60F5F500 00000000 40404040 40404040
40404040 40404040 40404040 40404040 40404040 40404040 00000002 7BD7E4E3
D8404040 40404040 40404040 40404040 40404040 40404040 F1F9F9F8 F0F9F0F3
F2F2F0F4 F4F1F0F2 40404040
Formatted MQXQH Data Contains:
   RemoteQName is: ORANGE.QUEUE
   RemoteQMgrName is: TPFMGR
   Name of reply-to queue is:
   Name of reply queue manager is: MQD2
   Application that puts the message is: #PUTQ
Application Data contains:
E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7
E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7
E7E7E7E7 E7E7E7E7 E7E7E7E7 E7E7E7E7
-------------------------------------------------------
TOTAL NUMBER OF TRACE MESSAGES ARE:  4.
MQ TRACE COMPLETED.
*******************************************************
```

Figure 57. TPF MQSeries Communications Trace Output Example (Part 2 of 2)

Function Trace Example

Consider the following command entries:

```
ZMQSC TRACE Q-PETE ON
ZMQSC TRACE TAPE-ON
ZMQSC TRACE START
```

The following is an example of function trace data.

```
BEGIN TRACE POSTPROCESS
MQ FUNCTION TRACE BEGIN
ECB SVA: 19BE000
TIME: Tue Jan 19 13:23:34 1999
entry CMQAPI-zstMQPUT1: PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-HASH_findEntry: QFNCTRCE
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-HASH_findEntry(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQAPI-zstMQOPEN: PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQQUE-resolveName: /PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQQUE-locateQEntry: PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-HASH_findEntry: MQCOMMON
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-HASH_findEntry(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQQUE-locateQEntry(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQQUE-resolveName(0): PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQAPI-updateDCcount
TIME: Tue Jan 19 13:23:34 1999
return CMQAPI-updateDCcount(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQAPI-zstMQOPEN(0): server
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-HASH_findEntry: QFNCTRCE
TIME: Tue Jan 19 13:23:34 1999
```

Figure 58. TPF MQSeries Function Trace Output Example (Part 1 of 4)

```
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-HASH_findEntry(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQAPI-zstMQPUT: PETE
TIME: Tue Jan 19 13:23:34 1999
entry CMQQUE-locateQEntry:
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-HASH_findEntry: MQCOMMON
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-HASH_findEntry(11)
TIME: Tue Jan 19 13:23:34 1999
entry CMQMGR-QM_storeOneQ
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_createTO2Env
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_createTO2Env(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_createQNameListCursor
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-createCursorForObject
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_getDSdictEntry: QUEUE NAME LIST
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_getDSdictEntry(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-createCursorForObject(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_createQNameListCursor(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_deleteTO2Env
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_deleteTO2Env(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQMGR-QM_storeOneQ(35)
TIME: Tue Jan 19 13:23:34 1999
```

Figure 58. TPF MQSeries Function Trace Output Example (Part 2 of 4)

```
entry CMQHSH-HASH_findEntry: MQCOMMON
TIME: Tue Jan 19 13:23:34 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:34 1999
return CMQHSH-HASH_findEntry(11)
TIME: Tue Jan 19 13:23:34 1999
return CMQQUE-locateQEntry(35)
TIME: Tue Jan 19 13:23:34 1999
entry CMQQUE-addMsgToQ
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_createTO2Env
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_createTO2Env(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQUTL-UT_commitScope: 003
TIME: Tue Jan 19 13:23:34 1999
return CMQUTL-UT_commitScope(0)
TIME: Tue Jan 19 13:23:34 1999
entry CMQQUE-putToBLOB
TIME: Tue Jan 19 13:23:35 1999
return CMQQUE-putToBLOB(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQUTL-UT_commitScope: 004
TIME: Tue Jan 19 13:23:35 1999
return CMQUTL-UT_commitScope(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQQUE-addQEntry
TIME: Tue Jan 19 13:23:35 1999
entry CMQQUE-lockQ
TIME: Tue Jan 19 13:23:35 1999
return CMQQUE-lockQ(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQQUE-locateQEntry: DEAD.LETTER.QUEUE
TIME: Tue Jan 19 13:23:35 1999
entry CMQHSH-HASH_findEntry: MQCOMMON
