

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers ***UNIX*[®] RTR Operating System**

System Maintenance Manual
Volume 1

254-303-106
Issue 1
February 2002



Copyright © 1998 Lucent Technologies. All Rights Reserved.

This information product (IP) is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity, including other Lucent Technologies Business Units or Divisions, without express written consent. For permission to reproduce or distribute, call 1-888-LTINFO6 (1-888-584-6366).

Notice

Lucent Technologies is the successor to the business and assets of AT&T Network Systems Business Unit. Every effort was made to ensure that the information contained in this IP was complete and accurate at the time of printing. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes, and you are cautioned that the hardware, software, features, and capabilities described herein may not be the same as the product you have.

Trademarks

5ESS is a registered trademark of Lucent Technologies.

AUTOPLEX is a registered trademark of Lucent Technologies.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

WE is a registered trademark of Lucent Technologies.

All other trademarks are identified as they appear in the text.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana. Orders for additional copies may be placed from the continental United States by calling 1-888-LUCENT8 (1-888-582-3688) or faxing to 1-317-322-6484. Customers outside the continental United States may place orders by contacting the in-country Lucent Technologies market manager. Lucent Technologies personnel outside the continental United States can order additional copies by faxing to 1-317-322-6699.

Information Product Support Telephone Number

Call 1-888-LTINFO6 (1-888-584-6366) to report errors or to ask questions about this IP.

Acknowledgment

Developed by Lucent Technologies Network Systems Customer Training and Information Products.

Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 1

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiph hotline@lucent.com



Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 1

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiph hotline@lucent.com



Contents

	About This Information Product	xxxi
	■ Purpose	xxxi
	■ Audience	xxxi
	■ Reason for Issue	xxxii
	■ How to Use This Information Product	xxxiii
	■ Conventions Used	xxxvi
	■ Safety Labels	xxxviii
	■ Related Information Products	xxxix
	■ How to Comment on This Information Product	xl
	■ Mandatory Customer Information	xl

1	Hardware Identification	1-1
	■ Introduction	1-1
	■ 3B20D Model 1 Computer Hardware	1-4
	■ 3B20D Model 2 Computer Hardware	1-18
	■ 3B20D Model 3 Computer Hardware	1-36
	■ 3B21D Computer Processor Cabinet	1-52

2	Maintenance Plan	2-1
	■ Introduction	2-1
	■ Site Maintenance Plan	2-2
	■ Automatic Recovery Actions	2-3
	■ Manual Recovery Actions	2-4
	■ Preventive Maintenance	2-4
	■ Responding to Alarms	2-6
	■ Documentation	2-17
	■ Using Automatically Generated Data for Troubleshooting	2-17
	■ Switching Control Center Interface	2-18
	■ Training	2-18
	■ Field Support	2-18

Contents

3	Recovery	3-1
	■ Introduction	3-1
	■ Responding to Automatic Recovery Actions	3-1
	■ Recovering the System Manually	3-8
	■ Using Processor Recovery Messages	3-33
	■ Responding to a Terminal Suspend	3-33
	■ Isolating Emergency Action Page Problems	3-45
	■ Recovery Using Software Backup Disk and Alternate Boot Disk Options	3-51
	■ Recovering from DIOP	3-59
	■ Loading Disk from Tape	3-85

4	Error Analysis	4-1
	■ Introduction	4-1
	■ Description of Error Reports	4-2
	■ Error Report Analysis	4-18
	■ Register Layouts	4-82

5	Routine Exercise Procedures	5-1
	■ Introduction	5-1
	■ Purpose of REX	5-1
	■ REX Operation	5-1
	■ REX Scheduling	5-2
	■ REX Scheduling Recommendations	5-2
	■ REX Manual Commands	5-3
	■ REX Inhibits	5-7
	■ REX Inhibit Recommendations	5-8
	■ REX Inhibit Implementations	5-9
	■ REX Summary	5-12

Contents

6	Diagnostics	6-1
	■ Introduction	6-1
	■ Maintenance Input Request Administrator	6-3
	■ Automatic Diagnostics	6-7
	■ Performing Diagnostics	6-8
	■ Synchronous Data Link Restore Tool	6-54
	■ Locating Pins	6-57
	■ Processor Recovery Messages	6-59

7	Off-Line Boot	7-1
	■ Introduction	7-1
	■ Using OFLBOOT	7-3
	■ Performing OFLBOOT	7-4
	■ Checking OFLBOOT Results	7-17
	■ Examples of Performing OFLBOOT	7-20

8	Log File Review	8-1
	■ Introduction	8-1
	■ Using Log Files	8-3
	■ Core Dump Administration	8-5

9	File-System Maintenance	9-1
	■ Introduction	9-1
	■ Checking File-System Resources	9-1
	■ Manipulating Files	9-2
	■ Using File-System Audits	9-5
	■ File Corruption Detection (SIV Only)	9-9
	■ Boot File Verification Through OFLBOOT	9-10
	■ System Backup	9-11

Contents

■ Copying to Backup Partitions	9-13
■ Making a Backup Disk	9-26
■ Making a Backup Tape Copy of a System Disk	9-34
■ Initialize Disk Partition with Zeros	9-36
■ Copying Files to Nonactive Disks	9-37
■ Comparing and Correcting Disk Data	9-39
■ Procedure for Handling Disk Errors	9-44

10	Maintaining Software Integrity	10-1
	■ Introduction	10-1
	■ Checking Operating System Resources	10-1
	■ Using Operating System Audits	10-4
	■ Manipulating Processes	10-6
	■ Maintenance Guide for ULARP	10-7
	■ Using CRON	10-18
	■ Writing SMD Disk Drive Defect Tables	10-19
	■ Adding to the Defect Table of a SCSI Disk Drive	10-23

11	System Update	11-1
	■ Introduction	11-1
	■ System Update Procedures	11-1
	■ SUPR Description	11-5
	■ System Update Menu Page Description	11-9
	■ System Update Session Examples	11-15
	■ Troubleshooting System Update Procedures	11-54
	■ Input Messages	11-59

Contents

12	Craft Interface	12-1
	■ Introduction	12-1
	■ Maintenance Interface	12-1
	■ Maintenance Procedures	12-4

13	Field Update	13-1
	■ Introduction	13-1
	■ Update Methods and Processes	13-1
	■ Broadcast Warning Messages	13-6
	■ Messages Files Execution and Display	13-22
	■ Field Update Session Example	13-34
	■ Input Messages	13-73
	■ System Generation Database	13-74

14	System Audits	14-1
	■ Introduction	14-1
	■ System Integrity Subsystem	14-3
	■ Procedures for Modifying Audit Records	14-38
	■ Operating-System Audits	14-50

15	Input/Output Processor Magnetic Tape Unit	15-1
	■ Introduction	15-1
	■ Using the Magnetic Tape System	15-3
	■ Operating, Maintaining, and Diagnosing the Magnetic Tape System	15-4
	■ Writing a File from a Disk	15-7

Contents

16	SC Status and SD Point Operation	16-1
	■ Introduction	16-1
	■ SCSD ECD Records	16-10
	■ Capabilities and User Requirements	16-18
	■ SCSD Commands	16-62
	■ Growth and Degrowth Procedures	16-64
	■ Verifying SCSD Operation	16-76

17	Spare Circuit Packs	17-1
	■ Introduction	17-1
	■ General Guidelines	17-1

18	Maintenance Tools	18-1
	■ Introduction	18-1
	■ Field Test Set	18-1
	■ GRASP/EGRASP	18-3
	■ Software Debugging Tools	18-4

19	SCSI Peripheral System	19-1
	■ Introduction	19-1
	■ Firmware vs. Pumpcode	19-2
	■ Verify the Integrity of Pumpcode	19-4
	■ Installing New Pumpcode	19-5
	■ SCSI Disk Drives	19-7
	■ SCSI Tape Drives	19-9

Contents

20	Magnetic Tape Special Device Filenames	20-1
	■ Introduction	20-1
	■ Tape Device <i>logdev</i> Records	20-1
	■ IOP Magnetic Tape <i>logdev</i> Records (3B20D Computer Only)	20-2
	■ SCSI/IOP Backward Compatible <i>logdev</i> Records	20-4
	■ SCSI <i>logdev</i> Records	20-5

21	Software Update	21-1
	■ Introduction	21-1
	■ Software Update Types	21-2
	■ Space Reclamation	21-12
	■ Error Conditions	21-12
	■ Enhanced SU Capabilities	21-14
	■ SU Installation	21-16
	■ SU Maintenance	21-20
	■ Routine Operations and Maintenance Procedures	21-24
	■ Page 1940 Easy SU Installation	21-44
	■ Page 1941 SU Automation Scheduling	21-49
	■ Page 1942 SU Automation Office Profile	21-53
	■ Page 1943 SU Automation Healthcheck	21-56
	■ Page 1950 SU Maintenance	21-58
	■ Page 1960 SU Installation	21-61

22	DSCH Computer Interconnect Feature	22-1
	■ Introduction	22-1
	■ Installation Procedure	22-5
	■ Maintenance Commands for DCI	22-10
	■ SG/ECD Setup	22-12

Contents

23	UUCP Feature	23-1
	■ Introduction	23-1
	■ Scope of the UUCP Feature Chapter	23-2
	■ Supported/Unsupported UUCP Commands	23-3
	■ Feature Activation/Deactivation	23-4
	■ Operation of UUCP	23-10
	■ UUCP Administration	23-18
	■ UUCP Security	23-39
	■ Troubleshooting	23-41
	■ Error Code Interpretation	23-45
	■ UUCP Administrative Files	23-63
	■ UUCP Manual Pages	23-65

A	3B20D Computer References	A-1
	■ Introduction	A-1
	■ Control Unit Diagnostic Test Sequence	A-2
	■ Diagnostic Hierarchy	A-4
	■ Test Procedures	A-6
	■ Supplementary Phases	A-11
	■ Main Store Failure Data	A-23
	■ Processor Recovery Messages	A-29
	■ Diagnostic Information Tables — CU, DFC, and IOP	A-30

B	3B21D Computer References	B-1
	■ Introduction	B-1
	■ Control Unit Diagnostic Test Sequence	B-2
	■ Diagnostic Hierarchy	B-4
	■ Test Procedures	B-6
	■ Supplementary Phases	B-13
	■ Main Store Failure Data	B-26

Contents

- Processor Recovery Messages B-29
- Diagnostic Information Tables — CU, DFC,
IOP, and DCI B-30

Glossary

GL-1

Index

IN-1

Figures

1

Hardware Identification

1-1.	Typical 3B20D Model 1 Computer Functional Block Diagram	1-5
1-2.	3B20D Model 1 Computer Control Unit Frame	1-7
1-3.	3B20D Model 1 Computer Peripheral Control Frame	1-10
1-4.	3B20D Model 1 Computer 300-MB Moving Head Disk Frame	1-13
1-5.	3B20D Model 1 Computer SCSI Disk Cabinet	1-15
1-6.	3B20D Model 1 Computer Power Distribution Frame	1-17
1-7.	Typical 3B20D Model 2 Computer Functional Block Diagram (SMM Release 1)	1-19
1-8.	Typical 3B20D Model 2 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-20
1-9.	Typical 3B20D Model 2 Computer Functional Block Diagram (VLMM Release 6 Retrofit)	1-21
1-10.	3B20D Model 2 Computer Processor Control Frame	1-22
1-11.	3B20D Model 2 Computer 300-MB Moving Head Disk Frame	1-28
1-12.	3B20D Model 2 Computer Tape/Disk Cabinet	1-30
1-13.	3B20D Model 2 Computer SCSI Disk Cabinet	1-32
1-14.	3B20D Model 2 Computer 160-MB Minimodule Disk Frame	1-34
1-15.	3B20D Model 2 Computer Power Distribution Frame	1-35
1-16.	Typical 3B20D Model 3 Computer Functional Block Diagram (SMM Release 1)	1-37

Figures

1-17.	Typical 3B20D Model 3 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-38
1-18.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SMD DFC Unit	1-39
1-19.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SCSI DFC Unit	1-40
1-20.	3B20D Model 3 Computer Processor Cabinet	1-41
1-21.	3B20D Model 3 Computer Tape/Disk Cabinet	1-48
1-22.	3B20D Model 3 Computer SCSI Disk Cabinet	1-50
1-23.	3B20D Model 3 Computer Power Cabinet	1-51
1-24.	3B21D Computer Processor Cabinet	1-53
1-25.	3B21D Computer Growth Cabinet	1-56

2

Maintenance Plan

2-1.	3B20D Computer TN2, TN3, TN5, and TN6 Power Switches Faceplate	2-10
2-2.	3B21D Computer UN373 (CU Power Switch) Faceplate	2-12
2-3.	3B21D Computer UN375 (CU Power Switch) Faceplate	2-13
2-4.	3B21D Computer UN376 (CU Power Switch) Faceplate	2-14

Figures

3

Recovery

3-1.	Recovery Procedure (System Configuration)	3-30
3-2.	Recovery Procedure (Application Failure)	3-31
3-3.	Recovery Procedure (Post Recovery)	3-32
3-4.	MTTY/3B20D Computer Interface (Model 1)	3-47
3-5.	MTTY/3B20D Computer Interface (Model 2 or Model 3)	3-48
3-6.	MTTY/3B21D Computer Interface with a TN983 MTTYC	3-49
3-7.	MTTY/3B21D Computer Interface with a UN583 MTTYC	3-50
3-8.	Poke 120 — DFC Status Page	3-62

4

Error Analysis

4-1.	3B20D Computer Postmortem Dump (Software Requested Initialization) - Releases Prior to Release 21.27	4-59
4-2.	3B21D Computer Postmortem Dump (Software Requested Initialization) - Releases Prior to Release 21.27	4-60
4-3.	3B20D Computer Postmortem Dump (Hardware Source Caused by Microinterrupt) - Releases Prior to Release 21.27	4-61
4-4.	3B21D Computer Postmortem Dump (Hardware Source Caused by Microinterrupt) - Releases Prior to Release 21.27	4-62
4-5.	3B20D Computer Postmortem Dump (Manual Request) - Releases Prior to Release 21.27	4-63

Figures

4-6.	3B21D Computer Postmortem Dump (Manual Request) - Releases Prior to Release 21.27	4-64
4-7.	3B20D Computer Postmortem Dump (Manual Request) - Release 21.27 and Later	4-65
4-8.	3B21D Computer Postmortem Dump (Manual Request) - Release 21.27 and Later	4-66

Figures

6

Diagnostics

6-1.	Sample Output from OP:DMQ	6-20
6-2.	Sample Diagnostic Failure	6-26
6-3.	Sample TLP Circuit Pack List	6-30
6-4.	Sample Diagnostic Listing	6-39
6-5.	Sample Output from CU DGN Audit	6-45
6-6.	Sample Output from Non-DGN CU Audit	6-51
6-7.	Backplane Pin Set	6-58

10

Maintaining Software Integrity

10-1.	ULARP ECD Record	10-7
10-2.	Control/Display Update Page	10-17

11

System Update

11-1.	System Update Procedure	11-2
11-2.	System Update Menu Page Screen	11-11
11-3.	Enter Arguments Page 1 Screen	11-13
11-4.	Enter Arguments Page 2 Screen	11-14
11-5.	System Update Menu Page	11-16

12

Craft Interface

12-1.	Maintenance Interface Configuration	12-3
12-2.	MTTY Terminal Critical Indicator Display	12-17
12-3.	Emergency Action Page for TN83	12-20
12-4.	Emergency Action Page for TN983	12-21

Figures

12-5.	Emergency Action Page for UN583 or UN597 without Alternate Boot Disk Feature (pre R21.35)	12-22
12-6.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (pre R21.35)	12-23
12-7.	Emergency Action Page for UN597 without Alternate Boot Disk Feature (R21.35 and later)	12-24
12-8.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (R21.35 and later)	12-25
12-9.	Lock Screen Display for UN597	12-26
12-10.	Page Index for RTR Release 21.3 and Earlier	12-37
12-11.	Page Index for RTR Release 21.5 Through Release 21.15	12-38
12-12.	Page Index for RTR Release 21.17 and Later	12-39
12-13.	Status Summary Area Page	12-43
12-14.	Common Processor Display Page	12-45
12-15.	Common Processor Display Page in DIOP	12-46
12-16.	Common Processor Display Page — MTTY and ROP States	12-47
12-17.	Common Processor Display Page — On-Line	12-48
12-18.	Common Processor Display Page — Off-Line	12-49
12-19.	3B20D Computer Common Processor Display Page with CU FAN Alarms	12-50
12-20.	3B21D Computer Common Processor Display Page with CU FAN Alarm	12-51
12-21.	3B21D Computer Common Processor Display Page with ABD Feature	12-53
12-22.	Control/Display Update Page	12-58
12-23.	Operating System Status Page	12-60
12-24.	CFTMTC Display Page 105	12-65
12-25.	CFTMTC Display Page 105 with Menu 912 Data	12-66

Figures

12-26.	CFTMTC Display Page 106	12-67
12-27.	System Update Menu Page	12-74
12-28.	DCI Status Page	12-76
12-29.	DCI Status Page for RCC	12-77
12-30.	DCI Status Page with DSMLNK	12-78
12-31.	Field Update Page	12-83
12-32.	Typical DFC Index Page	12-89
12-33.	Typical DFC Status Page for a System Equipped with Storage Module Drive (SMD) DFCs 0 and 1	12-91
12-34.	Typical DFC Status Page for a System Equipped with SMD DFCs 2 and 3	12-92
12-35.	Typical DFC Status Page for a System Equipped with SCSI DFCs 0 and 1	12-93
12-36.	Typical DFC Status Page for a System Equipped with SCSI DFCs 2 and 3	12-94
12-37.	Typical DFC Status Page for a System, in Full DIOP, Equipped with SCSI DFCs 0 and 1	12-95
12-38.	Typical DFC Performance Page for a System Equipped with SMD DFCs 0 and 1	12-102
12-39.	Typical DFC Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-103
12-40.	Typical DFSA Index Page	12-106
12-41.	Typical DFSA Status Page for a System Equipped with SMD DFCs 0 and 1	12-108
12-42.	Typical DFSA Status Page for a System Equipped with SMD DFCs 2 and 3	12-109
12-43.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 0 and 1	12-110
12-44.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 2 and 3	12-111

Figures

12-45.	Typical DFSA Status Page for a System, in FULL DIOP, Equipped with SCSI DFCs 0 and 1	12-112
12-46.	Typical DFSA Performance Page for a System Equipped with SMD DFCs 0 and 1	12-118
12-47.	Typical DFSA Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-119

Figures

13

Field Update

13-1.	BWM File Structure	13-8
13-2.	BWM SCANS File	13-10
13-3.	BWM HEADER File	13-11
13-4.	BWM MSGS File (MML Format)	13-14
13-5.	Update Package States	13-18
13-6.	Field Update Page	13-23

14

System Audits

14-1.	Audit-Control Subsystem	14-4

16

SC Status and SD Point Operation

16-1.	Software Structure	16-3
16-2.	Three Levels of Software Interfacing SCSD Hardware	16-4
16-3.	Sample <i>scsdbody</i> Form	16-10
16-4.	3B20D Computer SCSD Point Assignments to Logical Groups Examples	16-12
16-5.	3B21D Computer SCSD Point Assignments to Logical Groups Examples	16-13
16-6.	Structure of the SCSD Option Block in the ECD	16-15
16-7.	UN33 Block Diagram	16-21
16-8.	UN933 Block Diagram	16-22
16-9.	Administrator Message Structure	16-32
16-10.	Error Codes for Administrator and Client Processes	16-34
16-11.	ECD Option Block Record	16-35
16-12.	Power Switch	16-40

Figures

16-13.	3B20D Computer Scan Point Transition Diagram	16-43
16-14.	PSM Data Structure	16-49

17 Spare Circuit Packs

17-1.	Circuit Pack Inventory and Testing Log	17-2

19 SCSI Peripheral System

19-1.	Tape Path Components	19-15
19-2.	Cleaning Points	19-19
19-3.	UN376 (NCR006-3503341), UN376C (NCR006-3300608/Comcode 407545243), and UN376E (KS-24367, L1/Comcode 407771260) SCSI DAT Drives — Front View	19-20
19-4.	UN376 and UN376B Series 1 and 2 Circuit Pack Layout	19-25
19-5.	UN376B Series 3 and Later Circuit Pack Layout	19-26
19-6.	Cassette Write Protect	19-27
19-7.	Tape Load Diagram	19-35

20 Magnetic Tape Special Device Filenames

20-1.	IOP Tape Device Partition Number Descriptor	20-2
20-2.	SCSI/IOP Tape Device Partition Number Descriptor	20-4
20-3.	SCSI Tape Device Partition Number Descriptor	20-6

Figures

21

Software Update

21-1.	Software Update Distribution	21-3
21-2.	Software Update Structure	21-5
21-3.	Example of a PDS Message File	21-6
21-4.	Software Update Page 109	21-15
21-5.	Page 1960 SU Installation Menu	21-17
21-6.	Typical 3B20D/3B21D Computer On-Line SU History	21-20
21-7.	Page 1940 Easy SU Installation Menu	21-45
21-8.	Page 1941 SU Automation Scheduling Menu	21-50
21-9.	Page 1942 SU Automation Office Profile Menu	21-53
21-10.	Page 1943 SU Automation Healthcheck Menu	21-57
21-11.	Page 1950 SU Maintenance Menu	21-58
21-12.	Page 1960 SU Installation Menu	21-62

22

DSCH Computer Interconnect Feature

22-1.	Software Architecture of the DCI Feature	22-3
22-2.	Hardware Architecture of the DCI Feature	22-4
22-3.	Example of DCI ucb Form	22-18
22-4.	Example of HA-OMP DCI ucb Form	22-20
22-5.	Example of HA-OMP Adjunct ucb Form	22-22
22-6.	Example of Normal Operations mdct Form	22-24
22-7.	Example of Diagnostic mdct Form	22-25
22-8.	Example of HA-OMP Adjunct mdct Form	22-26

Figures

22-9.	Example of Diagnostic <i>logdev</i> Form	22-27
22-10.	Example of Non-HA-OMP <i>logdev</i> Form	22-27
22-11.	Example of HA-OMP <i>logdev</i> Form	22-27
22-12.	Example of <i>chopt</i> Form	22-28
22-13.	Example of <i>dciopt</i> Form	22-28
22-14.	Example of <i>device</i> Form	22-29
22-15.	Example of <i>classdef</i> Form	22-30
22-16.	Example of <i>errthld</i> Form	22-31

A 3B20D Computer References

A-1.	BOT and EOT Marker Placement	A-22
A-2.	Sample Memory Failure Histogram	A-25

B 3B21D Computer References

B-1.	Sample Memory Failure Histogram	B-27

Tables

About This Information Product

1. IPs Supporting 3B21D Computer	xxxix

1 Hardware Identification

1-1. 3B20D Model 2 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-25
1-2. 3B20D Model 3 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-44

3 Recovery

3-1. Initialization Levels	3-2

4 Error Analysis

4-1. Heading Determination	4-6
4-2. EAI Parameter Buffer Values	4-11
4-3. Error Interrupt Handler Faults	4-20
4-4. Address Conversion Table (TN14 Memory Array)	4-37
4-5. Address Conversion Table (TN28 Memory Array)	4-38
4-6. Address Conversion Table (TN56 Memory Array)	4-38
4-7. Address Conversion Table (TN2012 Memory Array)	4-39
4-8. Address Conversion Table (3B21D Computer Memory Pack)	4-39
4-9. HSR/Channel Conversion Table	4-44
4-10. Direct Memory Access Information	4-51

Tables

4-11.	Process/Problem Identification	4-67
4-12.	Faulty CU Registers	4-72
4-13.	Memory Error Interrupts	4-75
4-14.	TLP for Nondiagnosable Memory Errors (3B20D Computer)	4-76
4-15.	TLP for Nondiagnosable Memory Errors (3B21D Computer)	4-80
4-16.	Bit and Byte Numbering (All Registers)	4-83
4-17.	Cache Error Register A (0x0030000)	4-84
4-18.	Cache Error Register B (0x0030004)	4-85
4-19.	Cache Error Register C (0x0030008)	4-86
4-20.	Cache Error Register D (0x003000C)	4-87
4-21.	Cache Error Register E (0x0030010)	4-88
4-22.	Cache Tag Trap Register (0x003001C)	4-89
4-23.	Cache Error Register G (Hit Flip-Flop) (0x0030014)	4-90
4-24.	Data Path Command Register	4-91
4-25.	Data Path Error Data Register	4-92
4-26.	Data Path Status Register	4-92
4-27.	DMA Channel Status Register — 3B20D Model 1 Computer	4-93
4-28.	DMAC Status Register — 3B20D Model 1 Computer	4-94
4-29.	DMAC Status Register — 3B20D Model 2 and 3 Computers	4-96
4-30.	DMAC Status Register for 3B21D Computer	4-99
4-31.	DSCH Channel Status Register	4-101
4-32.	Error Register	4-102
4-33.	Error A Indicators (3B20D Computer Only)	4-106
4-34.	Faulty Circuit Packs Indicated by ER 1	4-107
4-35.	Error A Indicators (3B21D Computer Only)	4-108
4-36.	Hardware Status Register	4-122
4-37.	Hardware Status Register 3-Out-of-6 Codes	4-124

Tables

4-38.	BGB Destination Codes	4-124
4-39.	Hardware Status Register Maintenance States	4-125
4-40.	Interrupt Source Register	4-126
4-41.	Main Store Controller Error Register 1 (3B20D Computer)	4-127
4-42.	Main Store Controller Error Register 2 (3B20D Computer)	4-128
4-43.	Main Store Controller Trapped Address Register (3B20D Computer)	4-129
4-44.	3B21D Computer Main Memory Register	4-130
4-45.	Memory Control Bank Address Boundary Register	4-131
4-46.	Memory Control Command Register	4-132
4-47.	Memory Control Error Register	4-133
4-48.	Memory Control Error Address Register	4-133
4-49.	Memory Control Status Register	4-134
4-50.	Memory Control Strobe Shape Register	4-135
4-51.	Microinterrupt Error Register	4-136
4-52.	Microinterrupt Error Register 1	4-137
4-53.	Processor Status Word Register	4-138
4-54.	Pulse Point Register	4-140
4-55.	Store Control Register	4-141
4-56.	System Status Register	4-143

Tables

5

Routine Exercise Procedures

5-1.	Early REX Inhibits	5-10
5-2.	New REX Inhibits	5-11

6

Diagnostics

6-1.	Diagnostic Hierarchy Appendix Index	6-8
6-2.	Suggested Routine Test Procedures	6-11
6-3.	EX Commands	6-13
6-4.	Diagnostic Results	6-27
6-5.	Explanation of Test Skipped Reasons	6-28
6-6.	Meaning of TLP Notes	6-31
6-7.	DIAMON Error Codes	6-43
6-8.	DGN CU Error Codes	6-46
6-9.	Selected System Error Codes	6-48
6-10.	Maintenance Channel Status	6-50
6-11.	DGN Audit IOP, DFC, and DCI Error Codes	6-52

7

Off-Line Boot

7-1.	Options for the EXC:OFLBOOT Input Message	7-6
7-2.	Precheck Override Conditions	7-9
7-3.	Secondary DFC Override Condition	7-9

Tables

9

File-System Maintenance

9-1.	File-System Maintenance Commands (Routine)	9-3
9-2.	File-System Maintenance Commands (Troubleshooting)	9-3
9-3.	File-System Maintenance Commands (Permanent)	9-4

10

Maintaining Software Integrity

10-1.	Operating System Audit Commands	10-4
10-2.	Process Execution and Termination Commands	10-6
10-3.	PRM Function Codes	10-14

12

Craft Interface

12-1.	KS-22921, L1 Maintenance Terminal Configuration	12-5
12-2.	KS-23554, L5 and L6 Maintenance Terminal Configuration	12-6
12-3.	KS-23996, L1 and L5 Maintenance Terminal Configuration	12-8
12-4.	Model 5310 ROP Option Selection	12-12
12-5.	Model 577 ROP Option Selection	12-13
12-6.	Model 602 ROP Option Selection	12-15
12-7.	System Critical Indicators	12-18
12-8.	Emergency Action Page Commands	12-27
12-9.	Emergency Action Page Display Indicators	12-32
12-10.	Emergency Action Page Poke Prompts	12-34

Tables

12-11.	Common Processor Display Page Commands Without ABD Feature	12-52
12-12.	Common Processor Display Page Commands with ABD Feature	12-54
12-13.	Parameters and Options for CFTMTC Commands	12-68
12-14.	Display Page 105 Commands, Parameters, and Options for Menu Numbers	12-71
12-15.	Display Page 106 Commands, Parameters, and Options for Menu Numbers	12-72
12-16.	DCI Status Page Commands	12-79
12-17.	Field Update Scope/Action Combinations	12-85
12-18.	DFC Status Page Commands	12-98
12-19.	DFC Performance Page Menu Commands	12-104
12-20.	Disk File System Access Status Page Menu Commands	12-115
12-21.	DFSA Performance Page Menu Commands	12-120

13

Field Update

13-1.	Allowed Scope/Action Combinations	13-27

14

System Audits

14-1.	Audit Families and Audit Names	14-6
14-2.	Audit Names and Member Numbers	14-8
14-3.	Audit Abort Codes	14-23
14-4.	REPT SIMCHK Error Codes	14-27
14-5.	Optional Hexadecimal Data for REPT SIMCHK Error Conditions	14-31

Tables

14-6.	CUMEM Audit Error Numbers	14-53
14-7.	CUMEM Error Reports	14-53
14-8.	CUSTAT Audit Error Numbers	14-60
14-9.	CUSTAT Error Reports	14-61
14-10.	AUD CUSTAT Error Messages	14-67
14-11.	ECD Audit Error Messages	14-76
14-12.	LLA Audit Error Messages	14-79
14-13.	ECDOWN Audit Error Messages	14-97
14-14.	FMGR Audit Errors	14-104
14-15.	FMGR Audit Error Reports	14-107
14-16.	FSBLK Audit Errors	14-136
14-17.	Special Error Messages	14-137
14-18.	FSLINK Audit Errors	14-147
14-19.	Special Error Messages	14-148
14-20.	FSCMPT Audit Errors	14-161
14-21.	Special Error Messages	14-162
14-22.	MMGR Audit Error Messages	14-171
14-23.	MSGBUF Audit Error Messages	14-193
14-24.	Special Error Messages	14-195
14-25.	PMS Audit Error Messages	14-208
14-26.	PROAD Audit Errors	14-219

15 Input/Output Processor Magnetic Tape Unit

15-1.	Magnetic Tape System Input Commands	15-3
15-2.	MTC Diagnostic Commands	15-6

Tables

16 SC Status and SD Point Operation

16-1.	Phase Alarm Levels	16-31
16-2.	3B20D Computer Logical Group Assignments Examples	16-37
16-3.	3B21D Computer Logical Group Assignments Examples	16-38
16-4.	3B20D Computer Scan Point Transitions and Actions	16-42
16-5.	3B21D Computer Scan Point Transitions and Actions	16-44
16-6.	3B20D Computer Scan Point and Signal Distributor Point Assignments	16-53
16-7.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Large Office	16-54
16-8.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Small Office	16-55

19 SCSI Peripheral System

19-1.	Cleaning Schedules and Equipment	19-13
19-2.	Cleaning Schedule Guidelines	19-16
19-3.	Description of Configuration Option Byte 15 Bit Settings	19-23
19-4.	DAT SCSI Device Identification Switch Settings	19-26
19-5.	KS-23909 9-Track Tape Capacity	19-28
19-6.	DAT Cartridge Tape Capacity	19-29
19-7.	NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT Drive Indicators	19-40
19-8.	NCR006-3503341, L1 3.5-Inch SCSI DAT Drive Indicators	19-41

Tables

20 Magnetic Tape Special Device Filenames

20-1.	IOP 9-Track Operating Parameters	20-2
20-2.	Typical <i>logdev</i> Values	20-7

21 Software Update

21-1.	File Transfer Error Conditions (SCANS/CSCANS Interface)	21-13
21-2.	Predefined SU Automation Healthcheck Conditions	21-56

23 UUCP Feature

23-1.	ED3T076-20 G57 and G58 Signal Connections	23-11
23-2.	UUCP Troubleshooting Corrective Action	23-41
23-3.	Error Codes and Definitions	23-50
23-4.	uucheck Error Messages	23-54
23-5.	uucheck Warning Messages	23-57
23-6.	uucheck Caution Messages	23-60
23-7.	uucheck Internal Error Messages	23-62

A 3B20D Computer References

A-1.	Recommended Frequency of Selected Supplementary Phases	A-11
A-2.	Supplementary Phases for CU	A-12
A-3.	Supplementary Phases for SMD DFC	A-18
A-4.	Supplementary Phases for SCSI DFC	A-19
A-5.	Supplementary Phases for IOP	A-20

Tables

A-6.	Partitioning of Address Spectrum Among Memory Arrays	A-27
A-7.	Application Channel Interface	A-31
A-8.	Central Control	A-32
A-9.	Cache Storage Unit	A-36
A-10.	Enlarged Cache Storage Unit	A-37
A-11.	Disk File Controller	A-38
A-12.	SCSI DFC	A-39
A-13.	Direct Memory Access	A-40
A-14.	Direct Memory Access Channel	A-41
A-15.	Dual Serial Channel	A-43
A-16.	Direct User Interface Controller	A-44
A-17.	Input/Output Processor	A-45
A-18.	Main Store Controller	A-46
A-19.	Default Refresh Data Parity Check Times for MASC Phase 96	A-47
A-20.	Moving Head Disk	A-48
A-21.	Magnetic Tape Controller	A-49
A-22.	Maintenance Teletypewriter Controller	A-49
A-23.	Store Address Translator	A-50
A-24.	Serial Channel	A-50
A-25.	Scan Point and Signal Distributor Controller	A-51
A-26.	Synchronous Data Link Controller	A-51
A-27.	Teletypewriter Controller	A-51
A-28.	Utility Circuit	A-52

B

3B21D Computer References

B-1.	Recommended Frequency of Selected Supplementary Phases	B-14
B-2.	Supplementary Phases for CU	B-15
B-3.	Supplementary Phases for DFC 2	B-21
B-4.	Supplementary Phases for DFC 3	B-22
B-5.	Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape	B-23

Tables

B-6.	Supplementary Phases for IOP 2	B-24
B-7.	Central Control	B-31
B-8.	Main Store Controller	B-34
B-9.	Direct Memory Access	B-36
B-10.	Direct Memory Access Channel	B-37
B-11.	Disk File Controller 2	B-38
B-12.	Disk File Controller 3	B-39
B-13.	SCSI Magnetic Tape	B-40
B-14.	Utility Circuit	B-41
B-15.	Input/Output Processor 2	B-42
B-16.	Dual Serial Channel Computer Interconnect	B-43

About This Information Product

Purpose

The System Maintenance Manual is an information product (IP) that is part of a documentation set used to support the *UNIX*® Real-Time Reliable (RTR) operating system running on Lucent Technologies 3B20D and 3B21D computers. It summarizes the corrective and preventive maintenance procedures used to operate and maintain the 3B20D Model 1, 2, and 3 computers and the 3B21D computer. The procedures range from recovering the system to analyzing the error conditions and performing diagnostics. Software diagnostic capabilities are also discussed.

Other topics are discussed that may not be considered a part of routine maintenance, such as the set up and operation of the maintenance terminal, performing updates, running audits, and the use of peripheral equipment.

Audience

This IP provides information for a variety of audiences. Anyone interested in the *UNIX* RTR operating system, 3B20D/3B21D computers, or a summary of corrective and preventive maintenance will find it useful. The primary audience for this IP includes the following:

- Hardware, software, or training course developers
- Craft or maintenance personnel
- 3B20D/3B21D computer system administrators or documentation users.

Reason for Issue

This IP is being issued to incorporate new information for 3B20D/3B21D computers and to bring the IP into compliance with new Lucent Technologies documentation standards. The major changes are as follows:

- Change the six-digit Select Code number for the *System Maintenance Manual* from 304-046 to the new nine-digit number 254-303-106.
- Add information on alarms and responding to alarms in Chapter 2, and remove references to the UN375C disk pack in Figure 2-3.
- Add circuit pack information for UN582 and UN583 in Chapters 3 and 12.
- Add SOURCE field p and h postmortem descriptions in Chapter 4.
- Add DFC to Tables 6-1, 6-2, and 6-5.
- Add TLP Note 04 for CC, CSU, DMA, and MASC in Table 6-6.
- Add new error codes and meaning to Table 6-11.
- Revise the “Copying to Backup Partitions” procedure in Chapter 9.
- Add new figures associated with the poke commands in Chapter 12.
- Add circuit pack information for UN933 in Chapter 16.
- Add circuit pack information for UN376E in Chapter 19.
- Update Appendixes A and B.

Change bars on the far right-hand side of the text are used to denote changes made to the current issue. Change bars will help you understand at a glance what has changed from the previous issue. The position of deleted text is shown by a single asterisk (*). Not all editorial changes will be marked with change bars. Occasionally, change bars do not work in tables or figures; therefore, those changes will be shown by adding a change bar to only the table or figure title.

Whenever this IP is reissued, the reason(s) for reissue will be listed in this paragraph.

How to Use This Information Product

This IP is composed of independent chapters and reference information (Table of Contents, Appendixes, Glossary, and Index). Specific topics are referenced in the Table of Contents and Index with page number locations. The Glossary defines the terminology used in this IP. This IP is organized in three volumes, as follows:

Volume 1

- Master Table of Contents
- About This Information Product
Provides general information about the IP. Information regarding audience, reason for issue, conventions, safety labels, how to order, and how to comment is included. A description of how the IP is organized and a brief description of the contents of each chapter is also provided.
- Chapter 1 — Hardware Identification
Describes the hardware in the computers and contains illustrations that show the location of specific units, circuit packs, components, and associated frames.
- Chapter 2 — Maintenance Plan
Describes the elements in the plan for maintaining the computers.
- Chapter 3 — Recovery
Provides the following recovery procedures: responding to automatic recovery actions, recovering the system manually, using processor recovery messages, responding to a terminal suspend, isolating emergency action page problems, recovering from a duplex outage, recovering from Disk Independent Operation (DIOP), and loading disk from tape.
- Chapter 4 — Error Analysis
Gives an overview of the Control Unit (CU) error messages and automatic postmortem dumps. It will help the user interpret the different types of error messages generated by the automatic fault recovery features.
- Chapter 5 — Routine Exercise Procedures
Provides a description of the Routine Exercise (REX) processes and procedures. Included are detailed descriptions and examples of the REX craft interface and recommendations for scheduling and inhibiting REX.
- Chapter 6 — Diagnostics
Gives general information for using the diagnostic capabilities of the computers, how and when to perform diagnostics, how to interpret the output, and how to use the information to clear hardware problems.

- **Chapter 7 — Off-Line Boot**
Covers the Off-Line Boot (OFLBOOT) feature for the *UNIX* RTR Operating System Release 21.1 and later. OFLBOOT is used primarily to verify the capability to boot essential files without interrupting the system operation.
- **Chapter 8 — Log File Review**
Gives general information about the history of certain automatic and manual maintenance actions that occur in the system. These histories are located in the */etc/log* directory.
- **Chapter 9 — File-System Maintenance**
Gives procedures for checking file-system resources, manipulating files, using the file-system audits, file-system corruption detection, and backing up and restoring file systems.
- **Chapter 10 — Maintaining Software Integrity**
Gives procedures for checking operating system resources by using operating system audits, the User-Level Automatic Restart Process (ULARP), the Clock Daemon (CRON), and manipulating processes.

Volume 2

- **Master Table of Contents**
- **About This Information Product**
- **Chapter 11 — System Update**
Describes system update concepts and defines procedures for implementing updates. This chapter also provides examples of successful system updates and how to back out and restore an unsuccessful update. It describes other processes and utilities related to performing system updates and lists the system update input messages with a short description of their function.
- **Chapter 12 — Craft Interface**
Provides an overview of the software required to perform the various craft interface functions. The information presented summarizes operation of the Maintenance Teletypewriter (MTTY) terminal.
- **Chapter 13 — Field Update**
Describes field update concepts and defines procedures for implementing updates. This chapter also describes Broadcast Warning Messages (BWMs) and gives some rules for applying BWMs. It also describes BWM components and states.

- Chapter 14 — System Audits
Describes the system audits available to the user and the software integrity subsystem (the system feature that monitors the operating system software reliability and stability). This chapter also provides procedural information dealing with system audits, audit output messages and error reports, and system faults.

Volume 3

- Master Table of Contents
- About This Information Product
- Chapter 15 — Input/Output Processor Magnetic Tape Unit
Describes the magnetic tape system components and gives guidelines for its operation, use, and maintenance.
- Chapter 16 — SC Status and SD Point Operation
Provides information on how to write client processes, communicate with the Scan Point and Signal Distributor Handler (SCSDH), physically connect scan points, develop messages for maintenance personnel, grow and degrow UN33/UN933 boards, and grow and degrow the Input/Output Processor (IOP).
- Chapter 17 — Spare Circuit Packs
Gives guidelines for testing and rotating spare circuit packs.
- Chapter 18 — Maintenance Tools
Describes the available maintenance tools for the 3B20D/3B21D computers: the Field Test Set (FTS) and the Generic Access Package (GRASP)/Enhanced Generic Access Package (EGRASP).
- Chapter 19 — SCSI Peripheral System
Provides information which is helpful in maintaining the Small Computer System Interface (SCSI) peripheral system.
- Chapter 20 — Magnetic Tape Special Device Filenames
Describes the magnetic tape special device filenames.
- Chapter 21 — Software Update
Provides a description of program changes that are possible within the database.
- Chapter 22 — DSCH Computer Interconnect Feature
Provides information on the Dual Serial Channel (DSCH) Computer Interconnect (DCI) feature. The feature is implemented in the RTR Release 21.3 and later software. This chapter presents an overview of the DCI feature, configuring the Equipment Configuration Database (ECD), installation procedures for the *Sun** system, and maintenance commands for the DCI board.

* Sun, Sun Microsystems, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

- Appendix A — 3B20D Computer References
Provides 3B20D computer reference information on the following topics: CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures; Supplementary Phases; Main Store Failure Data; Processor Recovery Messages; and the CU, Disk File Controller (DFC), and IOP Diagnostic Information Tables.
- Appendix B — 3B21D Computer References
Provides 3B21D computer reference information on the following topics: CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures; Supplementary Phases; Main Store Failure Data; Processor Recovery Messages; and the CU, DFC, IOP, and DCI Diagnostic Information Tables.
- Glossary
- Index.

Conventions Used

Command, Filename, and Display Notations

The following notations are used to show commands and filenames in the text and displays.

NOTE:

System filenames and command names are case-sensitive, so you must enter them exactly as they are shown.

- Command names in text appear in **bold** type; for example, the **/usr/bin/ls** command. Command names in headings appear in **bold** type.
- Filenames and form names in text appear in *italic* type; for example, the */usr/lib/uucp/System* file or the *scsdbody* form. Filenames and form names in headings appear in ***bold-italic*** type.
- Text that you enter, such as a command or response to a prompt, appears in **bold** type; for example, the **ls -la** command. The convention of making text that you enter **bold** overrides other conventions; for example, enter **scsdbody** for the form name.
- Variables that appear in a command line or file appear in *italic* type; for example, **grep *username* /etc/passwd**. In this example, *username* is a variable indicating a user's name is required.

- Screen displays and system messages appear in constant width type; for example, Please enter your password. Program code listings and file listings are also shown in constant width type. Input messages are shown in **constant-width bold** type.
- Comments and explanations within a display are indented and shown in *italic* type. These are for information only and will not appear on your screen.
- A line in a file or on the computer screen that is too long to be shown as it actually appears in this IP will be shown with a backslash (\) at the end of the first line. This indicates the next line should be read as a continuation of the current line.
- Square brackets around an argument on a command line indicate that the argument is optional; for example, the **lpstat [-t]** command. In this example, the **-t** argument is optional and can be omitted.
- A vertical bar (|) between words in an argument on a command line indicates that one of the arguments is to be selected.
- The key identified on your keyboard as Return, Enter, or a bent arrow (↵) is referred to as the Return key. Occasionally, representations of this Return key will be boxed; for example, Return.
- There is an implied Return at the end of each command and menu response that you enter. Some examples do not explicitly show the Return. Where you may be expected to enter a Return (as in the case where you are accepting a menu default), the symbol <CR> is shown to indicate that you are to press the Return key.
- Key combinations appear in a hyphenated format; for example, Ctrl-d. Press and hold down the first key of a key combination while pressing the second key.
- Ellipses (three dots) on a command line indicate that the previous argument can be repeated; for example, **ls [file ...]**. In this example, multiple files can be listed after the command.
- References to manual pages are followed by their manual page location number in parentheses; for example, **mount(1M)**.

Hexadecimal Notation

Hexadecimal (base 16) numbers are denoted with a **0x** prefix; for example, 0x00A is decimal 10.

Signal Designations

The term “asserted” is used in the descriptions to mean that a signal is driven to its active state. The term “negated” is used in the descriptions to mean that a signal is driven to its inactive state.

Signal names used in the diagrams and descriptions that end in a 0 or 1 indicate the active state of the signal. Names ending in “0” are “active low” signals; names ending in “1” are “active high” signals. For example, CCIOD(31-00)1 describes the Central Control Input/Output Data bits 31 through 00 which are “active high” signals.

Equipment Locations

A coordinate numbering system is used to identify Equipment Locations (EQLs) in units and cabinets. The origin is the lower left front of the cabinet or unit. Vertical increments are measured in inches. Horizontal increments are measured in eighths of an inch. The coordinate location of a circuit pack is expressed as the horizontal and vertical location of the center lines of the connector into which the circuit pack is inserted. The location of a unit in a cabinet is identified by the placement of the lower left corner of the unit in the cabinet.

For example, a connector at EQL 004-080 is located 4 inches above the origin and 10 inches ($80 \times 0.125 = 10.0$) to the right of the origin.

Safety Labels

Safety labels are reminders used to assure the safety of personnel and to minimize service interruptions, loss of data, and damage to equipment, products, and software.

Three types of safety labels are used in Lucent Technologies documentation. The three types, in descending order of priority, are as follows:

1. **DANGER** indicates the presence of a hazard that **will** cause death or severe personal injury if the hazard is not avoided.
2. **WARNING** indicates the presence of a hazard that **can** cause death or severe personal injury if the hazard is not avoided.
3. **CAUTION** indicates the presence of a hazard that **will** or **can** cause minor personal injury or property damage if the hazard is not avoided.

This IP contains safety labels in the form of CAUTION statements.

Related Information Products

Table 1 lists by IP number and Select Code the Lucent Technologies IPs supporting the 3B21D computer. IPs with Select Codes will be converted to nine-digit IP numbers as they are reissued.

Table 1. IPs Supporting 3B21D Computer

NEW IP NUMBER	OLD SELECT CODE	TITLE
254-303-100	-	3B21D Computer Growth/Retrofit Tasks
254-303-101	-	3B21D Computer Routine Maintenance Tasks
254-303-102	-	3B21D Computer Trouble Clearing Tasks
254-303-103	303-007	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Processor Recovery Messages Guide
254-303-104	303-010	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Recent Change and Verify Manual
254-303-105	304-045	3B21D Computer Hardware Reference Manual
254-303-106	304-046	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System System Maintenance Manual
254-303-107	303-072	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Software Troubleshooting Guide
254-303-110	303-080	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Input Messages Manual
254-303-111	303-081	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Output Messages Manual
254-303-112	303-082	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Input Messages Manual
254-303-113	303-083	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Output Messages Manual

IP references will not include full titles; for example, "Refer to 254-303-105, *Hardware Reference Manual*, for more information."

How to Comment on This Information Product

Lucent Technologies welcomes your comments on this IP. Your comments will aid us in improving the quality and usefulness of Lucent Technologies documentation. Please use the Feedback Form provided at the front of this IP. If the Feedback Form is missing, mail your comments to the following address:

Lucent Technologies
Customer Training and Information Products (CTIP)
2400 Reynolda Road
Winston-Salem, NC 27106-4606

Mandatory Customer Information

The 3B21D computer is used as the Administrative Module (AM) in various switching system applications and is not provided as a stand-alone product. Therefore, the application documentation is responsible for providing this information.

Refer to the applicable application documentation for "Mandatory Customer Information."