TIME: Tue Jan 19 13:23:35 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:35 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQHSH-HASH_findEntry(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQQUE-locateQEntry(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQQUE-unlockQ
TIME: Tue Jan 19 13:23:35 1999
```

Figure 58. TPF MQSeries Function Trace Output Example (Part 3 of 4)

```
return CMQQUE-unlockQ(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQQUE-addQEntry(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQUTL-UT_deleteTO2Env
TIME: Tue Jan 19 13:23:35 1999
return CMQUTL-UT_deleteTO2Env(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQQUE-addMsgToQ(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQAPI-zstMQPUT(0): server
TIME: Tue Jan 19 13:23:35 1999
entry CMQHSH-HASH_findEntry: QFNCTRCE
TIME: Tue Jan 19 13:23:35 1999
entry CMQHSH-hashName
TIME: Tue Jan 19 13:23:35 1999
return CMQHSH-hashName(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQHSH-HASH_findEntry(0)
TIME: Tue Jan 19 13:23:35 1999
entry CMQAPI-zstMQCLOSE: PETE
TIME: Tue Jan 19 13:23:35 1999
entry CMQAPI-updateDCcount
TIME: Tue Jan 19 13:23:35 1999
return CMQAPI-updateDCcount(0)
TIME: Tue Jan 19 13:23:35 1999
return CMQAPI-zstMQCLOSE(0): server
TIME: Tue Jan 19 13:23:35 1999
return CMQAPI-zstMQPUT1(0): server
TOTAL TRACE MESSAGES: 78
MQ FUNCTION TRACE COMPLETED
```

Figure 58. TPF MQSeries Function Trace Output Example (Part 4 of 4)

Index

A

activating loadsets for C function trace
  tracing
    ECB origin 136
    production environment 136
  using E-type loader 136
activation of test 13
ADD 29
additional blocks mapped 100
ADDST 29
alpha pointers
  in optional log function 56
  in selective file dump 54
  in selective file trace 55
  in SFD 54
  in SFT 55
  parameter descriptions for C function trace 139
altering records 23
AMSG records 23
area of program error 74
areas referenced by general registers 75

B

basic subsystem 37
BMP0 tables 66
branch trace table 92
breakpoints
  customizing trace information for C function trace 137
BSTA06 30

C

C function trace
  C function trace table 137
  CID field 137
  controlling traces 134
  customizing trace information 151
  description of 133
  description of output 138
  environment 133
  extended output 138
  format of output 138
  starting 133
  stopping 133
  summary of commands 134
  summary of macro options 134
  system error processing 136
  user area 135
  using the SETOC macro 152
C function trace environment
  customizing 151
  elements of 133
C function trace output
  description of 138
  extended output 138
C function trace output *(continued)*
  format of output 138
  samples of trace output
    overview 138
    with other breakpoint entries 149
    with stack and static data 142
    with stack and without static data 145
    with static and without stack data 147
    without stack and static data 138
  standard output 138
  types of 138
C function trace table
  altering the size 135
  description of 137
C function trace user area
  defining a size 135
  formatting 137
  printing 137
C program tracing
  disabling 135
  enabling 135
  SETTC macro 135
C/C++ language register environment 76
Call register assumptions 76
CCP
  See communication control program
CDEB user exit
  for C function trace 135, 136, 151
CDMP dump option code 10
CEXP user exit
  for C function trace 136, 151
CID field
  description of for C function trace 137
collated I/O trace 88
collated I/O trace table 63
collated macro trace table 81
command
  summary of for C function trace 134
  ZCNTM 48
  ZNETW 119
  ZOLDR ACTIVATE 136
  ZSELD 53
  ZSTOP 48
  ZSTRC 133, 134
  ZTHLN 64
  ZTHLT 53, 66
  ZTRAC 48, 49, 50
  ZTRCE 53, 66
communication control program
  trace routine 64
communication control program trace
  output on log tape 61
communications trace
  for TPF MQSeries 153
continuation indicator 31
control blocks
  PTV 14
control record information 3

controlling ECBs for C function trace
ENATC macro 135
SETTC macro 135
using the ENATC macro 136
using the SETTC macro 135, 152
using user exits 136
controlling traces for C function trace
activating loadsets
E-type loader 136
compile time 134
controlling ECBs 135
overview of 134
selecting programs 134
setting trace options 134
system wide trace options 134
CP user exits 151
creating records 23
CTRC user exit
for C function trace 136, 151
customizing trace environment
CDEB user exit 151
CDEB user exit for C function trace 136
customizing trace information for C function trace
breakpoints 137
overview 151
using user exits 151

D

data generation
continuation character 26
data operation code 26
description of 32