Hardware Identification

1

Contents

Introduction	1-1
■ 3B20D Computer Hardware Overview	1-1
■ 3B21D Computer Hardware Overview	1-2
■ Circuit Pack Codes	1-2
Suffix Coding of Circuit Packs	1-2
Advantages of Suffix Coding	1-3
3B20D Model 1 Computer Hardware	1-4
■ Control Unit Frame	1-6
Direct Memory Access Input/Output Unit	1-6
Central Control Unit	1-8
Main Store Unit	1-9
Cooling Unit	1-9
Power Units	1-9
■ Peripheral Control Frame	1-10
Input/Output Processor Unit	1-11
Storage Module Drive – Disk File Controller Unit	1-11
■ Moving Head Disk Frame	1-12
Disk Inverter	1-12
Control Panel	1-12
300-MB Disk	1-12
■ Tape Unit Frame	1-14
Magnetic Tape Unit	1-14
Format Control Unit	1-14
■ IOP and SMD-Type DFC Growth Frames	1-14
■ SCSI Disk Cabinet (Optional)	1-14
Power Distribution Unit	1-16
SCSI DFC Units	1-16

Contents

Cooling Unit	1-16
SCSI DUP Units	1-16
■ Power Distribution Frame	1-16
3B20D Model 2 Computer Hardware	1-18
■ Processor Control Frame	1-18
Central Processor Unit	1-23
MAS-IO-DFC Unit	1-24
IOP Basic Unit	1-26
Growth Unit	1-27
■ 300-MB MHD Frame	1-27
Disk Inverter	1-28
Control Panel	1-28
300-MB Disk	1-29
■ Tape Unit Frame	1-29
Magnetic Tape Unit	1-29
300 V A Inverter Synthesizer	1-29
■ Tape/Disk Cabinet (Optional)	1-29
Tape Drive	1-29
340-MB Disk	1-29
Disk Power Supply	1-30
Disk Power Switch	1-30
■ SCSI Disk Cabinet (Optional, Found in Release 6 Only)	1-31
Power Distribution Unit	1-31
SCSI DFC Units	1-31
Cooling Unit	1-31
SCSI DUP Units	1-31
■ 160-MB Minimodule Disk Frame (Optional)	1-33
Moving Head Disk Units	1-33
Disk Inverters	1-33
■ Power Distribution Frame	1-33
3B20D Model 3 Computer Hardware	1-36
■ Two-Bay Processor Cabinet	1-36
Central Processor Unit	1-42
MAS-IO-DFC Unit	1-43
IOP Basic Unit	1-45
Growth Unit	1-46
■ Tape/Disk Cabinet	1-46
Tape Drive	1-46

Contents

340-MB Disk	1-47
Disk Power Supply	1-47
Disk Power Switch	1-47
■ SCSI Disk Cabinet (Optional)	1-49
Tape Drive	1-49
Power Distribution Unit	1-49
SCSI DFC Units	1-49
Cooling Unit	1-49
SCSI DUP Units	1-50
■ Power Cabinet (Optional)	1-51
3B21D Computer Processor Cabinet	1-52
■ Control Unit	1-53
Central Control	1-54
Cache Storage Unit	1-54
Main Store and Main Store Update	1-54
Direct Memory Access	1-54
Expansion Slots	1-55
Utility Circuit	1-55
■ Disk File Controller Unit	1-55
■ Input/Output Processor Unit	1-55
■ Peripheral Growth Cabinet	1-56

Hardware Identification

1

Introduction

The 3B20D (Models 1, 2, and 3) and 3B21D computer systems are dedicated to the *UNIX*® Real-Time Reliable (RTR) operating system. The computers consist of various frames or cabinets that can be arranged to match many application requirements.

The hardware units in the computers consist mainly of TN- and UN-type circuit packs. Each computer system consists of its own disk drives, tape drive, and maintenance-related periphery with growth capabilities.

3B20D Computer Hardware Overview

The 3B20D computer is a high-availability processor that monitors and controls operations in various processing and switching applications. All major functional units are duplicated to ensure uninterrupted and reliable service.

The hardware contains self-checking and error correction circuitry. The software detects faulty processes and equipment, reconfigures or reinitializes the system, and diagnoses and identifies faulty equipment.

The hardware for the 3B20D Model 1, 2, and 3 computers is described in the following sections.

3B21D Computer Hardware Overview

The 3B21D computer is a high-speed, high-reliable, fault-tolerant duplex processor. It is designed to be a bit-compatible version of the 3B20D computer, which is used in various roles in Lucent Technologies switching products.

The architecture of the 3B21D computer is identical to that of the 3B20D computer except for the addition of Expansion (EX) slots in the Control Unit (CU) and support of the Small Computer System Interface (SCSI) tape devices in the 3B21D computer. The hardware is packaged differently than the 3B20D computer.

The advantages of the 3B21D computer are reduced floor space and higher reliability because of a reduction in integrated circuits.

The 3B21D computer hardware consists of the following cabinets:

- Basic processor cabinet
- Growth cabinet (optional).

The hardware units located in these cabinets (also known as frames) are described in the following sections. Refer to 254-303-105, *Hardware Reference Manual*, for more detail on the 3B21D computer hardware.

Circuit Pack Codes

The circuit pack codes identified in this IP are the minimum circuit pack codes required for the features described.

- Circuit packs with later letter suffix codes may be substituted for the circuit pack codes identified. If a UN133C is identified as the required pack, a UN133C or UN133D may be substituted for the required UN133C.
- Circuit packs with earlier letter suffix codes cannot be substituted for the circuit pack codes identified. A UN133 or UN133B cannot be substituted for the required UN133C.

Suffix Coding of Circuit Packs

Unidirectional interchangeability (also known as downward compatibility or backward compatibility) is defined as the condition where the new design version can be substituted in all applications of the previous design versions without coordinated hardware, software, or database changes. Older design versions cannot be substituted for the new design version where the new design version is required.

When a new or modified circuit pack has unidirectional interchangeability, the new circuit pack code can be the current circuit pack code number followed by a letter suffix or the next letter suffix. For example, the UN133 would be followed by UN133B, UN133C, UN133D, and so forth. (The “A” suffix, such as UN133A, is never used.)

When design changes are so extensive that it is not possible to preserve unidirectional interchangeability on the new design, or when the new design depends on changes on other circuit pack codes, completely new circuit pack code(s) are used.

Advantages of Suffix Coding

New suffix code circuit packs with unidirectional interchangeability have several advantages over new circuit pack codes.

- **Recognition of Circuit Pack Function.**

When maintenance personnel become familiar with the function of a particular code, such as UN133, they will also become familiar with any suffix letter code with the same base number. Thus, if it is known that UN133 is an update circuit, UN133B will also be recognized as an update circuit that incorporates some change from UN133.

- **More Economical Stocking of Spares Is Possible.**

In some instances, a particular code, (such as UN133B) might be used in several applications. If a unidirectional change to UN133C is required for only one application, the customer does not need to maintain spares of both UN133B and UN133C. Only UN133C would need to be kept on hand, since it could be used in all applications.

- **System Software Updates are Reduced.**

The introduction of a later letter suffix circuit pack code does not require system software updates or database changes, until/unless a later letter suffix circuit pack code is required for implementing a feature.

3B20D Model 1 Computer Hardware

A typical 3B20D Model 1 computer hardware consists of the following frames:

- Control Unit Frame
- Peripheral Control Frame
- Moving Head Disk Frame
- Tape Unit Frame
- Input/Output Processor (IOP) and Storage Module Drive (SMD)-Type Disk File Controller (DFC) Growth Frames
- Small Computer System Interface (SCSI) Disk Cabinet (Optional)
- Power Distribution Frame.

The hardware units located in these frames are described in the following subsections. Figure 1-1 is a functional block diagram of the hardware configuration in a typical 3B20D Model 1 computer.

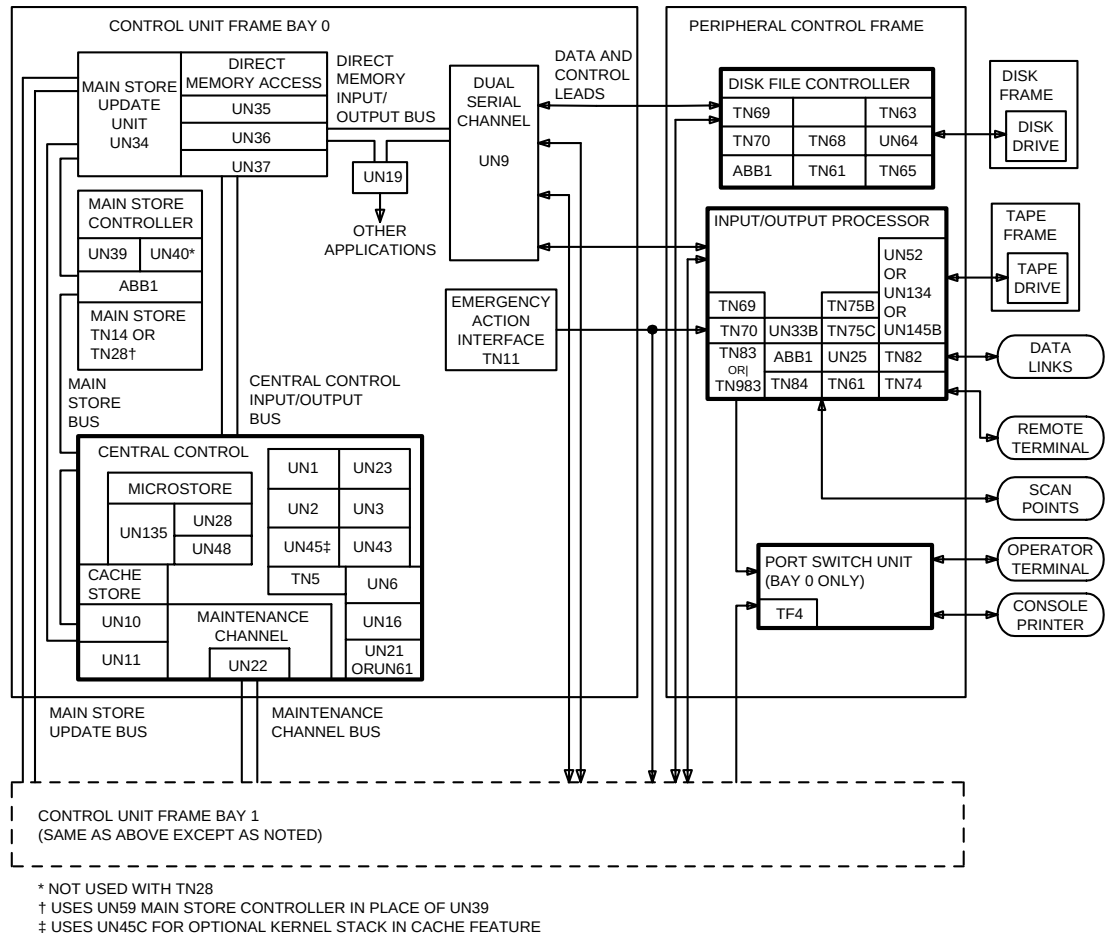


Figure 1-1. Typical 3B20D Model 1 Computer Functional Block Diagram

Control Unit Frame

The computer has two CU frames. The CU frame (Figure 1-2) contains the main computer processing and main memory (computer memory) facilities of the 3B20D Model 1 computer. The units contained in the CU are as follows:

- Direct Memory Access (DMA) Input/Output (IO) Unit
- Central Control (CC) Unit
- Main Store (MAS) Unit
- Cooling Unit
- Power Units.

Direct Memory Access Input/Output Unit

The DMA IO unit contains two DMAs and five programmed IO positions. Each DMA consists of a Direct Memory Access Controller (DMAC – UN35, UN36, and UN37 circuit packs) and up to four Dual Serial Channels (DSCHs – UN9s). The DMA provides direct memory transfers between the MAS and the peripheral devices. Each DMAC controls its four channels over the DMA IO bus. The five IO positions communicate between the CU and peripherals over either the DSCH or the Application Channel Interface (ACHI – UN19). These programmed IO positions are used when more than the two main IO channels in the CC are needed. The interfaces are as follows:

- DSCH – Interfaces with up to 16 peripheral devices over 16 sets of five-pair serial data cables. Two serial data streams are simultaneously transmitted (dual serial). The DSCH operates in either a word-transfer or block-transfer mode.
- ACHI – Allows the CC to communicate with the application. The CC interfaces with the ACHI over the central control IO bus.
- DMAC – Controls data transfers between the DSCHs and the MAS.

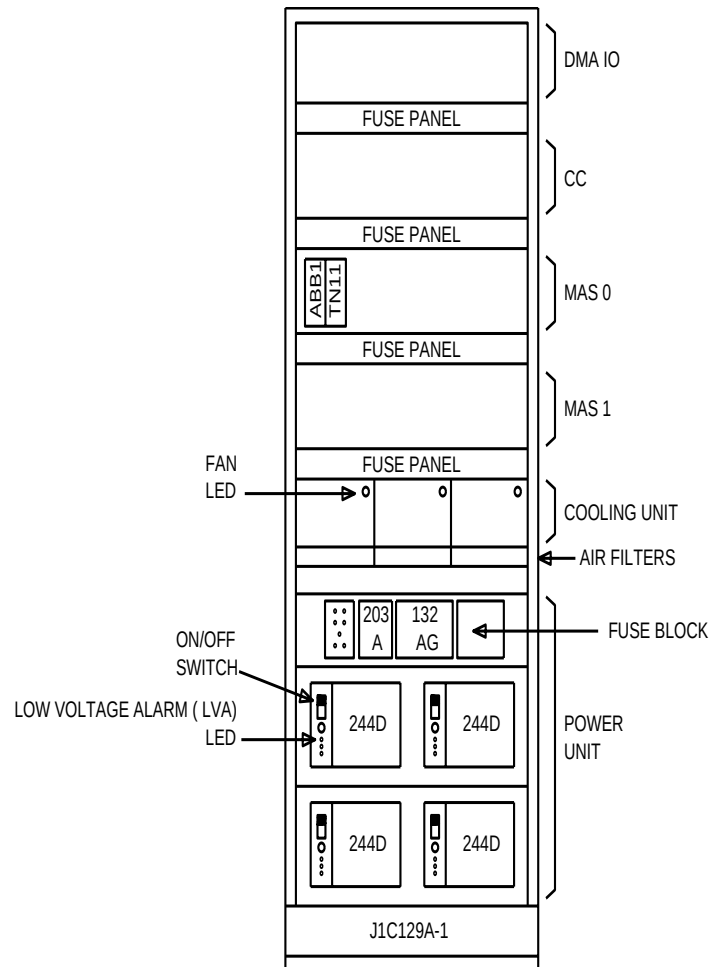


Figure 1-2. 3B20D Model 1 Computer Control Unit Frame

Central Control Unit

The physical organization of the CC unit is as follows:

- Microstore (MIS – UN48 and UN28 circuit packs) – Stores the microinstructions necessary to decode main store instructions.
- Microcontrol (MC – UN135) – Controls the MIS and the microinstruction decoding.
- Data Manipulation Unit (DMU – UN1 and UN23) – Performs all arithmetic and logic operations required by the computer.
- Special Registers (SREG – UN2 and UN3) – Supply, identify, signal, fetch, record, interface, and control functions within the CPU.
- Store Address Control (SAC – UN43) – Controls the store interface circuits and contains the store address.
- Cache Memory (CAM – UN11) – Provides memory with high-speed access (*optional feature*).
- Cache Memory Controller (CAC – UN10) – Controls cache memory addressing and data transfer (*optional feature*).
- Store Address Translator (SAT – UN45 or UN45C) – Translates a virtual address to a corresponding physical address. UN45C is used for an optional kernel stack in the cache feature.
- Store Data Control (SDC – UN6) – Interfaces the CC to the main store. It passes information read from or written to the Main Store Array (MASA – TN14 or TN28).
- Main Store Update (MASU – UN34) – Allows an on-line CU to keep both the active and standby MAS updated with the latest information. The MASU allows the Main Store Update Bus (MASUB) to be shared by both CUs.
- Maintenance Channel (MCH – UN22) – Provides a diagnostic access at the microcode level to the other CU over a serial bus. Under certain error conditions, the MCH forms and transmits messages that require stop-and-switch action.
- Micro Level Test Set Interface (MLTSI – UN16) – Accesses the internal CC buses when plugged in. (The MLTSI is supplied with the Micro Level Test Set.)
- IO positions – Connect the application peripherals to the central control IO bus. They are available for either DSCH (UN9) or ACHI (UN19).
- Utility Circuit (UC – UN21) – Functions as an optional circuit pack used for software debugging and troubleshooting.
- Dual Utility Circuit (DUC – UN61) – Interfaces with the Field Test Set (FTS) and is used in place of the UC.

Main Store Unit

A CU frame can contain up to two main store units. Each main store unit (or module) consists of one Main Store Controller (MASC) that checks incoming data for errors, distributes data to the MASAs, and returns data to other CUs. The MASC can be a combination of the MASC – UN39 and UN40 circuit packs (with MASA – TN14s) or the UN59 circuit pack (with MASA – TN28s). The MASA – TN14s (0.5-MB or 128-K words) require the UN39 and UN40 circuit packs. The MASA – TN28s or TN56s (1-MB or 256-K words) require the UN59 circuit pack. Each main store, the storage area for program instructions and data, can have from 1 to a maximum of 16 arrays. (A maximum of 32 MB of memory storage is allowed.)

A control panel (ABB1) for manual control of the CU power and the Emergency Action Interface (EAI – TN11) circuit pack is located in main store 0. The ABB1 and EAI are not directly associated with the main store.

The ABB1 has an alarm indicator and provides power on/off functions for the CU frame.

The TN11 interprets incoming emergency action requests and attempts to complete the requested function by setting control bits in the CC System Status Register (SSR).

Cooling Unit

Three cooling units force air upward in each CU because of the high packaging density. Each cooling unit is powered by –48 V DC. The fan units have failure alarms.

Power Units

Power is provided by up to four 244D power converters, a 132AJ power unit, and a reference power supply. The power supplies +5 V, –5 V, and +12 V to the hardware units in the CU frame. Each power unit has an alarm circuit.

Peripheral Control Frame

The peripheral control frame (Figure 1-3) contains the following:

- IOP Unit
- SMD DFC Unit.

The peripheral control frame contains a basic IOP, an SMD DFC unit, and three fan units for the IOPs. It has space for a growth IOP and a port switch unit. The peripheral control frame is duplicated.

An ABB1 control panel, located in the controller portion, provides manual control of the IOP. This ABB1 has an alarm indicator for the IOP. A separate ABB1 provides the same functions for the DFC.

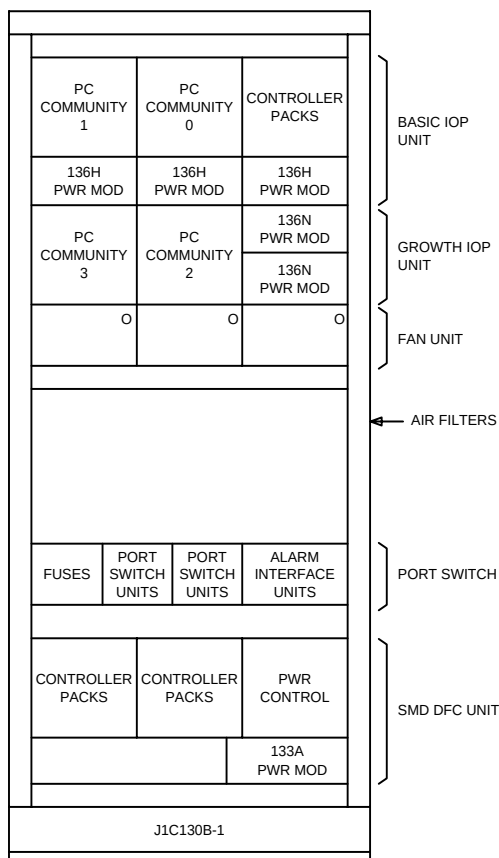


Figure 1-3. 3B20D Model 1 Computer Peripheral Control Frame

Input/Output Processor Unit

The IOP has its own microprocessor to transfer, receive, and buffer data between complementary peripheral devices and the CU. The basic subcircuits of the IOP are as follows:

- Duplex Dual Serial Bus Selector (DDSBS – TN69) – Connects the IOP to the CC DSCH.
- Bus Interface Controller (BIC – TN70) – Buffers data and commands from the DDSBS to the Peripheral Interface Controller (PIC – TN61) and buffers data and status information from the PIC to the DDSBS.
- Peripheral Interface Controller (PIC) – Manages the Peripheral Controllers (PCs) through the Input/Output Microprocessor Interface (IOMI). The PIC consists of two parts: a controller (TN61) and a Microcontrol Store (MCS – TN84).
- Input/Output Microprocessor Interface (IOMI – UN25) – Buffers the information between the PIC and the PCs.
- Peripheral Controllers (PCs) – Serve as an intelligent interface between the IOMI and a slow-to-medium-speed Peripheral Device (PD). Communication between the IOMI and the PC is through a dual-access memory located on each PC. PCs are specialized microprocessor-based controllers.

Storage Module Drive – Disk File Controller Unit

The DFC buffers data between the CU and the Moving Head Disk (MHD) and can control up to eight MHD units.

The SMD DFC unit also utilizes a DDSBS (TN69), BIC (TN70), and PIC (TN61 or TN20). The PIC can use either a TN62 or TN19 circuit pack for the MCS. Other major subcircuits of the DFC are as follows:

- Moving Head Disk Controller (MHDC – TN63) – Interfaces the PIC to the moving head disk drive command and control circuits.
- Parallel Serial Data Interface (PSDI – TN65) – Performs parallel/serial and serial/parallel data format conversion and provides error correction code generation and detection.
- Moving Head Disk Data Clock (MHDDC – TN64) – Interfaces the PSDI with up to eight MHDs, and performs voltage-level conversion on the serial data/clock streams to and from the MHD.

Moving Head Disk Frame

The MHD frame (Figure 1-4) contains the following major units.

- Disk Inverter
- Control Panel
- 300-MB Disk.

Disk Inverter

The disk inverter receives –48 V DC from the local power distribution frame and converts it to AC power for the disk drive.

Control Panel

The control panel controls the inverter that supplies power to the MHD. In addition, the panel is used to remove the MHD from service and restore the MHD to service. The control panel also has an alarm indicator.

300-MB Disk

The 300-MB (unformatted) disk is a secondary memory storage device for programs and data.

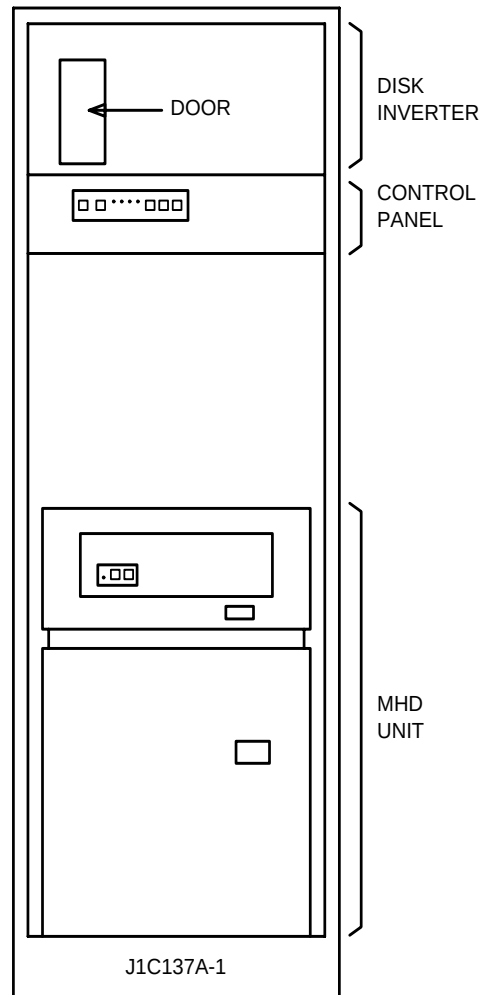


Figure 1-4. 3B20D Model 1 Computer 300-MB Moving Head Disk Frame

Tape Unit Frame

The tape unit frame contains the 9-track Magnetic Tape Units (MTUs) and the format control unit.

Magnetic Tape Unit

The MTU performs read/write functions and controls magnetic tape movement.

Format Control Unit

The format control unit adjusts the data format on read/write operations to allow communication between the Magnetic Tape Controller (MTC) and the MTUs. The format control unit can support up to two MTUs.

IOP and SMD-Type DFC Growth Frames

The peripheral control frames described in the previous paragraphs are required for the 3B20D Model 1 computer operation. The peripheral control frames can be configured as growth frames for different applications as follows:

- IOP growth frame – Similar to the peripheral control frame but contains only IOP units.
- DFC growth frame – Similar to the peripheral control frame but contains only SMD-type DFC units.

SCSI Disk Cabinet (Optional)

The SCSI disk cabinet (Figure 1-5) is used with the units shown when converting a 3B20D Model 1 computer having SMD-type DFCs to one having SCSI-type DFCs.

This capability is optional only with the proper version of operating system software. The SCSI disk cabinet contains the following:

- Power Distribution Unit
- SCSI DFC Units (2)
- Cooling Unit
- SCSI Disk Unit Package (DUP) Units (up to 16 SCSI DUPs).

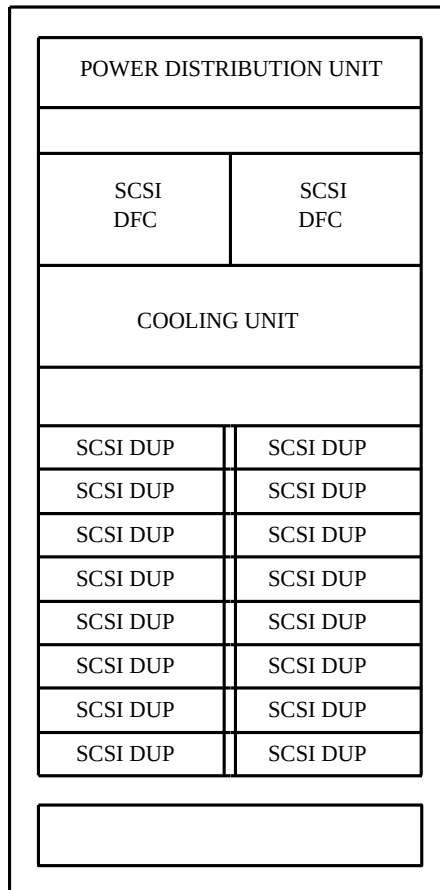


Figure 1-5. 3B20D Model 1 Computer SCSI Disk Cabinet

Power Distribution Unit

The power distribution unit distributes power to, and provides fusing for, the units contained in the SCSI disk cabinet.