detail records 25
format 25
error messages 44
examples 34
examples for input messages 36
examples for mixed SDMF, DRIL, manual 36
examples using DRIL 35
field location 25
field set operands 26
manual generation 34
operations 27, 28, 29
overview 23
STC prefix 25
data generation routine
definition of 23
data operation code 26
DATA record 23, 24
data record information library
definition of 17
description of 31
example of a record 19
record layout 18
records 27
data record prefix entries 29
database restoration tape 12
database restore tape 12
database security 4
DATD dump option code 10
DBR 12
delimiter for STC 26
device addresses
for terminal simulation options
LEID 9
LNIATA 8
LNIATA(CD) 9
overview 8
device types
for terminal simulation options 8
DGR
definition of 23
DGR0 44
diagnostic output formatter
selective file trace output 66
SFDT 54
Diagnostic output formatter
dump headings and labels 66
Diagnostic Output Formatter
collated macro trace table 62
description of 2, 48, 61
macro table 62
online 61
path information unit (PIU) trace table 63
program test vehicle output 64
real-time trace output 63
screen reset indicator 66
selective file dump output 66
terminal simulation output 65
disk allocation records
overview 21
disk type records
overview 21
DOF
description of 2
output formatting 14
SFDT 54
DRIL
definition of 17
example of a record 19
record layout 18
records 27
dump
description of 66
selective file 53
dump cover page 67
dump headings and labels 66
dump label index 69
dump option
record 22
dump option codes
an overview 10
CDMP 10
DATD 10
EBDW 10
ECBD 10
dump options
program test vehicle 65
dump options (DUMP)
for PTV control records 9
dump record information 10

dumps of ECB working storage in the SVM 103

E

E-type loader
use for controlling traces for C function trace 136
EBDW dump option code 10
ECB origin for C function trace
tracing 136
ECB virtual memory (EVM) 93
ECB working storage in SVM 103
ECBD dump option code 10
ENATC system macro
controlling traces for C function trace 135, 136
turning on breakpoints 137
turning on CDEBUG parameter 133
ENT 27
ENTAC macro
starting C function trace 133
stopping C function trace 133
Enter register assumptions 76
ENTIT 27
error messages
data generation 44
for STC 44
linkage editor 44
program test vehicle (PTV) 15
standard data message update 44
STC control program 44
STC I/O program 44
STC loader 44
error recovery for PTV 13

F

field set operands
for STC 26
file address
imbedded 30
file address logging 53
file module 6
files
PTVA 14
formatting and dumping C function trace entries
C function trace table 137
overview 137
user area 137
function trace
for TPF MQSeries 153

G

GEND record 25
description of 24
group record 31
GRP15 macro group identifier 9
GSTAR record
blank 25
description of 24
overview 24
positioning 25
GSTAR record *(continued)*
record type 25

I

I-stream status display 70
ICDF (in-core dump formatter)
See terminal simulation, online
identification number of pilot system 6
identifier, macro 10
IEBUPDTE
DRIL 17
imbedded file address 30
in-core dump formatter (ICDF)
See terminal simulation, online
indicator bytes, tracing 47
initializing user trace area
CEXP user exit 136, 151
input
for running PTV
input messages 11
online disk packs 6
overview 5
pilot tape 6
PTV control records 7
required input sequence 11
sample JCL for test unit tape (TUT) 11
tape requirements 12
test unit tape (TUT) 5
input messages
input for running PTV 11
MSG record 11
STC data generation facility 11
input sequence
input for running PTV 11
required 43
instruction fetching events
tracing 62
ISO-C program 133, 134, 135, 136

J

JCL
sample for running PTV 11
samples
for phase 3 (package test) mode 11
for system test vehicle (STV) mode 11
job control language (JCL)
sample for running PTV 11

K

keyboard lock
CLEAR key 66
RESET option 66
keypoint record for PTV 14
kinds of test activities 3

L

LEDT 44
LEID
  device address 9
link map, C load modules 110
linkage editor
  error message 44
Live Test 14
LNIATA
  device address 8
LNIATA(CD)
  device address 9
loadsets
  E-type loader 136
  use for controlling traces for C function trace 136
logging function
  for selective file trace 66

M

macro
  ENATC 133, 137
  ENATC macro 135
  ENTAC 133
  SETTC 135
  SETTC macro 135
macro group identifier 10
macro options
  summary of for C function trace 134
macro table 62
main storage dump
  headings 112
  labels 112
main storage dump headings and labels 66
message stream
  PTV 15
MSG record 23, 24

N

nesting levels 142
numbers for macro groups 10

O

online disk packs
  input for running PTV 6
operational system
  definition of 3
option codes for dumps
  an overview 10
  CDMP 10
  DATD 10
  EBDW 10
  ECBD 10
optional log function
  alpha pointers description 56
ORD record
  format of 30
output
  system test compiler 44
output formatting
  by DOF 14
overview
  PTV 16
  STC 45

P

package test
  RUNID 7
  selective file dump 53
Package Test 13
path information unit (PIU) trace table
  Diagnostic Output Formatter 63
PDG 12
PER
  events traced
    instruction fetching 62
    storage alteration events 62
    successful branching 62
  user exit 62
  ZSPER command
    start PER tracing 62
PERCC
  disable PER tracing 62
  enable PER tracing 62
phase 3 (package test)
  required input sequence 11
phase 3 (package test) mode
  input for running PTV 5
pilot system
  creating 43
pilot system identification number 6
pilot tape
  input for running PTV 6
  overview 3
pilot tapes
  basic subsystem 37
  creation 37
  description of 17
  generation of 37
  global storage load mode 37
  loading data records 17
  pilot system identification number 37
  RUNID 37
  tape requirements 12
PIU trace table
  Diagnostic Output Formatter 63
pool directory generation tape 12
postprocessor
  See PARM
prefix page 91
problems associated with testing 1
procedures for test activation 13
processing overview
  program test vehicle (PTV) 16
  real-time trace 51
  STC 45

production environment for C function trace
  tracing 136
program security 4
program segment name 10
program test vehicle
  dump options 65
  offline 61
  output on log tape 61
Program Test Vehicle
  description of 3
program test vehicle (PTV)
  control blocks 14
  control of SFDT 53
  description of 1, 17
  message stream 15
  output
    diagnostic output formatter 64
  overview 16
  pilot tapes 17
  required input sequence
    phase 3 (package test) 11
    system test vehicle (STV) 11
  sample JCL
    for phase 3 (package test) mode 11
    for system test vehicle (STV) mode 11
  sample output 15
program testing 1
PTV
  description of 1, 3
  overview 16
  required input sequence
    phase 3 (package test) 11
    system test vehicle (STV) 11
  running
    overview of input 5
PTV control records
  input for running PTV
    dump options (DUMP) 9
    overview 7
    terminal simulation options 8
    test unit identification record (RUNID) 7
PTV modes
  live 4
  overview 4
  phase 3 (package test) 4, 5, 11
  system test vehicle (STV) 4, 6, 11
PTVA 14

R

real-time log tape 12
Real-Time Trace (RTT)
  activation 48
  description of 1, 47
  example of RTT output 48
  input 48
  minimum trace output 48
  output 48
  output on log tape 61
  processing overview 51
record
  AMSG 23
  creating or altering 23
  DATA 23, 24
  disk allocation 21
  disk type 21
  dump option 22
  GEND 24, 25
  group 31
  GSTAR 24
  MSG 23, 24
  SDMF GROUP 23
  SIZ 31
register environment for C/C++ language 76
REP 28
REPST 28
required formats for STC input 21
required input
  SDMU function 43
  sequence 43
  STC programs 44
RTL/RTA 12
RTT processing overview 51
RUNID
  fields 21
  for PTV control records 7
  format 21
  identification of pilot systems 21
  identification of test units 17, 21
  keyboard lock 66
  package test 7
  pilot tapes 37
  Program Test Vehicle 5
  unit test 7
RUNID record
  specifications for
    database restore indicator 7
    global storage load 7
    message input mode 7
    number of messages in system 7
    online terminal use 7
    pilot system identification 7
    running state 7
    screen reset indicator 7
    test unit identifier 7
    user comments 8
    user of pilot tape 7
running PTV
  overview of input 5

S

sample SDMU REGEN run 42
sample SDMU UPDAT run 40
screen reset indicator
  for Diagnostic Output Formatter 66
SDF 12, 17
SDMF
  description of 17, 23
  GROUP records 23

SDMF generation
input for 38
SDMF regeneration 41
SDMU
ALTER 40
CREAT 38
DELET 40
DUMPD 42
DUMPR 42
DUMPT 43
END 39
ENTER 38
generation 38
overview 37
REGEN 41
required input 43
sample REGEN run 42
sample UPDAT run 40
UPDAT 39
security, program and database 4
segment name, program 10
selective file dump
alpha pointers description 54
description of 53
Diagnostic Output Formatter 66
output 54, 66
output on log tape 61
selective file dump and trace
description of 53
system test compiler 53
Selective File Dump and Trace
description of 1
Selective File Dump and Trace (SFDT)
description of 53
selective file trace
alpha pointers description 55
diagnostic output formatter 66
file address logging 53
logging function 66
output 54, 66
output on log tape 61
processing overview 56
trace period 53
SETOC macro
relationship to C function trace 152
SETTC macro
altering size of a C function trace user area 135
altering size of the C function trace table 135