SCSI DFC Units

Both SCSI DFC units interface the SCSI disks to the 3B20D computer. In providing this interface, the DFC performs job control, timing, error handling, and SCSI bus protocol. The SCSI DFC can support up to eight SCSI MHDs.

Cooling Unit

The high packaging density of the components on the SCSI DFC circuit boards requires the use of forced air cooling. Each cooling unit is powered by –48 V DC and has failure alarms.

SCSI DUP Units

Each SCSI DUP contains a SCSI disk drive, power supply, power switch, fan, and identity selector switch. Connectors are provided at the rear of the unit to allow attachment to the SCSI bus, input power, and scan/alarm grids.

Power Distribution Frame

The Power Distribution Frame (PDF) (Figure 1-6) distributes office battery voltage (–48 V DC) to all frames within a 3B20D Model 1 computer configuration. The PDF contains filter fuse panels and a control panel.

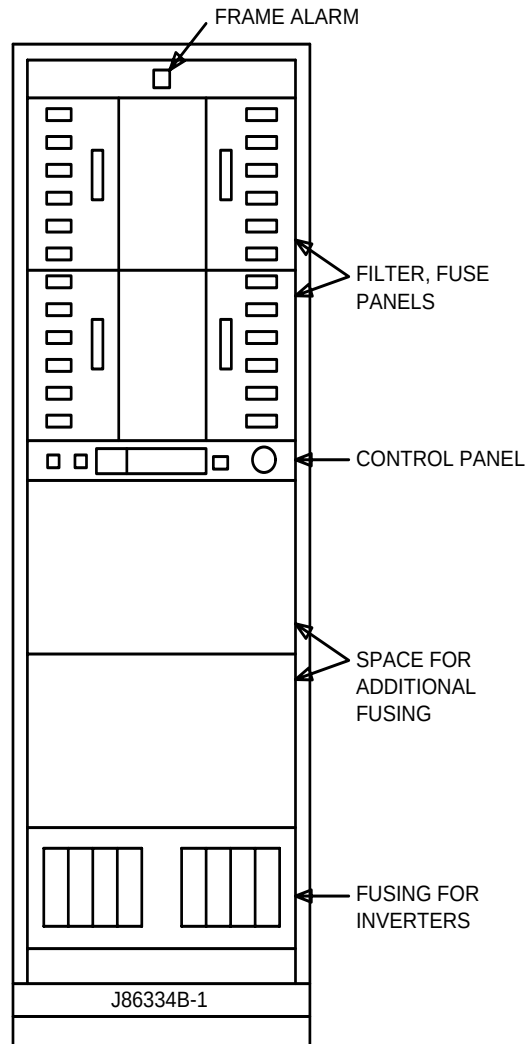


Figure 1-6. 3B20D Model 1 Computer Power Distribution Frame

3B20D Model 2 Computer Hardware

A typical 3B20D Model 2 computer consists of the following frames:

- Processor Control Frame
- 300-MB MHD Frame
- Tape Unit Frame
- Tape/Disk Cabinet (Optional, found in retrofits only)
- SCSI Disk Cabinet (Optional, found in Release 6 only)
- 160-MB Minimodule Disk Frame (Optional)
- Power Distribution Frame.

The hardware units located in these frames are described in the following subsections. Figures 1-7 through 1-9 are functional block diagrams of the hardware configurations found in a typical 3B20D Model 2 computer based on the type of Main Memory as follows:

Figure 1-7, Standard Main Memory – Release 1 (SMM – R1)

Figure 1-8, Extended Main Memory – Release 1 (EMM – R1 Retrofit)

Figure 1-9, Very Large Main Memory – Release 6 (VLMM – R6 Retrofit).

Processor Control Frame

The two-bay (Bay 0 and Bay 1) processor control frame (Figure 1-10) contains the following:

- Central Processor Unit (CPU)
- MAS-IO-DFC Unit
- IOP Basic Unit
- Growth Unit (MAS-IOP) – Optional
- Cooling Unit – Refer to ED-4C378-30 for more information.
- Port Switch Unit (bay 0 only) – Refer to J1C130BC-1 for more information.

Power is supplied to the frame through a power distribution unit.

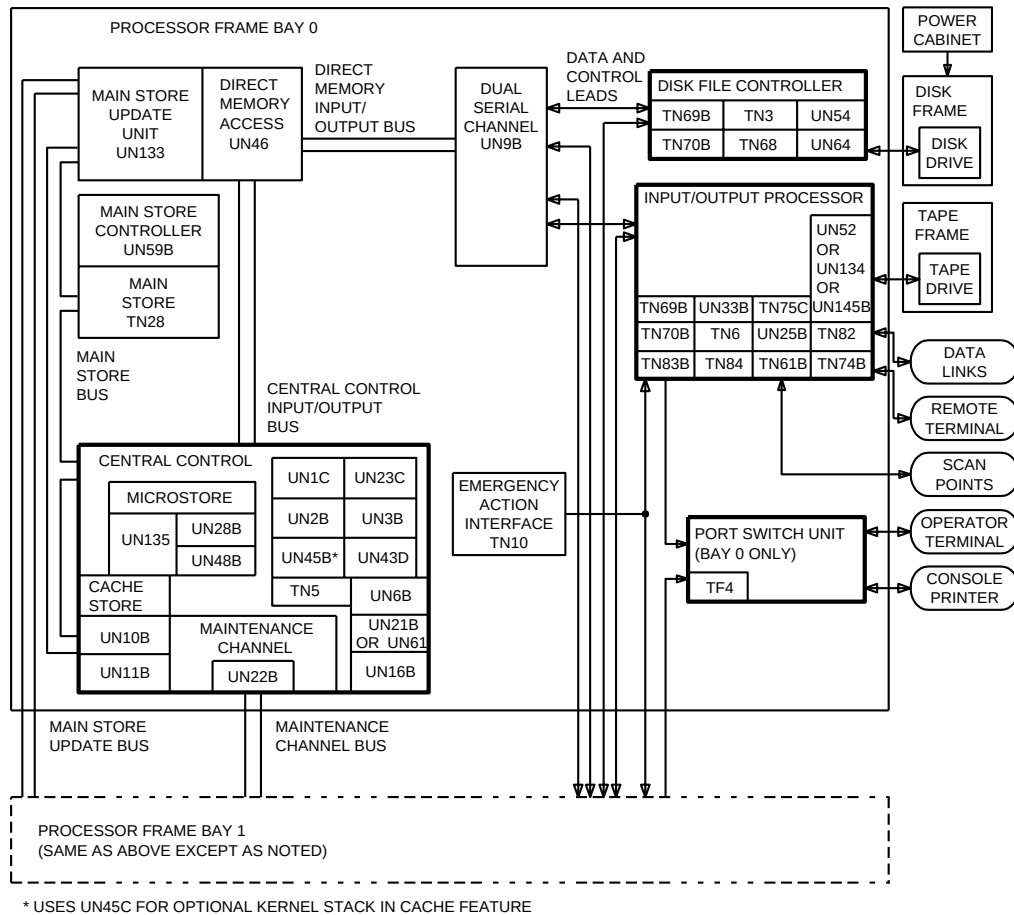


Figure 1-7. Typical 3B20D Model 2 Computer Functional Block Diagram (SMM – Release 1)

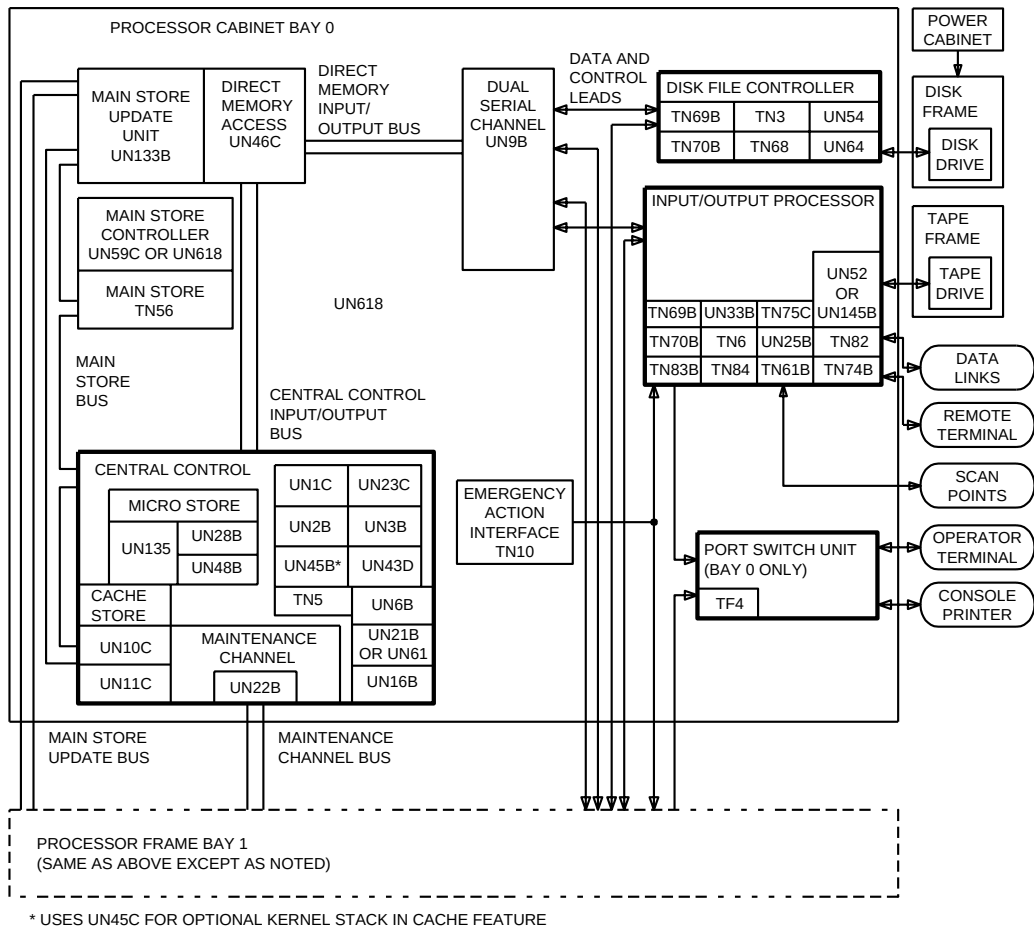


**Figure 1-8. Typical 3B20D Model 2 Computer Functional Block Diagram
(EMM – Release 1 Retrofit)**

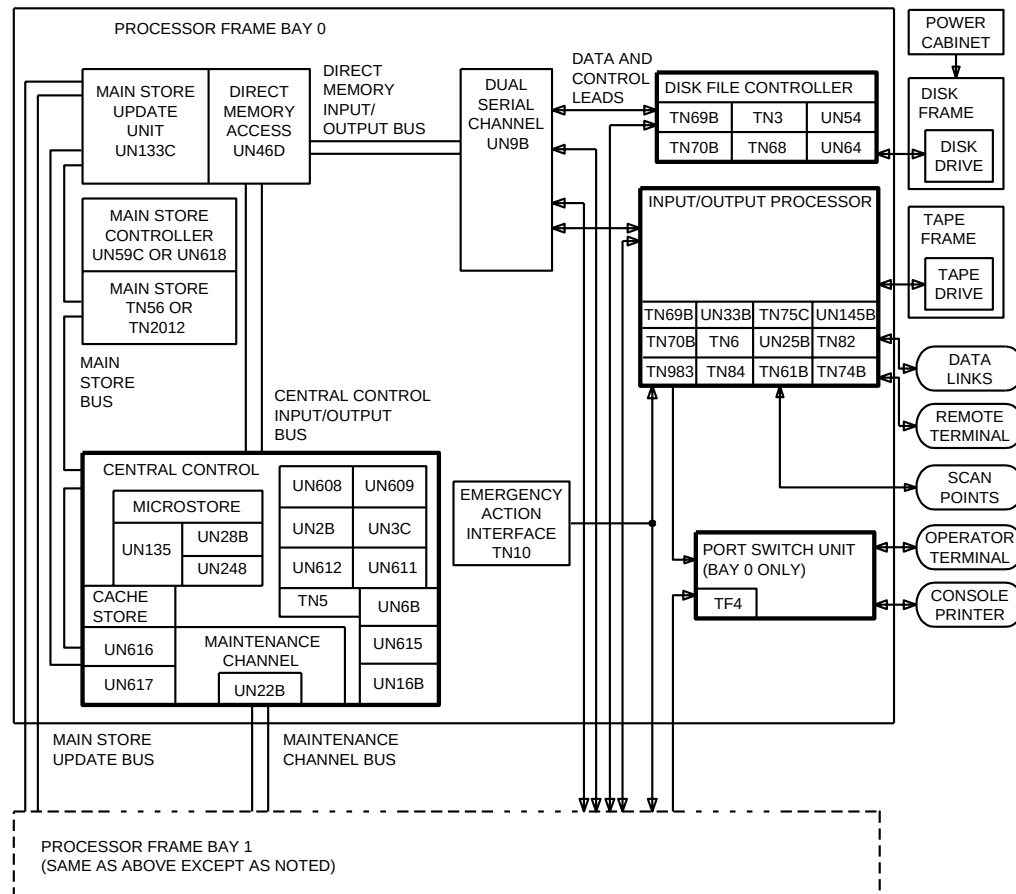


Figure 1-9. Typical 3B20D Model 2 Computer Functional Block Diagram (VLMM – Release 6 Retrofit)

PWR DISTR UNIT J1C147BE-1	PWR DISTR UNIT J1C147BE-1
CPU J1C147BA-1	CPU J1C147BA-1
MAS-IO-DFC UNIT J1C147BB-1	MAS-IO-DFC UNIT J1C147BB-1
MAS-IOP GROWTH UNIT J1C147BC-1	MAS-IOP GROWTH UNIT J1C147BC-1
IOP BASIC UNIT J1C147BD-1	IOP BASIC UNIT J1C147BD-1
COOLING UNIT ED-4C378-30	COOLING UNIT ED-4C378-30
PORT SWITCH UNIT J1C130BC-1	
J1C147B-1 J1C173B-1	J1C147B-1 J1C173B-1
BAY 0	BAY 1

Figure 1-10. 3B20D Model 2 Computer Processor Control Frame

Central Processor Unit

The CPU is the core of the computer. The CPU can be optionally equipped with user microcode, cache memory, Micro Level Test Set Interface, and a utility circuit. Table 1-1 supports SMM – R1, EMM – R1 retrofit, and VLMM – R6 retrofit circuit packs. The physical organization of the CPU is as follows:

- Microstore [MIS – UN28B and Writable Microstore (WMS) – UN48B or UN248] – Stores the microinstructions necessary to decode main store instructions.
- Microcontrol (MC – UN135) – Controls the MIS and starts microinstruction decoding.
- Data Manipulation Unit (DMU – UN1C and UN23C or – UN608 and UN609) – Performs all arithmetic and logic operations required by the computer.
- Special Registers (SREG – UN2B and UN3) – Supply, identify, signal, fetch, record, interface, and control functions within the CPU.
- Store Address Control (SAC – UN43 or UN611) – Controls the store interface circuits and contains the store address.
- Cache Memory (CAM – UN11 or UN617) – Provides memory with high-speed access (optional).
- Cache Memory Controller (CAC – UN10 or UN616) – Controls cache memory addressing and data transfer (optional).
- Store Address Translator (SAT – UN45 or UN612) – Translates a virtual address to a corresponding physical address.
- Store Data Control (SDC – UN6B) – Interfaces the CPU to the main store. Passes information read from or written to the memory array (MASA – TN28, TN56, or TN2012).
- Main Store Update (MASU – UN133) – Allows an on-line CU to keep both the active and standby main store updated with the latest information. The MASU allows the MASUB to be shared by both CUs.
- Maintenance Channel (MCH – UN22B) – Provides a diagnostic access at the microcode level to the other CU over a serial bus. MCH forms and transmits messages requiring stop-and-switch action under certain error conditions.
- Micro Level Test Set Interface (MLTSI – UN16B) – Accesses the internal CPU buses when plugged in. (The MLTSI is supplied with the Micro Level Test Set.)
- Emergency Action Interface (EAI – TN10) – Provides status and communications with the computer during processor recovery phases.
- Utility Circuit (UC – UN21B) – Functions as an optional circuit pack used for software debugging and troubleshooting.

- Dual Utility Circuit (DUC – UN61 or UN615) – Interfaces with the Field Test Set (FTS) and is used in place of the UC.
 - The UN61 is required for *UNIX* RTR Operating System Release 1, Generic Access Package (GRASP).
 - The UN615 is required for *UNIX* RTR Operating System Release 6, GRASP/Enhanced Generic Access Package (EGRASP) (optional).
- Central Processing Unit (CPU) power control (TN5) – Contains manual controls for the CPU.

MAS-IO-DFC Unit

An MAS-IO-DFC unit is located directly below the CPU. Power control for the DFC unit is provided by the TN3 circuit pack.

Interconnection between the CPU and the MAS-IO-DFC unit is via tape cable assemblies. One assembly interconnects the UN133 position in the upper unit to the UN59 or UN618 in the lower unit. (See Table 1-1.) Two more cable assemblies [mainly for the Central Control Input/Output (CCIO) bus] interconnect from terminal fields located (1) between the UN2B and UN3 positions and (2) between the UN48B or UN248 and UN135 positions to a terminal field located between the TN3 and UN64 positions of the lower unit.

The main store portion has an MASC – UN59 or UN618 and positions for eight MASA – TN28 (1 MB), eight MASA – TN56 (2 MB), or eight MASA – TN2012 (4 MB) circuit packs. Bus termination resistor paddleboard assemblies are applied on terminals located on the rear side of the unit. The main store packs are equipped starting with the rightmost position viewed from the front.

The IO section contains two DMAs. Each DMA consists of a DMAC – UN46 and up to two DSCHs – UN9B. The DMA provides direct memory transfers between the main store unit and the peripheral devices. Each DMAC controls its channel(s) over the DMA IO bus.

Table 1-1. 3B20D Model 2 Computer Circuit Packs Used for SMM – R1, EMM – R1 Retrofit, and VLMM – R6 Retrofit

Function	SMM Release 1	EMM Release 1 Retrofit	VLMM Release 6 Retrofit
MIS	UN28B (MC4C077)	UN28B (MC4C077A1C)	UN28B (MC3T0003)
WMS	UN48B	UN48B	UN248
DMU	UN1C UN23C	UN1C UN23C	UN608 UN609
SREG	UN2B UN3B	UN2B UN3B	UN2B UN3C
SAC	UN43B	UN43D	UN611
CAM	UN11B	UN11C or UN617	UN617
CAC	UN10B	UN10C or UN616	UN616
SAT	UN45B†	UN45C	UN612
MASA*	TN28 or TN56	TN56 or TN2012	TN56 or TN2012
MASU	UN133	UN133B	UN133C
MCH	UN22B	UN22B	UN22B
DUC	UN61‡ (UN21B std)	UN61‡	UN615
MASC*	UN59 or UN59B	UN59C or UN618	UN59C or UN618
DMAC	UN46	UN46C or UN145 or TN82	UN46D

* Located on MAS-IO-DFC unit.

† Uses UN45C for optional kernel stack cache feature.

‡ Required for GRASP/EGRASP.

The four IO positions can be equipped to communicate between the DMA and peripherals over either the DSCH or the ACHI. These programmed IO positions are used when more than the two main IO channels in the CPU are needed.

A DSCH can interface with up to 16 peripheral devices over 16 sets of five-pair serial data cables. Two serial data streams are simultaneously transmitted (dual serial) by the DSCH. The DSCH operates in either a word-transfer or block-transfer mode.

The ACHI allows peripheral application devices to access the central control IO bus. The ACHI provides the interface logic and connection to the CPU which allows for communication with the application device.

The SMD DFC unit controls the transfer of data between the CPU and up to eight SMD MHDs. The major subcircuits are as follows:

- Duplex Dual Serial Bus Selector (DDSBS – TN69B) – Connects the DFC to the DSCH.
- Bus Interface Controller (BIC – TN70B) – Buffers data and commands to and from the DFC interface.
- Microcontrol Store (MCS – TN68) – Is the program store (8 K by 40 bits) for the DFC interface.
- Disk File Controller (DFC – UN54) – Controls the interface with the MHD command and control circuits.
- Peripheral Disk Interface (PDI – UN64) – Connects parallel/serial and serial/parallel data format, and detects errors, and voltage level conversions on serial data/clock streams to and from the MHD. The PDI can interface with up to eight MHDs.

IOP Basic Unit

The IOP has a microprocessor to transfer, receive, and buffer data between complementary peripheral devices and the CU. The subcircuits of the IOP are as follows:

- Duplex Dual Serial Bus Selector (DDSBS – TN69B) – Connects the IOP to the DSCH.
- Bus Interface Controller (BIC – TN70B) – Buffers data and commands from the DDSBS to the PIC (TN61B) and buffers data and status information from the PIC to the DDSBS.
- Peripheral Interfaces Controller (PIC) – Manages the PCs through the IOMI. The PIC consists of two parts: a controller (TN61B) and an MCS (TN84).
- Input/Output Microprocessor Interface (IOMI – UN25B) – Buffers the information between the PIC and the PCs.

- IOP power controller (TN6) – Has manual controls for the IOP unit and contains an alarm indicator.
- Peripheral Controllers (PCs) – Serve as an intelligent interface between the IOMI and a slow-to-medium-speed Peripheral Device (PD). Communication between the IOMI and the PC is through a dual-access memory located on each PC. There are eight PCs that are specialized microprocessor-based controllers.
- PC memory power control (TN9) – Provides independent +12 V and –5 V power for PC memories and +12 V and –12 V power for communication interfaces.

Growth Unit

An optional growth unit is located between the MAS-IO-DFC unit and the IOP basic unit. This unit has positions for 8 MB of main store (TN28s), 16 MB of MAS (TN56s), or 32 MB of MAS (TN2012); four IO channels (two usable); and eight PCs. Tape cables connect the main store growth unit with the store portion. The growth main store is equipped starting with the leftmost position, and bus termination diode paddleboard assemblies are applied at the right. The four IO channel positions in the growth unit can also be arranged for either CCIO or DMA Input/Output (DIO) and are cabled to the DMA positions in the unit above the growth unit.

Each peripheral community (2 and 3) in the growth unit also has independent +12 V, –12 V, and –5 V communication power supplied by a dedicated TN9 circuit pack.

300-MB MHD Frame

The MHD frame (Figure 1-11) contains the following major units:

- Disk Inverter
- Control Panel
- 300-MB Disk.

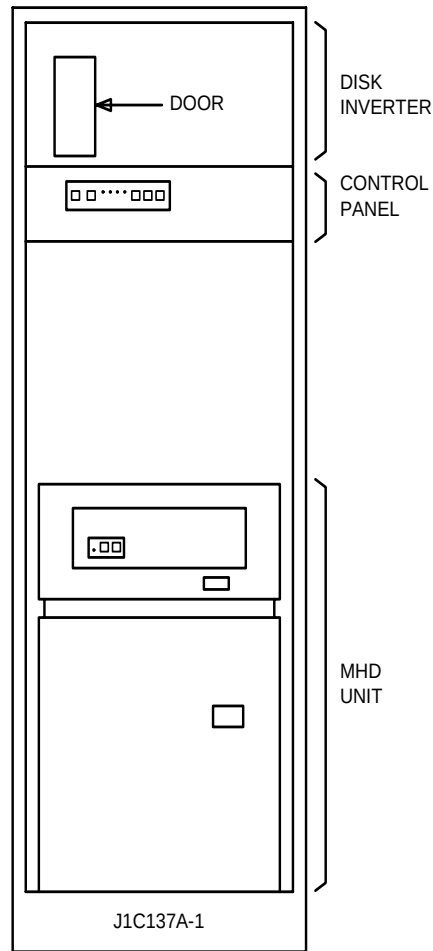


Figure 1-11. 3B20D Model 2 Computer 300-MB Moving Head Disk Frame

Disk Inverter

The disk inverter receives -48 V DC from the local power distribution frame and converts it to AC power for the disk drive.

Control Panel

The control panel controls the inverter that supplies power to the MHD. In addition, the panel is used to remove the MHD from service and restore the MHD to service. The control panel also has an alarm indicator.

300-MB Disk

The 300-MB (unformatted) disk is the secondary memory storage device for programs and data.

Tape Unit Frame

The tape unit frame contains the 9-track magnetic tape drive and 300 V A inverter synthesizer.

Magnetic Tape Unit

The tape unit (KS-22762) performs tape read/write functions and controls tape movement. The format control function is performed by microprocessor circuitry within the tape drive.

300 V A Inverter Synthesizer

The inverter synthesizer (J1C134BA) converts –48 V DC to AC power for use by the tape drive.

Tape/Disk Cabinet (Optional)

The tape/disk cabinet (optional, found in retrofits only) (Figure 1-12) contains the following major units:

- Tape Drive
- 340-MB Disk
- Disk Power Supply
- Disk Power Switch.

The cabinet can be equipped with one tape and up to four disks or up to eight disks without a tape.

Tape Drive

The tape drive can operate at a 1600 bpi or a 6250 bpi density. The tape drive performs tape read/write functions and controls tape movement. The format control function is performed by microprocessor circuitry within the tape drive.

340-MB Disk

The 340-MB (unformatted capacity) disk provides memory storage on a nonremovable disk. Each disk drive requires a power supply and a power switch.

Disk Power Supply

The disk power supply converts -48 V DC power to various DC voltages required for each disk drive.

Disk Power Switch

A disk power switch circuit pack controls the out of service, power up/down, and alarm functions for each disk and power supply unit.

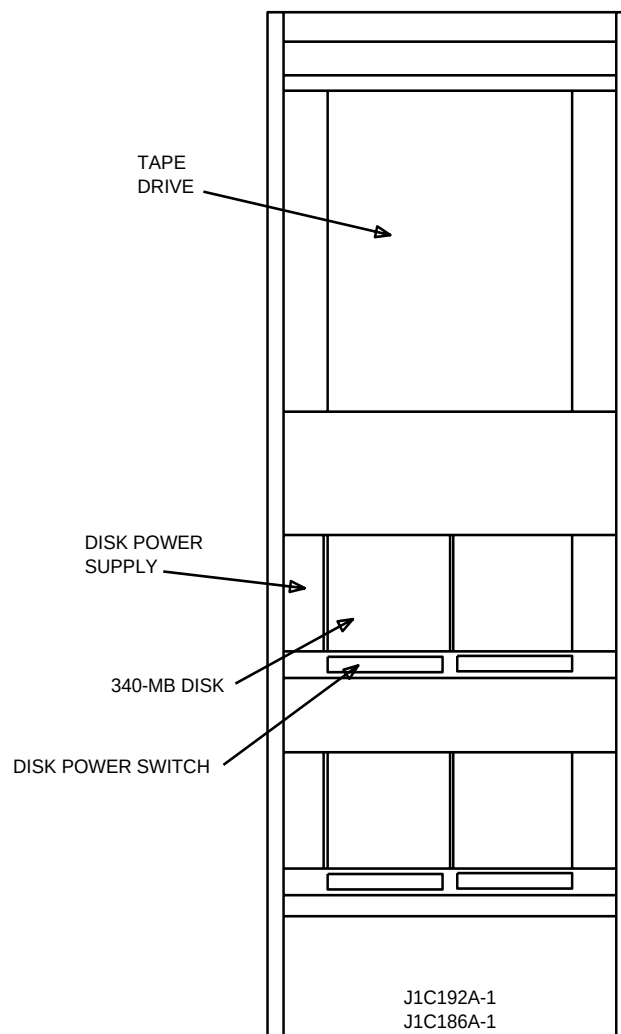


Figure 1-12. 3B20D Model 2 Computer Tape/Disk Cabinet

SCSI Disk Cabinet (Optional, Found in Release 6 Only)

The SCSI disk cabinet (Figure 1-13) is used with the units shown when converting a 3B20D Model 2 computer having SMD-type DFCs to one having SCSI-type DFCs.

This capability is optional only with the proper version of operating system software. The SCSI disk cabinet contains the following:

- Power Distribution Unit
- SCSI DFC Units (2)
- Cooling Unit
- SCSI DUP Units (up to 16).

Power Distribution Unit

The power distribution unit distributes power to, and provides fusing for, the units contained in the SCSI disk cabinet.

SCSI DFC Units

Both SCSI DFC units interface the SCSI disks to the 3B20D computer. In providing this interface, the DFC performs job control, timing, error handling, and SCSI bus protocol. The SCSI DFC can support up to eight SCSI MHDs.

Cooling Unit

The high packaging density of the components on the SCSI DFC circuit boards requires the use of forced air cooling. Each cooling unit is powered by –48 V DC and has failure alarms.

SCSI DUP Units

Each SCSI DUP unit contains a SCSI disk drive, power supply, power switch, fan, and identity selector switch. Connectors are provided at the rear of the unit to allow attachment to the SCSI bus, input power, and scan/alarm grids.

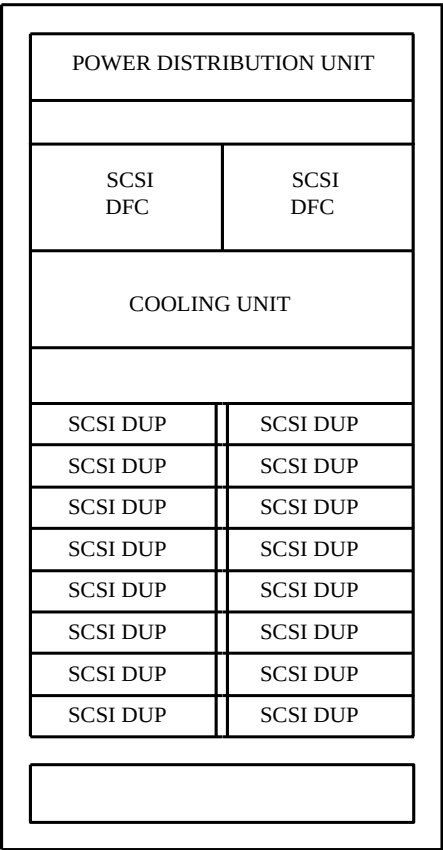


Figure 1-13. 3B20D Model 2 Computer SCSI Disk Cabinet

160-MB Minimodule Disk Frame (Optional)

The 160-MB minimodule disk frame (Figure 1-14) contains up to three 160-MB fixed media disk drives and up to three disk inverters.

Moving Head Disk Units

The 160-MB (unformatted) disk drives provide the 3B20D Model 2 computer with memory storage for programs and data. The 160-MB MHD has nonremovable storage media sealed in a plastic disk module. The modules are “clean-room” assembled to prevent environmental contaminants from damaging the media and heads; they cannot be repaired in the field. However, the module itself, circuit packs, and power supply are replaceable in the field.

Disk Inverters

The disk inverters convert –48 V DC from the local power source for use by the 160-MB MHD. The inverter also contains the power control for the MHD, scan and distribution points, and an indicator for alarms.

Power Distribution Frame

The PDF (Figure 1-15) distributes office battery voltage (–48 V DC) to all frames within a 3B20D Model 2 computer configuration. This frame contains filter fuse panels and a control panel. When 300-MB MHDs are used, the PDF also contains a 40- to 60-amp fuse panel.

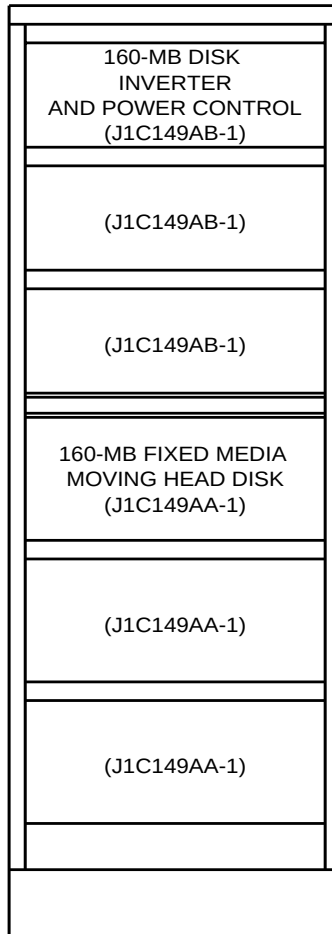


Figure 1-14. 3B20D Model 2 Computer 160-MB Minimodule Disk Frame

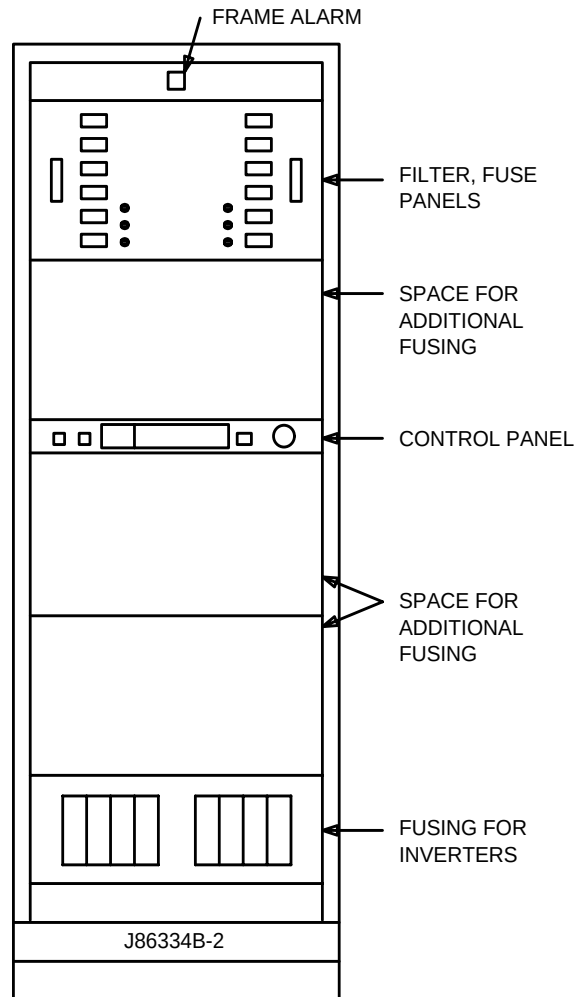


Figure 1-15. 3B20D Model 2 Computer Power Distribution Frame

3B20D Model 3 Computer Hardware

The 3B20D Model 3 computer hardware consists of the following frames:

- Two-Bay Processor Cabinet (Bay 0 and Bay 1)
- Tape/Disk Cabinet
- SCSI Disk Cabinet (Optional)
- Power Cabinet (Optional).

The hardware units located in these frames are described in the following subsections. Figures 1-16 through 1-19 are functional block diagrams of the hardware configurations in a typical 3B20D Model 3 computer based on the type of Main Memory and how the MAS-IO-DFC is equipped (with an SMD DFC unit or a SCSI DFC unit) as follows:

Figure 1-16, Standard Main Memory – Release 1 (SMM – R1)

Figure 1-17, Extended Main Memory – Release 1 (EMM – R1 Retrofit)

Figure 1-18, Very Large Main Memory – Release 6 (VLMM – R6) with
SMD DFC Unit

Figure 1-19, Very Large Main Memory – Release 6 (VLMM – R6) with
SCSI DFC Unit.

Two-Bay Processor Cabinet

The two-bay processor cabinet (Figure 1-20) contains the following:

- Central Processor Unit (CPU)
- MAS-IO-DFC Unit



NOTE:

For *UNIX* RTR Operating System Release 6 only, the MAS-IO-DFC can be equipped with either the SMD DFC unit or the SCSI DFC unit.

- IOP Basic Unit
- Growth Unit (MAS-IOP)
- Cooling Unit – Refer to ED-4C387-30 for more information.
- Port Switch Unit (Bay 0 only) – Refer to J1C130BC-1 for more information.

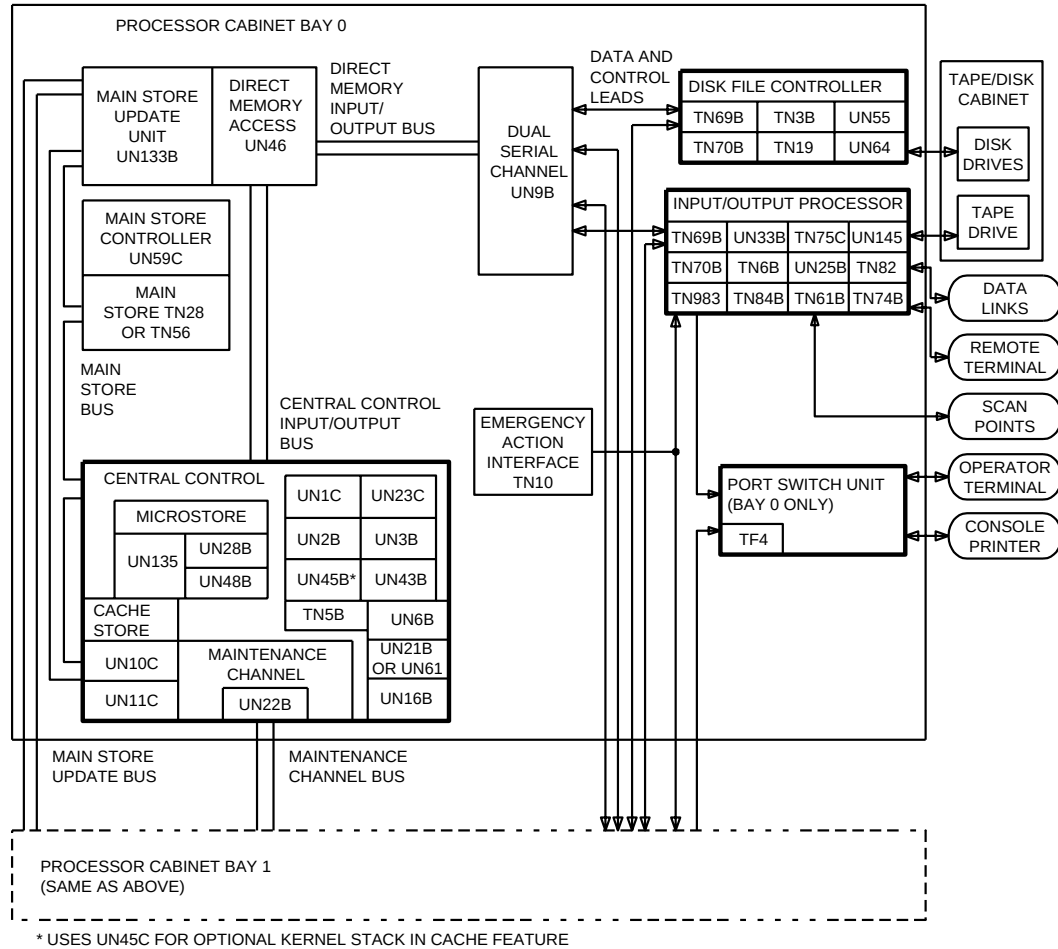


Figure 1-16. Typical 3B20D Model 3 Computer Functional Block Diagram (SMM - Release 1)

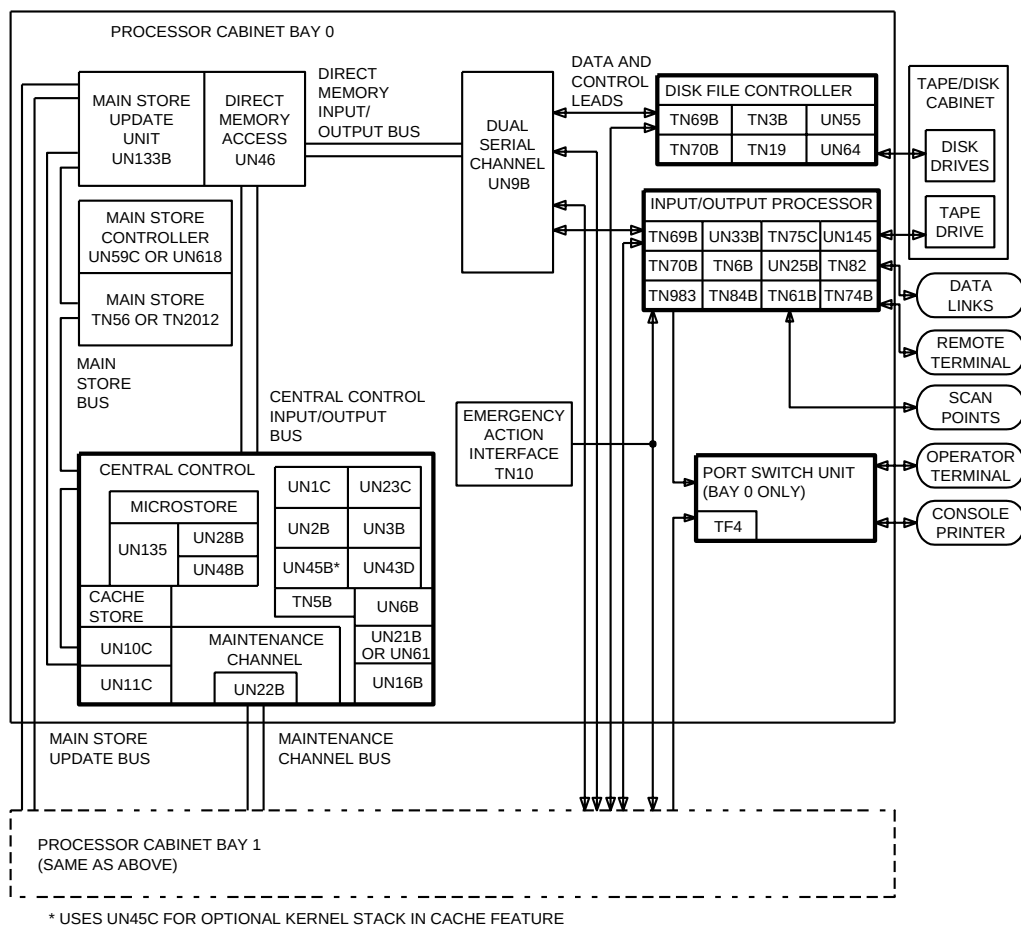


Figure 1-17. Typical 3B20D Model 3 Computer Functional Block Diagram (EMM – Release 1 Retrofit)

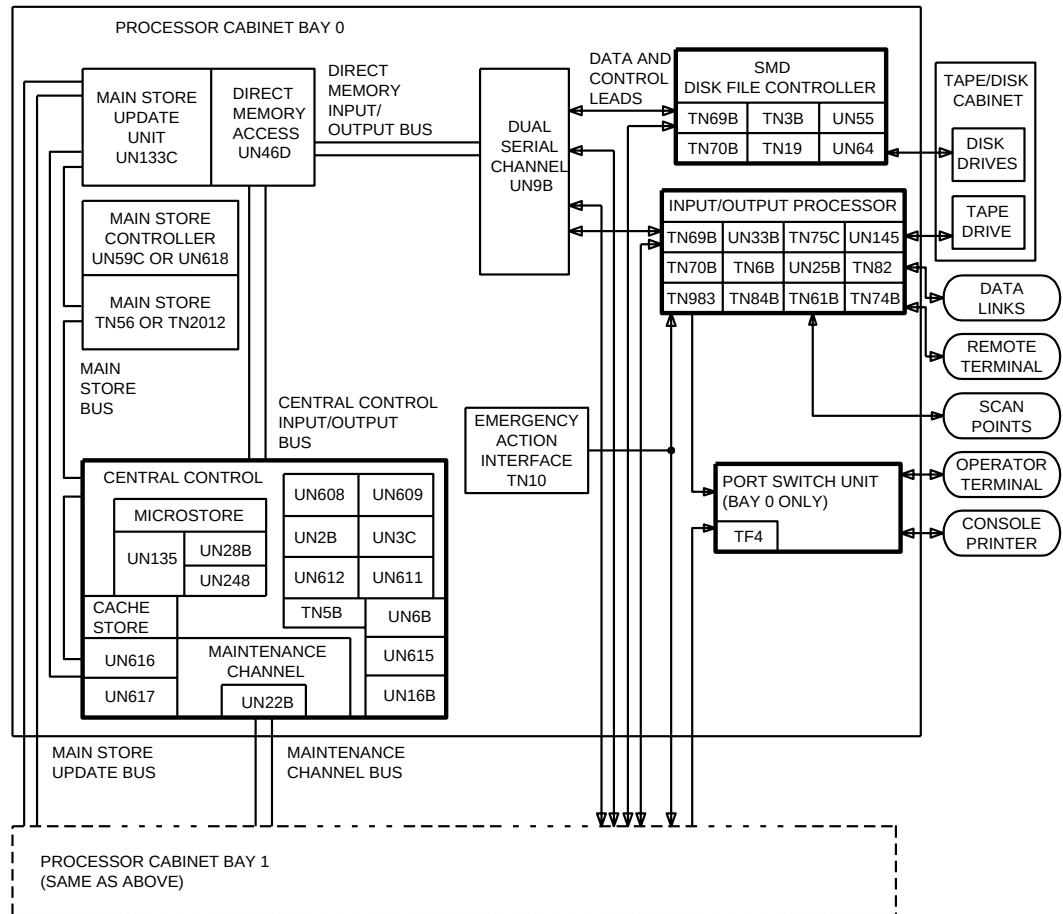


Figure 1-18. Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM – Release 6) with SMD DFC Unit

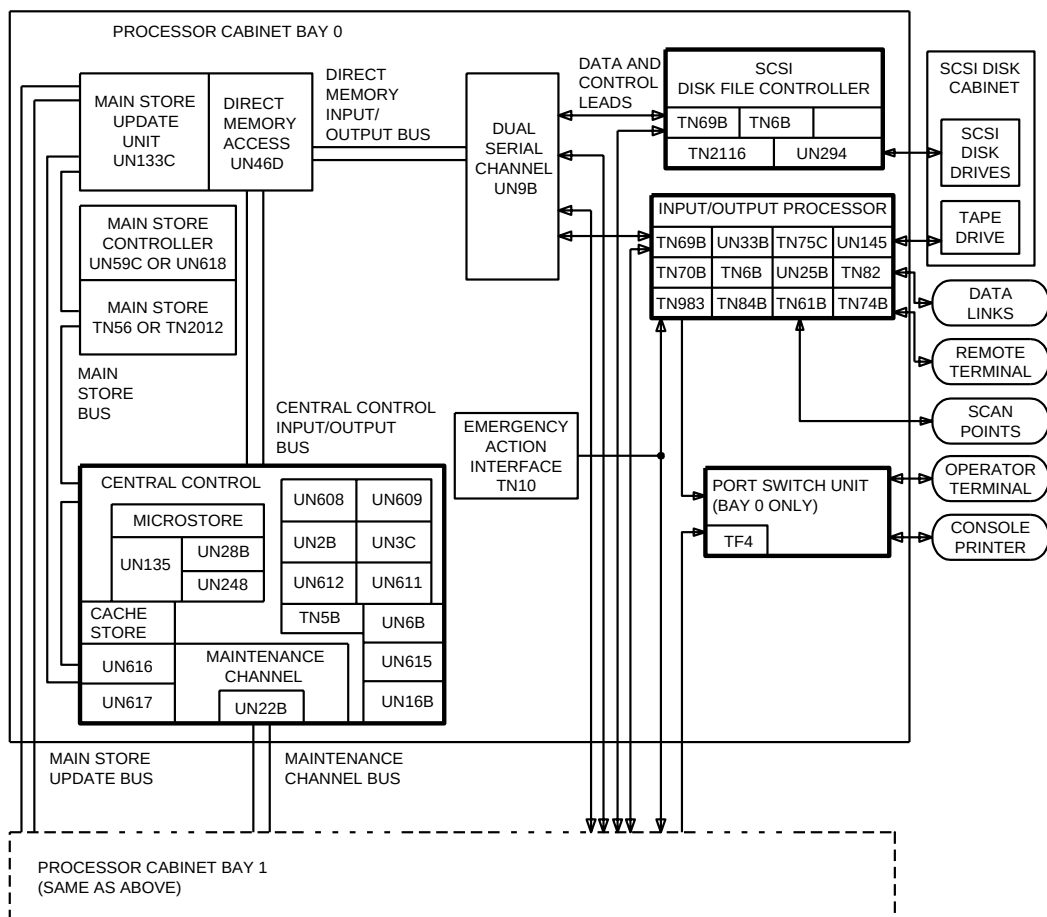


Figure 1-19. Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM – Release 6) with SCSI DFC Unit

PWR DISTR UNIT J1C147BE-1	PWR DISTR UNIT J1C147BE-1
CENTRAL CONTROL UNIT J1C147BA-1	CENTRAL CONTROL UNIT J1C147BA-1
MAS-IO-DFC* UNIT J1C147BB-1†	MAS-IO-DFC* UNIT J1C147BB-1†
FILLER UNIT LIST 1 OR MAIN STORE-IOPG J1C147BC-1	FILLER UNIT LIST 1 OR MAIN STORE-IOPG J1C147BC-1
IOP BASIC UNIT J1C147BD-1	IOP BASIC UNIT J1C147BD-1
COOLING UNIT ED-4C387-30	COOLING UNIT ED-4C387-30
PORT SWITCH UNIT J1C130BC-1	
J1C187A-1, J1C176B-1, J1C176C-1	J1C187A-1, J1C176B-1, J1C176C-1
BAY 0	BAY 1

* EFFECTIVE WITH RELEASE 6, THE DFC CAN BE AN SMD DFC
OR SCSI DFC.