controlling traces for C function trace 135, 152
disable C program tracing 135
enable C program tracing 135
setting trace options for C function trace
using the ZSTRC command 134
SFD
alpha pointers description 54
description of 53
SFDT
description of 1, 53
system test compiler 53
SFT
alpha pointers description 55
SFT *(continued)*
file address logging 53
processing overview 56
trace period 53
SIZ record
record length 31
standard data message update
error messages 44
generation 38
overview 37
standard data message update (SDMU)
description of 21
standard data/message file (SDMF) 17, 23, 29, 31
status of PTV objects 14
STC
description of 17
STC control program
error messages 44
STC data generation facility
for input messages 11
STC I/O program
error messages 44
STC overview 45
STC prefix 25, 29
STC processing
overview 45
STCC 44
STCI 44
STCL 44
STLDR 44
storage alteration events
tracing 62
storage dump
description of 66
STPP
See PARM
STTS (system test terminal simulator)
See system test terminal simulator (STTS) program
SUB 28
SUBST 29
subsystem user
for pilot tapes 37
macro table 62
SYCON 6
system allocator 6
system dump 137
system error dump headings 66
system error processing
for C function trace
criteria for formatting and dumping ECBs 137
overview 136
performance considerations 137
selecting ECBs for formatting and dumping 137
system loader
disk pack creation 6
system storage 102
system test
selective file dump 53
System Test 14
system test compiler
activation 53

system test compiler *(continued)*
description of 17
file input 44
output 44
with SFDT 53
System Test Compiler
description of 1
system test terminal simulator (STTS) program 61
system test vehicle (STV)
required input sequence 11
system test vehicle (STV) mode
input for running PTV 6
system tests
BMP0 tables 66
keyboard lock
RESET option 66

T

tape requirements
input for running PTV 12
task communication area (TCA) 133
TCA 133
terminal simulation
for phase 3 (package test) 11
online 61
output
diagnostic output formatter processing 65
output on log tape 61
overview 22
records 22
terminal simulation options
device addresses 8
device types 8
for PTV control records 8
test activation procedures 13
test activities 3
test output 14
test unit
creating 43
test unit identification record (RUNID)
for PTV control records 7
test unit tape (TUT) 12
content of 5
input for running PTV 5
Test unit tape preparation 3
test unit tapes 17
testing levels for PTV 4
testing philosophy 3
TPF MQSeries local queue manager support 153
trace support
overview 153
postprocessing data 154
using 154
trace activation
system test compiler 53
trace count output
an example 50
trace information
customizing for C function trace
breakpoints 137
trace information *(continued)*
customizing for C function trace *(continued)*
overview 151
using user exits 151
for TPF MQSeries
controlling 154
postprocessing 154
sample 154, 156
trace nesting levels
interpretation of 142
trace options for C function trace
setting of
ZSTRC command 134
trace output for C function trace
with other breakpoint entries
sample of 149
with stack and static data
sample of 142
with stack and without static data
sample of 145
with static and without stack data
calling/return parameters field 150
sample of 147
without stack and static data
description of alpha pointers 139
sample of 138
trace nesting levels 142
trace period
SFT 53
traces
controlling for C function trace 134
tracing for C function trace
ECB origin 136
production environment 136
tracing indicator bytes 47
tracing user data
CTRC user exit 136, 151
TRCE 53
TUT 3, 12, 17
content of 5
input for running PTV 5

U

unit test
RUNID 7
selective file dump 53
system test compiler 53
user abends
for STC 44
user exits
CDEB 135, 136, 151
CEXP 136, 151
controlling traces for C function trace 136
CTRC 136, 151
customizing trace information for C function trace 151
user trace area
initialization of 136, 151
user-defined records
PTV input 11

Z

ZCNTM command 48
ZNETW command
  network command status table relationship 119
ZOLDR ACTIVATE command
  activating loadsets
    for ISO-C programs 136
  activating loadsets with E-type loader 136
  controlling ECB access
    SEL parameter 136
  ISO-C program
    activating loadsets 136
ZSELD command 53
ZSTOP command 48
ZSTRC command
  setting trace options for C function trace 134
  turning on CDEBUG parameter 133
ZTHLN command 64
ZTHLT command 53
  for logging function 66
ZTRAC command 48, 49, 50
ZTRCC command
  macro table 62
ZTRCE command 53
  for logging function 66
  for selective file trace 66

File Number: S370/30XX-31
Program Number: 5748-T14

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH31-0164-13