† J3T027AA-1 FOR SCSI.

Figure 1-20. 3B20D Model 3 Computer Processor Cabinet

Central Processor Unit

The CPU is the core of the 3B20D Model 3 computer. The CPU can be optionally equipped with user microcode, cache memory, a Micro Level Test Set Interface, and a utility circuit. Table 1-2 supports SMM – R1, EMM – R1 retrofit, and VLMM – R6 circuit packs. The physical organization of the CPU is as follows:

- Microstore (MIS – UN28B and WMS – UN48B or UN248) – Stores the microinstructions necessary to decode main store instructions.
- Microcontrol (MC – UN135) – Controls the MIS and starts microinstruction decoding.
- Data Manipulation Unit (DMU – UN1C and UN23C or UN608 and UN609) – Performs all arithmetic and logic operations required by the computer.
- Special Registers (SREG – UN2B and UN3) – Supply, identify, signal, fetch, record, interface, and control functions within the CPU.
- Store Address Control (SAC – UN43 or UN611) – Controls the store interface circuits and contains the store address.
- Cache Memory (CAM – UN11 or UN617) – Provides memory with high-speed access (optional).
- Cache Memory Controller (CAC – UN10 or UN616) – Controls cache memory addressing and data transfer (optional).
- Store Address Translator (SAT – UN45 or UN612) – Translates a virtual address to a corresponding physical address. UN45C is used for optional kernel stack in the cache feature.
- Store Data Control (SDC – UN6B) – Interfaces the CPU to the main store. Passes information read from or written to the memory array (MASA – TN28, TN56, or TN2012).
- Main Store Update (MASU – UN133) – Allows an on-line CU to keep both the active and standby main store updated with the latest information. The MASU allows the MASUB to be shared by both CUs.
- Maintenance Channel (MCH – UN22B) – Provides a diagnostic access at the microcode level to the other CU over a serial bus. MCH forms and transmits messages requiring stop-and-switch action under certain error conditions.
- Micro Level Test Set Interface (MLTSI – UN16B) – Accesses the internal CPU buses when plugged in. (The MLTSI is supplied with the Micro Level Test Set.)
- Emergency Action Interface (EAI – TN10) – Provides status and communications with the computer during processor recovery phases.
- Utility Circuit (UC – UN21B) – Functions as an optional circuit pack used for software debugging and troubleshooting.

- Dual Utility Circuit (DUC – UN61 or UN615) – Interfaces with the Field Test Set (FTS) and is used in place of the UC.
 - The UN61 is required for *UNIX* RTR Operating System Release 1, GRASP.
 - The UN615 is required for *UNIX* RTR Operating System Release 6, GRASP/EGRASP (optional).
- CPU power control (TN5B) – Contains manual controls for the CPU.

MAS-IO-DFC Unit

An MAS-IO-DFC unit is located directly below the CC unit. Power control for the DFC unit is provided by the TN3 circuit pack (with SMD) or TN6B circuit pack (with SCSI). Power is supplied for the DFC by the 495FA power unit (with SMD) or 410AB power unit (with SCSI).

Interconnection between the CC unit above and the MAS-IO-DFC unit is via tape cable assemblies. One assembly interconnects the UN133 position in the upper unit to the UN59 or UN618 in the lower unit. (See Table 1-2.) Two more cable assemblies (mainly for the CCIO bus) interconnect from terminal fields located (1) between the UN2B and UN3 positions and (2) between the UN48B or UN248 and UN135 positions to a terminal field located between the TN3 and UN64 positions of the lower unit.

The main store portion has an MASC – UN59 or UN618 and positions for eight MASA – TN56 (2 MB) or eight MASA – TN2012 (4 MB) circuit packs. Bus termination resistor paddleboard assemblies are applied on terminals located on the rear side of the unit. The main store packs are equipped starting with the rightmost position viewed from the front.

The IO section contains two DMA units. Each DMA consists of a DMA controller (DMAC – UN46) and up to two Dual Serial Channels (DSCHs – UN9B). The DMA provides direct memory transfers between the main store unit and the peripheral devices. Each DMAC controls its channel(s) over the DMA IO bus.

The four IO positions can be equipped to communicate between the DMA and peripherals over the DSCH.

A DSCH can interface with up to 16 peripheral devices over 16 sets of five-pair serial data cables. Two serial data streams are simultaneously transmitted (dual serial) by the DSCH. The DSCH operates in either a word-transfer or block-transfer mode.

Table 1-2. 3B20D Model 3 Computer Circuit Packs Used for SMM – R1, EMM – R1 Retrofit, and VLMM – R6 Retrofit

Function	SMM Release 1	EMM Release 1 Retrofit	VLMM Release 6 Retrofit
MIS	UN28B (MC4C077)	UN28B (MC4C077A1C)	UN28B (MC3T0003)
WMS	UN48B	UN48B	UN248
DMU	UN1C UN23C	UN1C UN23C	UN608 UN609
SREG	UN2B UN3B	UN2B UN3B	UN2B UN3C
SAC	UN43B	UN43D	UN611
CAM	UN11B	UN11C or UN617	UN617
CAC	UN10B	UN10C or UN616	UN616
SAT	UN45B†	UN45C	UN612
MASA*	TN28 or TN56	TN56 or TN2012	TN56 or TN2012
MASU	UN133	UN133B	UN133C
MCH	UN22B	UN22B	UN22B
DUC	UN61‡	UN61	UN615
MASC*	UN59 or UN59B	UN59C or UN618	UN59C or UN618
DMAC	UN46	UN46C or UN145B or TN82	
UN46D			

* Located on MAS-IO-DFC unit.

† Uses UN45C for optional kernel stack cache feature.

‡ Required for GRASP/EGRASP.

The ACHI (UN19B) allows peripheral application devices to access the central control IO bus. The ACHI provides the interface logic and connection to the CPU, allowing communication with the application device by programmed IO.

The SMD DFC unit controls the transfer of data between the CPU and up to eight SMD MHDs. The major subcircuits are as follows:

- Duplex Dual Serial Bus Selector (DDSBS – TN69B) – Connects the DFC to the DSCH.
- Bus Interface Controller (BIC – TN70B) – Buffers data and commands to and from the DFC interface.
- Microcontrol Store (MCS – TN19) – Is the program store (8 K by 40 bits) for the DFC interface.
- Disk File Controller (DFC – UN55) – Controls the interface with the peripheral disk interface.
- Peripheral Disk Interface (PDI – UN64) – Provides parallel/serial and serial/parallel data format conversions, error detections, and voltage level conversions on serial data/clock streams to and from the MHD. The PDI can interface with up to eight MHDs.

The SCSI DFC controls the transfer of data between the CU and up to eight SCSI disk drives. The major subcircuits are as follows:

- DDSBS (TN69B) – Connects the DFC to the DSCH.
- Host Adapter (HA) – Interfaces the DDSBS bus, intelligent control logic, and two differential SCSI buses. The HA consists of a UN294 and TN2116 circuit pack.

IOP Basic Unit

The IOP has a microprocessor that transfers, receives, and buffers data between complementary peripheral devices and the CPU. The subcircuits of the IOP are as follows:

- DDSBS (TN69B) – Connects the IOP to the DSCH.
- BIC (TN70B) – Buffers data and commands from the DDSBS to the PIC (TN61B) and buffers data and status information from the PIC to the DDSBS.
- PIC – Manages the PCs through the Selectable Microprocessor Interface (SMI). The PIC consists of two parts: a controller (TN61B) and an MCS (TN84).
- SMI (UN25B) – Buffers the information between the PIC and the PCs.
- IOP Power Controller (TN6B) – Has manual controls for the IOP unit.

- Peripheral Controllers (PCs) – Serve as an intelligent interface between the SMI and a slow-to-medium-speed Peripheral Device (PD). Communication between the SMI and the PC is through a dual-access memory. There are eight PCs that are specialized microprocessor-based controllers.
- PC Memory Power Control (TN9) – Provides independent +12 V and –5 V power for PC memories and +12 V and –12 V power for communications interfaces.

Growth Unit

A growth unit can be located between the MAS-IO-DFC unit and the IOP basic unit. This unit has positions for 16 MB of MAS (TN56s), 32 MB of MAS (TN2012), four IO channels, and eight PCs. Tape cables connect the main store growth unit with the store portion. The growth main store is equipped starting with the leftmost position, and bus termination diode paddleboard assemblies are applied at the right. The four IO channel positions in the growth unit can also be arranged for either CCIO or DIO and are cabled to the DMA positions in the unit above the growth unit.

Each peripheral community (2 and 3) in the growth unit also has independent +12 V, –12 V, and –5 V communication power supplied by a dedicated TN9 circuit pack.

Tape/Disk Cabinet

The tape/disk cabinet (Figure 1-21) contains the following major units:

- Tape Drive
- 340-MB Disk
- Disk Power Supply
- Disk Power Switch.

The cabinet can be equipped with one tape and up to four disks or up to eight disks without a tape.

Tape Drive

The tape drive (KS-23113 or KS-22762) can operate 9-track magnetic tapes at 1600 bpi or 6250 bpi density. The tape drive performs tape read/write functions and controls tape movement. The format control function is performed by microprocessor circuitry within the tape drive.

340-MB Disk

The 340-MB disk (unformatted) provides memory storage on a nonremovable disk. Each disk drive requires a power supply and a power switch.

Disk Power Supply

The disk power supply converts –48 V DC power to various DC voltages required for each disk drive.

Disk Power Switch

A disk power switch circuit pack controls the out of service, power up/down, and alarm functions for each disk and power supply unit.

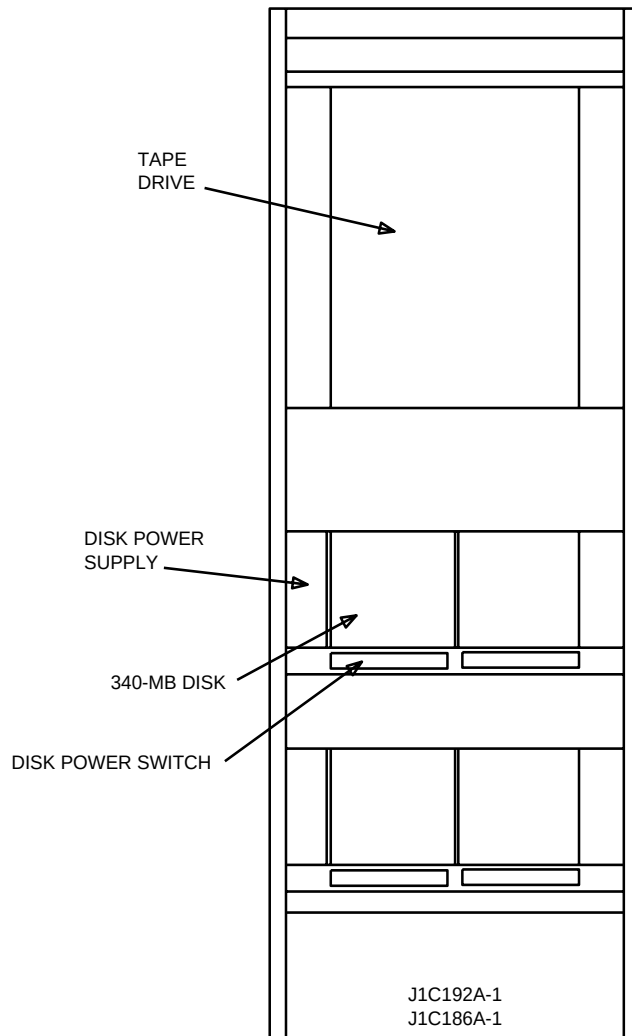


Figure 1-21. 3B20D Model 3 Computer Tape/Disk Cabinet

SCSI Disk Cabinet (Optional)

The SCSI disk cabinet shown in Figure 1-22(A) is used in new systems. It contains the following major units:

- Tape Drive
- SCSI DUP Units (up to 16).

The SCSI disk cabinet shown in Figure 1-22(B) is used in growth/conversion configurations. It contains the following major units:

- Power Distribution Unit
- SCSI DFC Units (SCSI DFCs 2 and 3)
- Cooling Unit
- SCSI DUP Units (up to 16).

The conversion of SMD to SCSI is optional and available only with the proper version of the operating system software.

Tape Drive

The KS-23113 dual density 9-track tape drive can operate at 6250 bits per inch (bpi) or 1600 bpi. The KS-22762 9-track tape drive can operate at 1600 bpi. The tape drive performs tape read/write functions and controls tape movement. The format control function is performed by microprocessor circuitry within the tape drive.

Power Distribution Unit

The power distribution unit distributes power and provides fusing for the units contained in the SCSI disk cabinet.

SCSI DFC Units

The SCSI DFC unit interfaces the SCSI disks to the 3B20D computer. In providing this interface, the DFC performs job control, timing, error handling, and SCSI bus protocol. The SCSI DFC can support up to eight SCSI MHDs.

Cooling Unit

The high packaging density of the components on the SCSI DFC circuit boards requires the use of forced air cooling. Each cooling unit is powered by –48 V DC and has failure alarms.

SCSI DUP Units

Each SCSI DUP unit contains a SCSI disk drive, power supply, power switch, fan, and identity selector switch. Connectors are provided at the rear of the unit to allow attachment to the SCSI bus, input power, and scan/alarm grids.

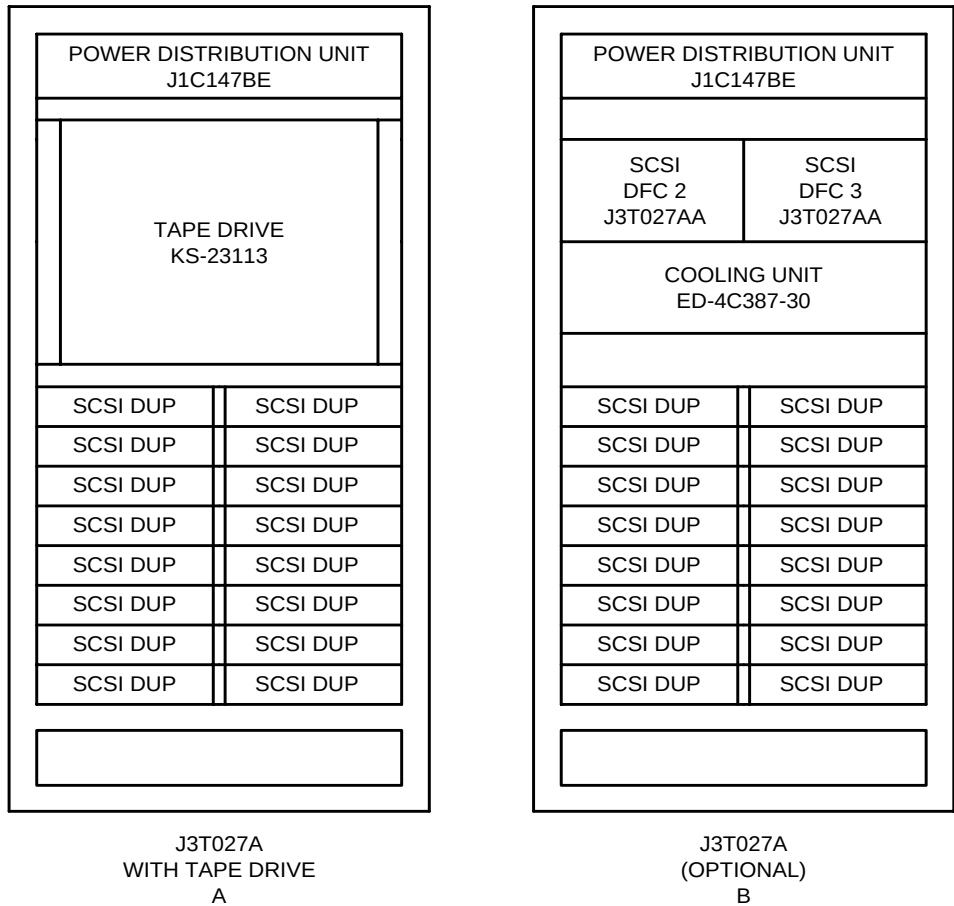


Figure 1-22. 3B20D Model 3 Computer SCSI Disk Cabinet

Power Cabinet (Optional)

The power cabinet (Figure 1-23) provides fusing and power distribution of -48 V DC power on two independent buses to all cabinets.

If -48 V DC power is not available, two AC/DC rectifiers are provided in this cabinet (one for each bus). The rectifiers convert 208 V three-phase AC power to -48 V DC power.

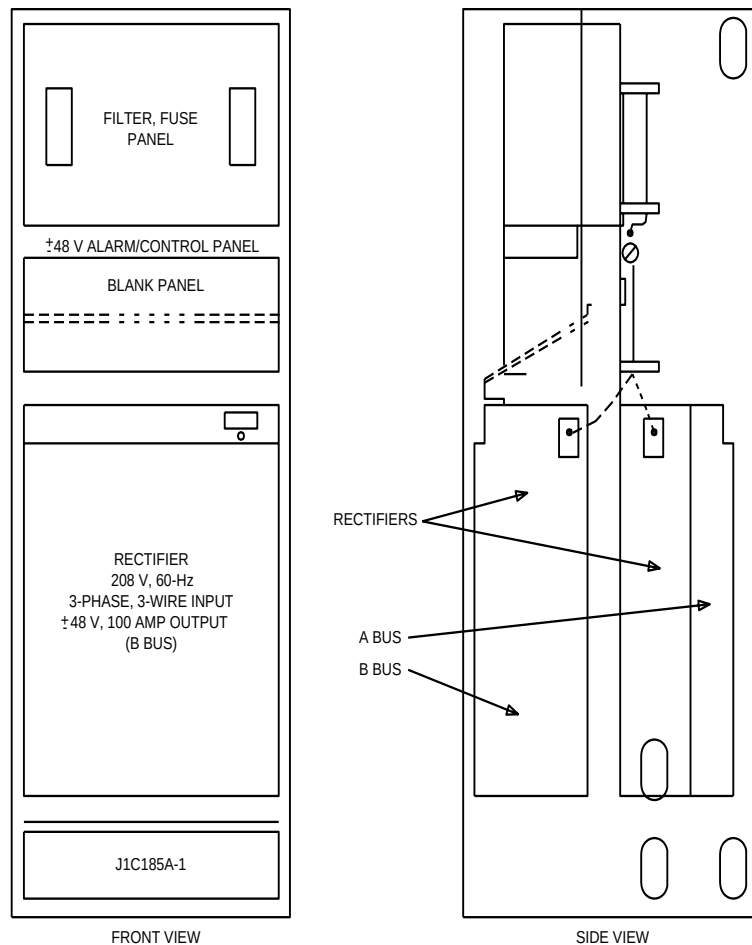


Figure 1-23. 3B20D Model 3 Computer Power Cabinet

3B21D Computer Processor Cabinet

This section provides a very brief overview of the 3B21D computer hardware. For detailed information, refer to 254-303-105, *Hardware Reference Manual*.

The typical 3B21D computer processor cabinet contains the following:

- Processor Unit
 - Control Unit (CU)
 - SCSI DFC Unit
 - IOP Unit
 - SCSI Peripheral Units (possibly).
- Growth Apparatus Housing Unit
- Cooling Unit
- Modular Fuse and Filter Unit.

Each processor cabinet (Figure 1-24) contains two processor units and two growth apparatus housings. The growth apparatus housings are equipped differently according to the configuration.

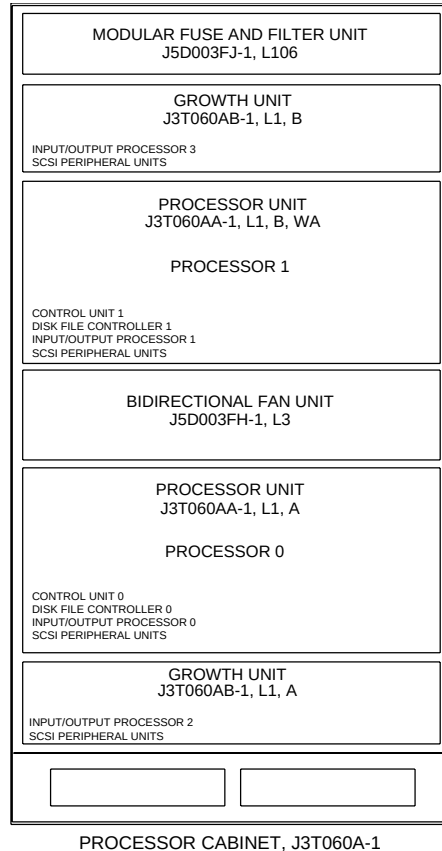


Figure 1-24. 3B21D Computer Processor Cabinet

Control Unit

The CU is the core of the 3B21D computer. The CU consists of the following:

- Central Control
- Cache Storage Unit
- Main Store and Main Store Update
- Direct Memory Access
- Expansion Slots
- Utility Circuit.

Central Control

The 3B21D computer CC is a microprogram-driven machine modeled after the 3B20D computer CC. The CC contains the following functional subsystems:

- Circuit Pack Identification (CPID)
- Microcontroller (MC)
- Microinstruction Store (MIS)
- Maintenance Channel (MCH)
- Emergency Action Interface (EAI)
- Data Manipulation Unit (DMU)
- Special Registers (SREGs)
- Store Address Interface (SAI)
- Store Data Interface (SDI)
- Store Address Translator (SAT)
- Cache Storage Unit (CSU)
- Micro Level Test Set Interface (MLTSI).

Cache Storage Unit

The Cache Storage Unit (CSU) provides a small, high-speed CC cache memory for the most recently accessed MAS words. The CC accesses the CSU concurrently with the MAS access on the MASB. The CSU is part of the CC circuit pack.

Main Store and Main Store Update

The MAS provides a minimum of 32 MB to a maximum of 256 MB of Dynamic Random Access Memory (DRAM). The MAS in both CUs is kept coherent using the update bus. One MAS circuit pack is equipped in each CU. The MAS circuit pack (KLW32 - KLW128) is initially equipped with 32 MB of DRAM. Additional memory is added to the circuit pack in either 8 MB or 32 MB increments. The MAS circuit pack (KLW256) is initially equipped with 256 MB of DRAM.

Direct Memory Access

The DMA provides peripheral devices access to the MAS via the MASB and provides the CC access to the peripheral devices through the CCIO bus. Each DMA supports connections for a maximum of 16 peripheral devices.

Peripheral devices are connected to the DMA through the DSCH. These devices have a dual ported DSCH interface that allows them to communicate with either CU 0 or CU 1. Two DMA circuit packs (KBN15s) are equipped in each CU.

Expansion Slots

The Expansion (EX) slots provide an interface for new (future) hardware features to access the MAS and CC via the Central Control Input/Output (CCIO) bus and the MASB. Each CU has two Expansion slots, EX 0 and EX 1.

Utility Circuit

The UC is an optional function that monitors operations between the CC and the main memory (cache or main store) for the purpose of program debugging and testing. One UC circuit pack (UN379) is optionally equipped in each CU.

Disk File Controller Unit

The DFC provides connection to peripherals with SCSI interfaces, which include disk and tape drives. (The term "DFC" is not accurate since support is not limited to disk peripherals; however, the nomenclature is retained from the 3B20D computer.) Each DFC provides two SCSI buses (SCSI 0 and SCSI 1). A maximum of seven SCSI peripheral devices can be equipped on a SCSI bus. Therefore, each DFC can support a maximum of 14 SCSI devices.

Input/Output Processor Unit

The IOP provides connections to terminals, networks, and some tape devices. The 3B21D computer can have four IOPs (IOP 0 through IOP 3).

- IOP 0 is equipped in the CU 0 processor unit.
- IOP 1 is equipped in the CU 1 processor unit.
- IOP 2 is equipped in the first growth unit.
- IOP 3 is equipped in the second growth unit.

Functionally, an IOP can support four PC communities with each community supporting four PCs. Therefore, 16 peripheral devices can be supported by an IOP.

- IOP 0 and IOP 1 are exceptions; each supporting up to 15 PC slots.
- IOP 2 and IOP 3 each support up to 16 PC slots.
- The maximum number of available PC slots in a basic processor unit is 15.
- The maximum number of available PC slots in a growth unit is 16.

Peripheral Growth Cabinet

The Peripheral Growth Cabinet supports additional DFC, SCSI peripherals, and IOP/PC functions to provide system growth capability. Two growth units can be equipped in the basic processor cabinet. An optional growth cabinet (Figure 1-25) can be added to house peripheral devices, such as tape drives.

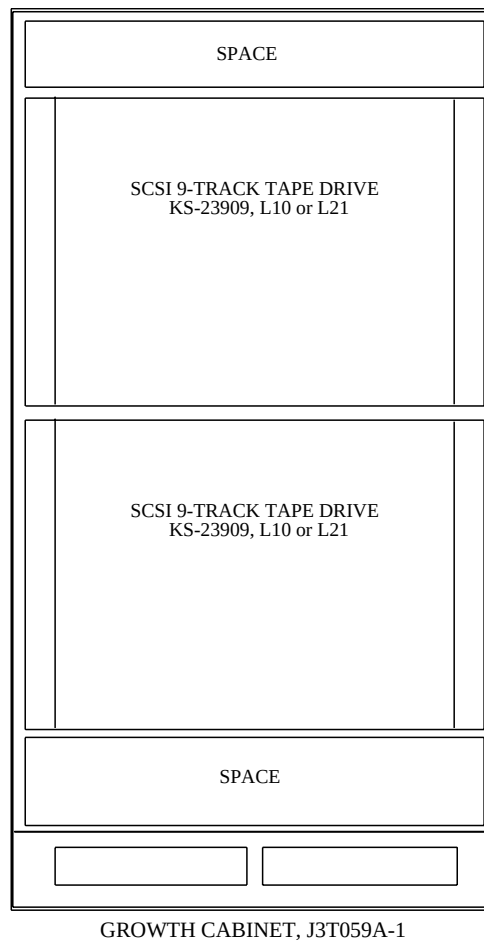


Figure 1-25. 3B21D Computer Growth Cabinet

Contents

Introduction	2-1
Site Maintenance Plan	2-2
Automatic Recovery Actions	2-3
Manual Recovery Actions	2-4
Preventive Maintenance	2-4
■ Routine Hardware Maintenance	2-4
■ Software Exercises	2-5
Responding to Alarms	2-6
■ Diagnostic Tests	2-7
■ Trouble Clearing	2-8
■ Overview of Power Switches	2-10
■ Procedure to Respond to an Alarm	2-15
Documentation	2-17
Using Automatically Generated Data for Troubleshooting	2-17
Switching Control Center Interface	2-18
Training	2-18
Field Support	2-18

Maintenance Plan

2

Introduction

Maintenance for the 3B20D and 3B21D computers is based on the following:

- Site Maintenance Plan
- Automatic Recovery Actions
- Manual Recovery Actions
- Preventive (Routine) Maintenance
- Responding to Alarms
- Documentation
- Using Automatically Generated Data for Troubleshooting
- Switching Control Center (SCC) Interface
- Training
- Field Support.

The maintenance personnel, following established site procedures, is responsible for keeping the computer operating fault free and returning the computer to service when faults cause partial or complete loss of system capabilities.

Site Maintenance Plan

Each 3B20D and 3B21D computer site should have a comprehensive plan for maintaining the computer. The plan should contain the following items:

- Up-to-date, well organized, and easily accessible documentation
- System activity log so that changes, problems, and solutions can be tracked and accessed by anyone needing information on the system
- Procedure for recognizing system trouble indicators
- Procedure for routinely examining Receive-Only Printer (ROP) output, log files, plant measurement data, and error files
- List of names and phone numbers for reporting troubles
- Predetermined escalation procedures for trouble-clearing
- Procedure for testing and rotating spare equipment
- Procedure for checking test equipment periodically
- Routine maintenance procedures for cleaning equipment, replacing air filters, and examining disk heads (300-MB disks only)
- Routine tests to ensure the stable operation of the computer
- Backup copies of system software in spare disk and tape formats
- Procedure for installing and testing software and hardware changes as these are made available.

Automatic Recovery Actions

An automatic recovery action is one that the machine takes in response to a fault detected through the self-checking features of the hardware or software.

Automatic recovery actions include:

- Interrupt – An interrupt has either a hardware or software source and results in minor recovery action. Interrupts are monitored by threshold counters that can signal initializations if the thresholds are exceeded.
- On-line Initialization – An on-line initialization is a hardware, microcode, and software sequence that causes hardware and process initialization in the on-line Control Unit (CU).
- Stop and Switch – A stop and switch is similar to an on-line initialization except that a CU switch occurs. A memory copy occurs if the memory to which the CU is being switched is not current with the memory from which the CU is being switched.
- Soft Switch – A soft switch is a switch between CUs without an accompanying initialization.
- *UNIX*® Real-Time Reliable (RTR) Operating System Phase – An operating system phase is a hardware and software initialization of the operating system processes and hardware. Applications usually initialize their hardware and software as well.
- Application Phase – An application phase is a hardware and software initialization of the application system.

Analyze any automatic recovery action that occurs to determine and correct the cause. Failure to take action on non-service-affecting recoveries can lead to more frequent or more severe recoveries and system downtime.

Manual Recovery Actions

A manual recovery action is an action that is requested manually to clear a problem or return equipment to service. Manual recovery actions include the following:

- Selecting boot time parameters
- Initializing the machine.

Four levels of operating system phases can be requested manually. (Refer to Chapter 3, "Recovery.") Each phase affects the system differently.

The number and significance of application phases differ according to the application. They can be requested manually with or without operating system phases. An application phase is called automatically if an operating system phase is requested manually. However, if an application phase is requested manually with no request for an operating system phase, no operating system phase is called. Refer to application documentation for details on application phases.

Preventive Maintenance

Preventive (routine) maintenance is performed on a specific schedule to ensure that the system continues to operate properly. Routine procedures should be performed during periods of light processor activity since most routine procedures require the unit be removed from service, which leaves the system in simplex operation.

Two types of preventive maintenance that must be performed are routine hardware maintenance and software exercises.

Routine Hardware Maintenance

Routine hardware maintenance includes cleaning and replacing filters, cleaning tape heads and guides, cleaning the moving head disk, and checking voltage levels. These procedures are described in the following information products (IPs):

- 254-301-811
*3B20D Model 1 Computer
Common Systems Routine Tasks*

- 254-302-831
*3B20D Model 2 and Model 3 Computers
Common System Routine Tasks
(PDS Formats)*
- 254-302-841
*3B20D Model 2 and Model 3 Computers
Common System Routine Tasks
(MML Formats)*
- 254-303-101
*3B21D Computer
Routine Maintenance Tasks*

For 3B21D computer hardware-specific or 3B21D computer trouble clearing information, refer to the following:

- 254-303-105
*3B21D Computer
Hardware Reference Manual*
- 254-303-102
*3B21D Computer
Trouble Clearing Tasks*

Software Exercises

Software exercises, such as demand diagnostics and system checks, must be performed to help identify problem areas before they cause system outage time. Refer to the following chapters for maintenance information:

Chapter 6	Diagnostics
Chapter 8	Log File Review
Chapter 9	File-System Maintenance
Chapter 15	Input/Output Processor Magnetic Tape Unit
Chapter 17	Spare Circuit Packs
Chapter 18	Maintenance Tools

Responding to Alarms

In the event of a trouble condition, system outage, or any other condition requiring manual intervention, an alarm is generated to alert the operator so proper action can be taken. The activation of an alarm generates an audible signal (provided by the application and dependent on the level and severity of the alarm), visual indications, and an output message on the console printer. Based on their importance, there are three levels of alarms.

- **Critical**—Generated as a result of total loss of processing or when a manual level 4 recovery phase is initiated. The **SYS EMER** and the **CRITICAL** labels in the status summary area of any maintenance terminal display page flashes. An immediate response by the operator is required.
- **Major**—Generated as a result of recovery Phases 1 through 3 or when a hardware fault is detected. The **MAJOR** label in the status summary area flashes. The operator should respond, but with somewhat less urgency than required for a critical alarm.
- **Minor**—Generated as a result of an interrupt, audit failure, or an out-of-tolerance voltage, when a peripheral unit is removed from service manually. The **MINOR** label in the status summary area is lighted. The operator should note the alarm, but immediate corrective action is not needed.

In summary, failures are reported to the craft by one or more of the following fault indicators:

- Audible alarm
- Visual alarm or labels on the maintenance terminal display pages
- Circuit pack alarm Light-Emitting Diodes (such as ALM LEDs)
- Output messages on the maintenance terminal or ROP.

Audible alarms (when equipped) and labels on the maintenance terminal display are retired by pressing the **ALM RLS** key on the maintenance terminal or the Alarm Cutoff/Test (ACO/T) switch on the unit power switch. However, minor alarms will automatically retire in 6 seconds.

Once an alarm occurs, emphasis is on corrective maintenance. The basic philosophy in retiring an alarm is to identify a fault as rapidly as possible, isolate the fault, and restore operations in a minimum amount of time. It is imperative that you know how to retire alarms before they occur. Take the proper training, get familiar with the available information products (documentation), and gain experience from others.

Failures in 3B20D/3B21D computers are software or hardware detected, isolated to a frame, and reported to the craft with sufficient details to locate and remove the malfunction. The main emphasis should be to get familiar with the equipment and corrective action procedures. When an alarm occurs, the faulty unit must be located and repaired.

There are several good training courses that teach hands-on maintenance. However, 3B20D/3B21D computer training courses are normally combined with other courses such as 5ESS® switch training and are not stand-alone courses. Also, refer to 254-303-102, *Trouble Clearing Tasks*, for the 3B21D computer.

Diagnostic Tests

A diagnostic test is the primary tool to identify and isolate a fault and to ensure that proper and effective corrective action was taken. Diagnostic tests are run using the **DGN** input command or the menu (poke) commands on the MTTY. To learn more about running diagnostics using the **DGN** input command, refer to the "Performing Diagnostics" section in Chapter 6, "Diagnostics," and to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*. To learn more about using menu (poke) commands to run diagnostics, refer to Chapter 12, "Craft Interface."

Diagnostic test results are output messages that indicate All Tests Pass (ATP), Conditional All Tests Pass (CATP), No Tests Run (NTR), or Some Tests Fail (STF). CATP and NTR indicate that some system resource was not available to the system to complete the diagnostic. The STF response provides the failing diagnostic phase and the failing test.

The TLP option when used with the **DGN** input command generates a list of suspected faulty equipment if failures are detected. To learn more about interpreting diagnostic output messages, refer to Chapter 6, "Diagnostics," and to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*. To learn more about the TLP option and taking 3B21D computer corrective action, refer to 254-303-102, *Trouble Clearing Tasks*, for the 3B21D computer.

Trouble Clearing

Once diagnostics have been used to isolate and identify possible faulty equipment, the suspected equipment needs to be replaced with equipment that works properly. The trouble clearing procedures used to correct faults can be found in the following IPs:

- 254-303-102, *Trouble Clearing Tasks*, for the 3B21D computer
- 254-302-832, *Trouble Clearing (PDS Format)*, for the 3B20D computer
- 254-302-842, *Trouble Clearing (MML Format)*, for the 3B20D computer.

Also, see the section “How to Remove and Restore Units” in Chapter 6.

When a unit is removed from service, a minor alarm will sound and an output message will be received similar to one of the following:

```
RMV DFC a COMPLETED
RMV DUI a COMPLETED
RMV DUIC a COMPLETED
RMV HSD a COMPLETED
RMV HSDC a COMPLETED
RMV IOP a COMPLETED
RMV SCSDC a COMPLETED
RMV SDL a COMPLETED
RMV SDLC a COMPLETED
RMV MHD a COMPLETED
RMV MT a COMPLETED
RMV MTC a COMPLETED
RMV MTTY a COMPLETED
RMV MTTYC a COMPLETED
RMV ROP a COMPLETED
RMV SBUS a COMPLETED
RMV SCC a COMPLETED
RMV TTY a COMPLETED
RMV TTYC a COMPLETED
```

Where a = member number

Once suspected faulty equipment has been replaced, verify the corrective action fixed the problem by repeating diagnostics. If the problem was fixed, an ATP output message will be received. Sometimes multiple circuit packs will be suspected. When this occurs, go through the process of replacing one circuit pack at a time as indicated in the following procedure:

Step	Procedure
1	Remove the unit from service and power down the failing unit.
2	Replace the circuit pack listed.
3	Restore power and restore the unit to service.
4	Repeat diagnostics.
5	If diagnostics results in an ATP message, stop! The problem has been corrected.
6	When diagnostic results remain the same (no ATP message), remove the unit from service, power down the failing unit, and reinstall the original circuit pack.
7	Repeat Steps 1 through 6 to replace each circuit pack listed until an ATP message is received or all circuit packs listed have been checked.

Overview of Power Switches

Hardware faults will light the ALM LED on the power switch that controls the faulty equipment and power down that unit. **The ACO/T switch at the faulty unit power switch must be operated to retire the alarm caused by a hardware fault.**

The faceplate of the TN2, TN3, TN5, and TN6 3B20D computer power switches is shown in Figure 2-1. If a TN9 power unit in an Input/Output Processor (IOP) is the cause of the fault, the ALM LED on the power unit will light as will the ALM LED on the TN6 power switch.

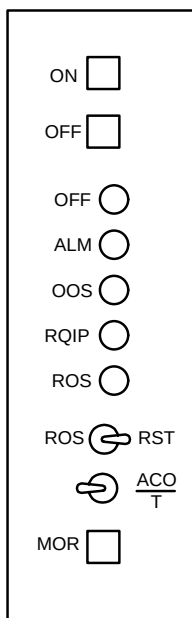


Figure 2-1. 3B20D Computer TN2, TN3, TN5, and TN6 Power Switches Faceplate

For the 3B21D computer, the power switches for the CU and IOP power groups are the TN1821 and TN1820, respectively. The TN1821 is strictly a power switch. However, the TN1820 serves three purposes:

- It is the power switch for the IOP.
- It generates and monitors the miscellaneous voltages (–12 V and –5 V) for the Peripheral Controller (PC) boards. In this function, it replaces the four 3B20D computer TN9 packs.
- It interfaces the secondary and growth boundary scan buses that bring Built-In Self Test (BIST) and self-ID to the IOP, Disk File Controller (DFC), and growth unit.

The power switch for the DFC is part of the UN373 circuit pack which, along with the TN2116 and a 410AA DC-DC converter, comprise the 3B21D computer DFC.

Figure 2-2 is an illustration of the UN373 faceplate. The TN1821, TN1820, and UN373 faceplates are identical in arrangement but of different widths. However, the UN375 disk pack (Figure 2-3) and the UN376 tape pack (Figure 2-4) have different layouts. Refer to 254-303-105, *Hardware Reference Manual*, for switch and indicator functions of these units.

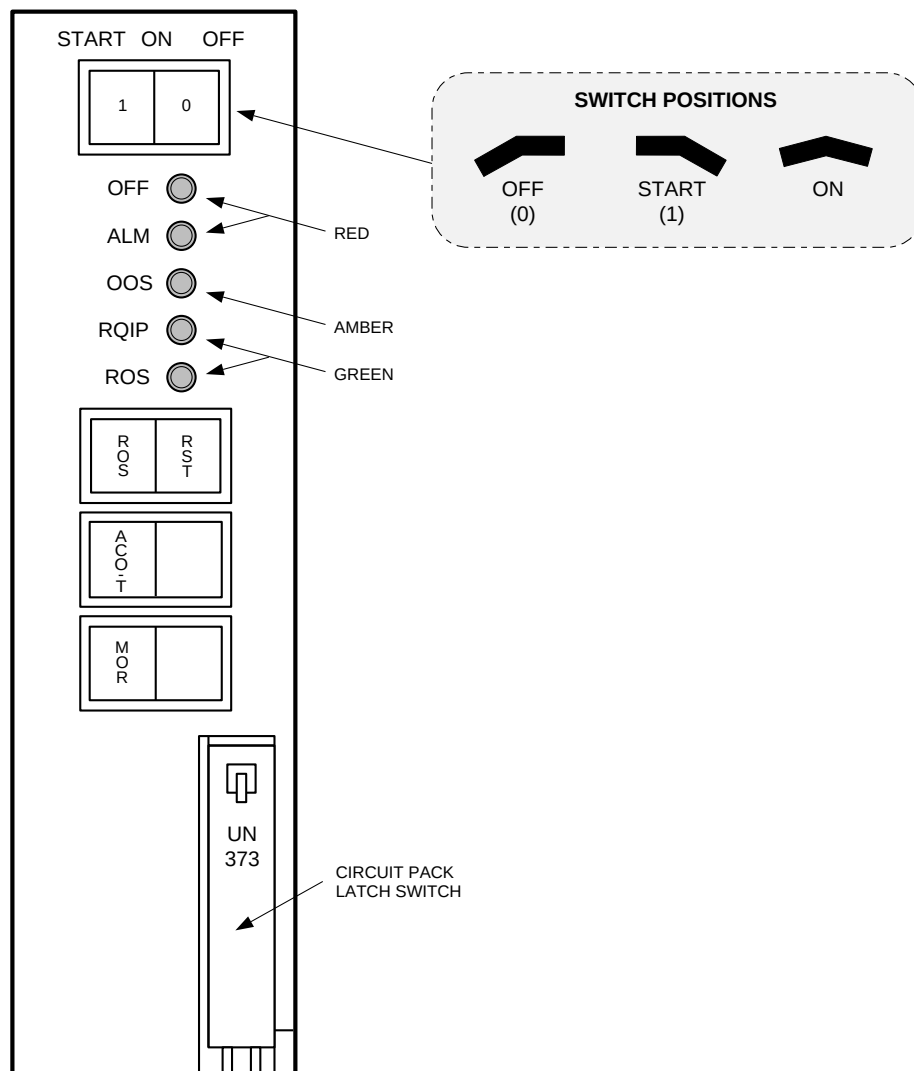


Figure 2-2. 3B21D Computer UN373 (CU Power Switch) Faceplate

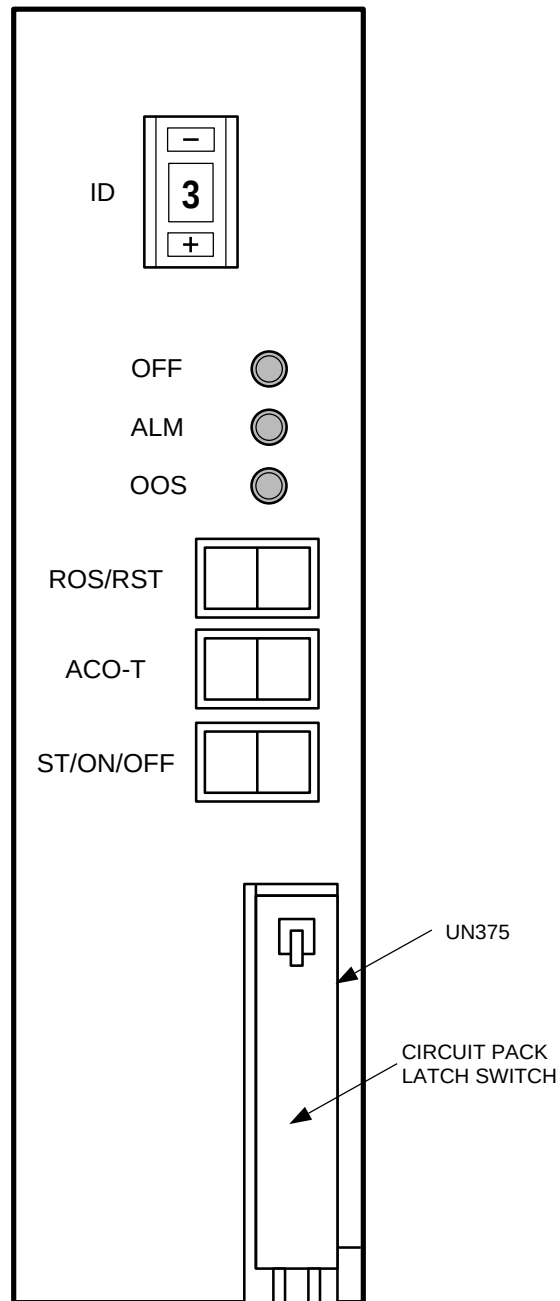


Figure 2-3. 3B21D Computer UN375 (CU Power Switch) Faceplate

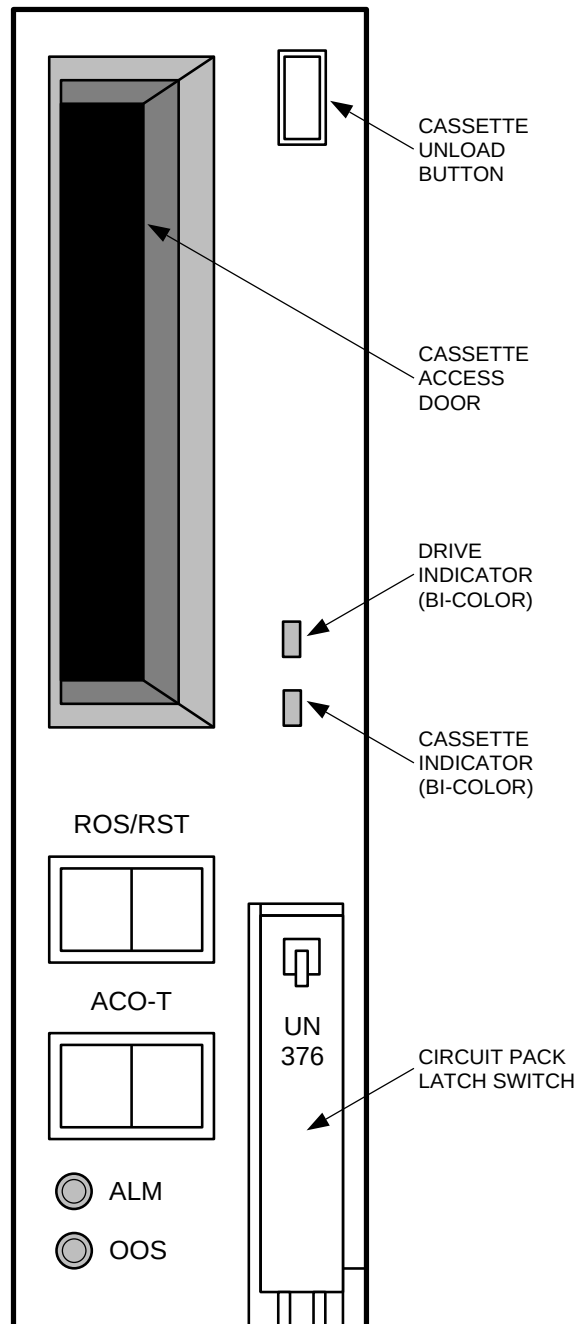


Figure 2-4. 3B21D Computer UN376 (CU Power Switch) Faceplate

Procedure to Respond to an Alarm

When an alarm is generated, the operator should take the following action:

- Note the cause and type of alarm from the display page, if possible retire the alarm from the MTTY by pressing **ALM RLS**, and read the ROP output.
- Look up the output message printed by the ROP in one of the following IPs:
 - 254-303-111, *PDS Output Messages Manual*
 - 254-303-113, *MML Output Messages Manual*.

Determine the possible cause of the alarm and if any additional data should be collected and forwarded to your technical assistance organization.

- If the alarm is caused by a reinitialization or recovery, do nothing. Refer to Chapter 3, "Recovery."
- If the alarm is caused by an interrupt, refer to Chapter 4, "Error Analysis."
- If the alarm is caused by a diagnostic failure, refer to Chapter 6, "Diagnostics," and to 254-303-102, *Trouble Clearing Tasks*.
- If the alarm is reported by an audit, refer to Chapter 14, "System Audits." Information on audit execution is reported via the AUD output messages. Typical AUD output messages are as follows:

AUD CUMEM
AUD CUSTAT
AUD ECD
AUD ECDOWN
AUD FMGR
AUD FSBLK
AUD FSCMPT
AUD FSLINK
AUD MMGR
AUD MSGBUF
AUD PMS
AUD PROAD

- If the alarm is a hardware alarm, determine the unit involved from the Common Processor Display Page or the console printer output.

When a hardware fault is detected, an output message is printed on the console printer. Typical generic hardware fault output messages and an indented specific example of each are as follows:

```
REPT FAN ALARM c CABINET
  REPT FAN ALARM BASIC CABINET
  REPT FAN ALARM GROWTH CABINET

REPT FAN MULTIPLE FAILURE a b
  REPT FAN MULTIPLE FAILURE CU 1

REPT FAN SINGLE FAILURE a b
  REPT FAN SINGLE FAILURE CU 1

REPT IOP a PC COMMUNITY j POWER REMOVED
  REPT IOP 0 PC COMMUNITY 2 POWER REMOVED

REPT POWER REMOVED a b [c d]
  REPT POWER REMOVED IOP 0

REPT POWER FAULT a b [c d]
  REPT POWER FAULT IOP 0

REPT POWER DISTRIBUTION FRAME e f ALARM
  REPT POWER DISTRIBUTION FRAME MINOR ALARM
  REPT POWER DISTRIBUTION FRAME MAJOR ALARM
```

- Using the ACO/T switch at the hardware unit power switch, retire the alarm.
- Inspect the unit for blown fuses and lighted LEDs on power units and cooling units.
- Replace the faulty hardware causing a blown fuse, replace the fuses, and power up the unit by operating the ON switch on the unit power switch. If the cooling unit is the cause of the alarm, inspect the filters before replacing the cooling unit. Fan failure, caused by overheating as a result of clogged filters, will cause an alarm.
- With the ROS/RST switch having been left in the RST position, unit diagnostics will be automatically requested. When the diagnostics pass, the unit will be restored to service.
- If the diagnostics fail, additional trouble clearing will be required.

If you are using a 3B20D Model 1 computer, refer to 254-301-812, *Control Unit Trouble Clearing*, and 254-301-813, *Peripheral Equipment Trouble Clearing*, for additional procedures.

If you are using a 3B20D Model 2 or 3 computer, refer to 254-302-832, *Trouble Clearing (PDS Format)*, or 254-302-842, *Trouble Clearing (MML Format)*, for additional procedures.

If you are using a 3B21D computer, refer to 254-303-102, *Trouble Clearing Tasks*, for additional procedures.

Documentation

You should be familiar with the other documents available in the 3B20D/3B21D computers documentation set.

Using Automatically Generated Data for Troubleshooting

When certain faults occur, the 3B20D/3B21D computers generate messages on the ROP and writes to files kept on the disk. In addition, a series of on-line readings and plant measurements is kept and printed.

Study this data every day to verify the stable operation of the machine. Items of interest are as follows:

- Log Files — These files contain any information generated that applies to system fault recovery actions. You must be aware of any entries in these log files and take action to correct the cause of any problems. (Refer to Chapter 8, “Log File Review.”)
- ROP Printout — This printout may indicate abnormality in system performance.
- Plant Measurement Data — Data printed on the ROP contains information on numbers of recovery actions, alarms, unit downtime, and application performance. Investigate any abnormality. The report can be generated using the **OP:PMCR** command.
- Error Files — These files contain failure data generated by some processes. If recent change and verify fails, error codes are written to the */etc/rcvecd.err* file. Additional failure information for Equipment Configuration Database (ECD) audits is written to one of the following files:

/etc/log/ecdaud_disk

/etc/log/ecdaud_struc

/etc/log/ecdaud_raw

Switching Control Center Interface

The 3B20D computer is equipped to interface with an SCC. When maintenance is shared between the SCC and the computer site, observe the following guidelines:

- Coordinate all maintenance activities. The SCC has the same terminal access to the computer as the on-site maintenance terminal. Both can select boot time parameters, request diagnostics or phases, and so forth. Without close coordination, both could perform different functions simultaneously with unpredictable results.
- Input all messages from the remote site or use local procedures to eliminate possible simultaneous messages being input.

Training

Installation and maintenance personnel training courses are available at local Lucent Technologies National and Regional Training Centers strategically located throughout the United States. These courses cover most aspects of installing and maintaining 3B20D/3B21D computers. Also, similar training courses for managers are available.

Field Support

The 3B20D/3B21D computer field sites are supported by a hierarchy of technical assistance organizations. Problems are referred to support organizations until the correct expertise is found to solve the problem. Know the correct procedures for your site.

Contents

Introduction	3-1
Responding to Automatic Recovery Actions	3-1
■ Initialization Causes	3-3
■ Recognizing an Initialization	3-3
■ Post Initialization Analysis	3-4
■ Initialization and Recovery Analysis Guidelines	3-6
Recovering the System Manually	3-8
■ Severity of Recovery Options	3-8
■ Recovery Options	3-9
Forced Configuration Options	3-9
Inhibit Options	3-11
Initialization Options	3-12
Bootstrap Options	3-13
Powerdown Options	3-13
Software Backup Disk and Alternate Boot Disk Options	3-15
Force Disk Selection Options	3-16
DIOP Options	3-18
Faulty Unit Repair	3-18
System Restoration	3-18
■ Recovery Procedure	3-19
Recovery from Backups Procedure	3-25
Application Recovery Procedure	3-31
Post Recovery Procedure	3-32
Using Processor Recovery Messages	3-33
Responding to a Terminal Suspend	3-33

Contents

■ One or More Terminals Faulty	3-33
■ All Terminals Do Not Function	3-34
■ Power Interrupt (Fiber Optic Feature Only)	3-34
■ Non-MTTY Suspend	3-36
■ MTTY Suspend	3-37
KS-23554, L5 and L6 Video Terminal	3-37
KS-23996, L1 and L5 Video Terminal (3B21D Computer Only)	3-38
KS-22921 Video Terminal	3-39
Teletype Model 4424 Terminal	3-40
■ Receive-Only Printer Suspend	3-42
■ Craft Initialization	3-42
■ Port Switch	3-42
■ Creating a Dump Tape	3-43
Isolating Emergency Action Page Problems	3-45
Recovery Using Software Backup Disk and Alternate Boot Disk Options	3-51
■ SBD Switching Identities Procedure	3-55
■ Recovery Using Alternate Boot Disk	3-56
Recovering from DIOP	3-59
■ Verify DIOP Mode in Effect Procedure	3-60
■ Prepare DFCs and MHDs for Recovery Procedure	3-63
■ Load Disk from Tape Procedure	3-67
■ Recovery Decision Procedure	3-70
■ Boot Procedure	3-74
■ SMD DFC Repair Guidelines	3-77
■ SMD MHD Repair Guidelines	3-79
■ SCSI DFC Repair Guidelines	3-81
3B20D Computer	3-81
3B21D Computer	3-83
■ SCSI MHD Repair Guidelines	3-84
Loading Disk from Tape	3-85
■ Loading Single Volume Tapes (Formerly the Standard LDFT Procedure)	3-86
■ Loading Multivolume Tapes	3-90

Introduction

The following sections describe the 3B20D/3B21D computer recovery procedures:

- Responding to Automatic Recovery Actions
- Recovering the System Manually
- Using Processor Recovery Messages
- Responding to a Terminal Suspend
- Isolating Emergency Action Page Problems
- Recovery Using Software Backup Disk and Alternate Boot Disk Options
- Recovering from Disk Independent Operation (DIOP)
- Loading Disk from Tape.

Responding to Automatic Recovery Actions

When a hardware or software fault of sufficient severity occurs (severe enough to prevent normal processing), the *UNIX*® Real-Time Reliable (RTR) operating system takes automatic action to recover.

Several levels of initialization may occur within the system. These levels are caused by repeated calls for a system phase and system reconfiguration. Table 3-1 lists the initialization levels. The 3B20D/3B21D computers attempt to initialize and recover through progressive initializations or “phases” up to and

including level 3. Beyond level 3, there must be a manual recovery action or a special request from the Application Integrity Monitor (AIM). The hardware sanity timer, error registers, and software program requests drive this process.

Automatic recoveries are grouped into three areas:

- Automatic system initialization (Phase 1, 2, or 3)
- Maintenance interrupts (initialization without memory copy or bootstrap)
- Error interrupts (errors processed by software).

The following sections discuss the sources, the indicators that are present during the action, and the response you should make.

Table 3-1. Initialization Levels

Level	Description – System Response	Type Auto vs. Manual
1	Initializes hardware; rolls back the software to a safe restart point.	Auto/manual
2	Boots operating system from disk using the incore Equipment Configuration Database (ECD) and saves write-protected application segments.	Auto/manual
3	Boots operating system and ECD.	Auto/manual
4	Clears Protected Application Segments (PAS) and boots the operating system and ECD.	Manual†
5*	LDFT: Loads selected disk with new release from tape unit 0, IOP 0 [or Small Computer System Interface (SCSI) tape device connected to DFC 0‡].	Manual
6*	LDFT: Loads selected disk with new release from tape unit 1, IOP 1 (or SCSI tape device connected to DFC 1‡).	Manual

* These initialization levels also require a level 4 (54) boot after the disk has been loaded.

† The AIM process can request a level 4 initialization.

‡ The 3B20D computer can only Load Disk from Tape (LDFT) using the Input/Output Processor (IOP) tape units. The 3B21D computer uses the SCSI tape units on the Disk File Controller (DFC).

Initialization Causes

The operating system initializes and attempts to recover itself when one of the following occurs:

- The software detects a condition requiring an initialization and requests a phase.
- The sanity timer exceeds a measured interval.
- The hardware self-checking circuitry detects a certain type of problem.
- Interrupt thresholds have been exceeded.

Some problems cause a stop-and-switch [one Control Unit (CU) halting and the system switching to the other]. The operating system phase strategy is progressive. Repeated calls for a system phase result in increasingly severe phase levels. Switches between primary and secondary disk are called if initializations reach level 2 or above. In addition to the system-generated initializations, application kernel processes may request an initialization based on their own internal criteria.

Recognizing an Initialization

When an initialization occurs, observe one or more of the following:

- Audible alarm from the Maintenance Teletypewriter (MTTY).
- Flashing **CRITICAL** indicator in the status summary area.
- Audible system alarm.



NOTE:

Audible and visual alarms may vary with the application.

- Flashing **RCVRY** message to the right of the CU status on the Emergency Action Page.
- Changing Processor Recovery Message (PRM) codes on the Emergency Action Page and on the Receive-Only Printer (ROP).

- The start of a system initialization is usually signaled by one of the following messages being output on the ROP:
 - REPT CU 0 MAINTENANCE INTERRUPT
 - REPT CU 1 MAINTENANCE INTERRUPT
 - REPT PHASE IN PROGRESS
 - START OF CU 0 RECOVERY
 - START OF CU 1 RECOVERY.
 - For the 3B20D computer only, display on the Emergency Action Interface (EAI) circuit pack on the initializing CU stepping from 0 through C. (All digits may not be visible because of timing and speed of the display update.)
- ⇒ **NOTE:**
If a bootstrap of the system (3B20D computer only) is unsuccessful, the indicator will not progress beyond the failing step in the bootstrap sequence.
- Postmortem dump prints on the ROP after the system gains sanity.

Post Initialization Analysis

Whether the system switched processors, bootstrapped, or initialized the hardware, post recovery action is necessary to correct the problem that caused the initialization.

Post initialization analysis after a successful automatic recovery is usually confined to four types of output:

- Processor recovery messages
- Postmortem dump
- Diagnostics
- Output messages.

When the processor regains sanity, perform the following procedure:

Step	Procedure
1	Check the PRMs on the ROP output. These messages will not pinpoint trouble but will point to a general area. A breakdown of PRMs is covered in 254-303-103, <i>Processor Recovery Messages Guide</i> .

Step	Procedure
2	Analyze the postmortem dump to determine the source and level of initialization. Additional procedures for postmortem dump analysis may be found in Chapter 4, "Error Analysis."
3	To assist your technical assistance organization in finding and correcting causes of software initializations and other system anomalies, you may want to make an off-line memory dump to tape when a level 2 (or higher) bootstrap occurs. (Refer to the "Creating a Dump Tape" section.) Capture a memory dump to tape before the off-line CU is diagnosed and restored to standby. To prevent a restore from occurring automatically, use the EAI options to "force active on-line" (Poke 12) and inhibit diagnostics (INH:DMQ). Refer to the "Creating a Dump Tape" section for additional instructions.
4	In most cases, a memory dump to tape is not necessary and diagnostics should be allowed. The system should be brought up to active-standby as soon as possible. Check and ensure that automatic diagnostics are <i>NOT</i> inhibited. This may be verified by entering the OP:DMQ command. ⇒ NOTE: Even though diagnostics are requested by the system, they are suspended until the system has recovered.
5	Check for units removed from service. This may be verified by entering the OP:OOS command.
6	If diagnostics are not requested, begin the process by entering the RST: command with the Trouble-Locating Procedure (TLP) option. ⇒ NOTE: At the completion of the diagnostics, a summary table is printed. Units that fail are marked Some Tests Failed (STF), and a TLP circuit pack list is produced. A detailed explanation of automatic and manual diagnostics is given in Chapter 6, "Diagnostics."

Initialization and Recovery Analysis Guidelines

Prior to an initialization occurring, develop a Recovery Plan using the following guidelines:

- Craft persons and technical support personnel should be familiar with these and other switching product recovery procedures and manuals. The first time to review 254-303-103, *Processor Recovery Messages Guide*, and this IP is not during an outage.
- Stored office backup tapes should include the following complete sets (copies of all essential file systems and data partitions):
 - The original retrofit tapes. These are the oldest tapes needed to be saved. When these guidelines refer to tapes older than the retrofit tapes then ignore those references.
 - More backup tapes should be made immediately after any successful system boot or growth. These should be saved until the next retrofit.
 - Quarterly backups for the last 3 quarters.
 - Monthly backups for the last 3 months.
 - Weekly (or consistent with the recent-change and software update schedule) backups for the last three sessions.
 - At least two complete sets of backup tapes should be located off site. One should be at least 2 months old and one should be no more than 1 month old.
 - Backup tapes are sometimes made as part of growth procedures. These should be made in addition to normally scheduled backups which will ensure two copies of backups containing major hardware configuration changes.
- Spare Moving Head Disk (MHD) circuit packs should be converted into "cold spare" disk backup using the "Making a Backup Disk" procedure in Chapter 9.
- Use of "warm spare" (Software Backup Disks) and "cold spare" (shelf copies of system disks) MHDs **DO NOT** eliminate the need for tape backups.
- Backup tapes should be verified after creation (VFY:TAPE) to ensure integrity.
- Each office should review recovery procedures annually.

When an initialization is occurring or has occurred, observe the following guidelines:

- Give the system time to recover itself. Check all indicators before intervening. The system will continually attempt to recover itself either through escalation or through “rolling phases.”
- Note the date and time of initialization.
- Obtain all printouts of all messages automatically generated by the system.
- Analyze the information, using the following:
 - 254-303-111, *PDS Output Messages Manual*
 - 254-303-113, *MML Output Messages Manual*
 - 254-303-103, *Processor Recovery Messages Guide*
 - Chapter 4, “Error Analysis,” in this IP.
- Use the **OP:LOG** command to dump the PMLOG file for the specified time and date.
- Inform the appropriate technical assistance organization if further analysis is required.

Recovering the System Manually

The 3B20D/3B21D computers restore normal processing through automatic initialization and recovery actions. However, if the processor cannot initialize and recover using automatic recovery, manual intervention is required.

While reviewing this section, refer to the “How to Access the Emergency Action Pages” section and the EAI Page figures in Chapter 12, “Craft Interface.”

To decide what manual actions are needed, observe the system to determine the extent of system insanity and whether previous actions (for example, forced configurations) prevented automatic recovery. This observation should answer the following questions:

- Are any computer units powered down?
- Are there any computer units with blown fuses?
- Are disk drives either powered down or not spinning?
- Are “force” commands on the Emergency Action Page active?
- Are inhibits activated?

An affirmative answer to any of these questions indicates that the computer could not attempt an automatic recovery using all possible processor configurations. Before taking any manual action, eliminate these and similar conditions.

Severity of Recovery Options

The recovery options available vary in severity; the time required to recover the system varies from a few hundred milliseconds in an initialization to several hours in an LDFT. The recovery options are ordered, from least severe to most severe, as shown in the following list:

- Force configuration options (least severe) initialization
- Bootstrap options
- Back up root bootstrap options
- Inhibit options
- Minimum configuration boot options
- Load disk from tape options (most severe).

Recovery Options

Manual recovery actions are categorized as follows:

- Forced configuration options
- Inhibit options
- Initialization options
- Bootstrap options
- Power down options
- Recovery using Software Backup Disk or Alternate Boot Disk (if equipped)
- Force Disk Selection (FDS) options (*UNIX* RTR Operating System Release 6.4 and Release 21.5 and later)
- Recovery from backup tapes (LDFT options are described in section "Loading Disk from Tape").
- DIOP options.

Forced Configuration Options

The forced configuration options (available on the Emergency Action Page) include the following:

- (1) Force CU on-line (Command 10 ,11) – Prevents any CU other than the CU forced from recovery. Prevents the CUs from switching active and standby roles to achieve a stable configuration using automatic recovery mechanisms.
- (2) Force CU to use either primary or secondary disk subsystem (Command 20, 22) – The option forces the use of a particular disk to allow the system to achieve a stable configuration. The primary disk for CU 0 is MHD 0 and the primary disk for CU 1 is MHD 1 unless the ABD feature is being used [see item (6) Alternate Boot Disk (Command 28) description that follows]. The CU must be forced on-line. Refer to the table in Figure 3-1.
- (3) Clear CU forces (Command 13) – Clears a forced configuration of the CUs.
- (4) Back up *root* file (Command 30) – Forces the processor to boot using the backup file partitions on the disk. This option is activated during a manual boot and remains in effect during subsequent automatic bootstraps until the option is manually removed and the system is manually rebooted.



CAUTION:

When a successful recovery has been completed on the backup root file system on a partially restored disk, the following action must be performed to certify the disk integrity. This is mandatory in the described configuration. Failure to follow these actions may jeopardize future system recovery.

- (a) Manual and automatic disk restores must be inhibited to prevent populating data onto other disks (**INH:DMQ:SRC=ALL**).*
 - (b) All file systems must be checked for consistency using the **FSLINK** and **FSBLK** audits. File partitions should be checked for correct hashsums using the **OP:STATUS:SUM** or **VFY:FILE** command.*
 - (c) Contact the next level of support to aid in assessing the data integrity of the currently active disk.*
-
- (5) Minimum configuration (Command 32) – Isolates the computer from the application when failing PRMs indicate that the application is preventing the processor from recovering. Application-specific recovery actions are required to recover the system. This option is activated during manual boot and remains in effect during subsequent automatic bootstraps until the option is manually removed and the system is manually rebooted.
 - (6) Alternate Boot Disk (Command 28) - Allows the craft personnel locally or at the SCC to select an alternate boot disk to be used during the next boot. The disk selected must be a Software Backup Disk (a warm spare image of the system disks). This feature is only available with the UN597 Maintenance Teletypewriter Controller (MTTYC) circuit pack used on the 3B21D computer.
 - (7) Force port switch – Forces the MTTYs and ROP to communicate with a specific IOP. The port switch is manually forced using a switch on the port switch circuit pack.

Inhibit Options

The inhibit options disable selected hardware and software functions. If the system can be recovered while these inhibits are active, the processor can then provide some degraded services. Use these options only as a last resort effort to recover a system with faults in both CUs.

The inhibit options (available on the Emergency Action Page) include the following:

- Inhibit Sanity Timer (Command 24) – Prevents a processor stop-and-switch or Maintenance Reset Function (MRF) when the sanity timer times out. This inhibit option is activated immediately after the command is entered. To remove this inhibit, enter EAI poke **25**.
- Inhibit Cache (Command 40) – Prevents the processor from using the cache memory. This inhibit option results in a slowdown of all processor operations. An initialization is required to activate this option. To remove this inhibit, enter EAI poke **41** followed by an Init (EAI poke **51**).



NOTE:

The following inhibits can be added or removed by command numbers followed by an initialization or by input messages. The command numbers add or remove the inhibits for the next initialization. One command number to remove an inhibit requires a level 3 or 4 initialization. Input messages add or remove the inhibits without an initialization, but they do not affect the inhibits for the next initialization (command numbers).

- Inhibit Software Check (Command 36) – Inhibits the System Integrity Monitor (SIM) process from phasing the system. An initialization is required to activate this option from EAI. An initialization is required to add or remove this inhibit via EAI commands.
- Inhibit Error Interrupt (Command 38) – Prevents the system from processing any error interrupts. Use this inhibit option if repeated interrupts are causing stop-and-switch action on both CUs. An initialization is required to add this inhibit via EAI command. Since this inhibit affects records in the database, a level 3 or 4 initialization is required to remove this inhibit via EAI command. The **ALW:ERRINT** command should be used to allow error interrupts followed by the EAI poke **39**. This sequence of commands (**ALW:ERRINT** followed by EAI poke **39**) will avoid a level 3 boot to clear this inhibit.
- Inhibit Hardware Check (Command 34) – Prevents stop-and-switch and on-line MRF caused by hardware errors. An initialization is required to add or remove this option via EAI commands.



CAUTION:

Normal system operations and fault recovery capabilities are compromised when any of the previously listed inhibits are in effect. Use these options when necessary for system recovery, but clear them as soon as possible.

Initialization Options

The initialization options (available on the Emergency Action Page) are also useful in system recovery. The options include the following:

- Application Initialization (Command 50) – Signals the application software to initialize to a level indicated by the *APPL-PARAM* on the Emergency Action Page. Refer to the application initialization documentation for use of this feature.
- Initialization (Command 51) – Initializes the processor but does not bootstrap the system. The level of initialization is the same as that of a maintenance interrupt or a Phase 1.
- Craft Interface Initialization (Command 15) – Initializes the craft interface in the active CU. Use this option when all system functions except the MTTY/TTY appear to be working. (Refer to the “Responding to a Terminal Suspend” section.)



NOTE:

There are three levels of the craft initialization (15). Each level requires a parameter provided by first poking the 42 on the Emergency Action Page. The parameters are 1, 2, and 3 for each level, respectively.

- Level 1 restarts both Maintenance Teletypewriter Controllers (MTTYCs).
- Level 2 adds the restart for all TTYCs.
- Level 3 reinitializes all IOPs with their subunits.

Bootstrap Options

The bootstrap options (available on the Emergency Action Page) are used to load system software. The options include the following:

- Initialization (Command 51) – Initializes the processor but does not bootstrap the system. The level of initialization is the same as that of a maintenance interrupt or a Phase 1.
- BOOT (Command 52) – Reloads programs from disk to main memory and executes them. The ECD and protected memory segments remain intact.
- BOOT + ECD (Command 53) – Bootstraps the operating system and loads the ECD. Protected memory segments are kept intact.
- BOOT + MEM (Command 54) – Clears PAS, boots the operating system, and loads the ECD.

Powerdown Options

As a final action to recover the system from hardware failures (especially where lightning strike may be involved), perform the following procedures to selectively remove power from the 3B20D/3B21D computers duplex units (CU, DFC, and IOP):

Step	Procedure
1	Use the EAI force options to force CU 0 (poke 10 followed by y) and primary disk subsystem (poke 20).
2	Use the manual port switch options to select ROP 0 and MTTY 0.
3	Power down all odd units: CU 1, DFC 1, IOP 1 plus IOP 3 or DFC 3, if equipped, and stand-alone 9-track tape drives connected to odd-numbered IOPs and DFCs.
4	From the EAI, select the 54 bootstrap option. (If problems exist with the EAI, power up IOP 1, power down IOP 0, and use the manual port switch options to select ROP 1 and MTTY 1.)
5	Observe the PRMs. Make a note of any changes.
6	If the boot is not successful, the message "START OF CU-0 RECOVERY" will be repeated on the ROP.



NOTE:

There may not be a failing PRM (PRM 0 F ...).

If the system recovers, go to Step 22. If the bootstrap is not successful, continue to the next step.

Step	Procedure
7	Restore power to DFC 1. Restore power to DFC 3, if equipped. Power down DFC 0. Power down DFC 2, if equipped, and stand-alone 9-track tape drives connected to even-numbered DFCs.
8	Power down IOP 0. Power down IOP 2, if equipped, and stand-alone 9-track tape drives connected to even-numbered IOPs. Power up IOP 1.
9	Use the manual port switch options to select ROP 1 and MTTY 1.
10	Use the EAI options to force CU 0 secondary disk subsystem (EAI poke 22).
11	From the EAI, select the 54 bootstrap option.
12	Observe the PRMs. If the system recovers, go to Step 22. If the bootstrap is not successful, continue to the next step.
13	Power down CU 0.
14	Use the EAI force options to force CU 1 (poke 11 followed by y) and primary disk subsystem (poke 20).
15	Power up CU 1.
16	From the EAI, select the 54 bootstrap option. (If problems exist with the EAI interface, power up IOP 0, power down IOP 1, and use the manual port switch options to select ROP 0 and MTTY 0.)
17	If the boot is not successful, the message "START OF CU-1 RECOVERY" will be repeated on the ROP.
	 NOTE: There may not be a failing PRM (PRM 1 F ...).
	If the system recovers, go to Step 22. If the bootstrap is not successful, continue to the next step.
18	Restore power to DFC 0. Restore power to DFC 2, if equipped. Power down DFC 1. Power down DFC 3, if equipped.
19	Use the EAI options to select secondary disk (EAI 22) and request a 54 bootstrap.

Step	Procedure
20	Observe the PRMs. Report any changes to your next level of technical support. Inform them that this power isolation procedure has been completed.
21	If the boot attempt is not successful, the powered down IOP, DFC, and CU should be powered up.
22	If the system recovers at any time during the "Powerdown Options" procedure, seek technical assistance before powering up the powered down units. Keep in mind that either CU 0 or CU 1, either DFC 0 or DFC 1, and either IOP 0 or IOP 1 must be powered up in order for the system to run with a maintenance terminal working. In some offices either DFC 2 or DFC 3 are required for full system operation.



NOTE:

A similar power down sequence can be done to isolate interference problems from application duplex devices (that is, DCIs, APIs, etc.) connected to the 3B20D/3B21D computers via the Dual Serial Channel (DSCH) interface.

Software Backup Disk and Alternate Boot Disk Options

This section provides information on the Software Backup Disk (SBD) and the recovery from Alternate Boot Disk (ABD) feature.

The 3B20D/3B21D computer system equipped with extra disk drives (usually MHD 14 and/or 15) that are backups of the system disks (MHD 0 and 1) can be used to manually recover an office with either a duplex system disk hardware failure or to recover by backing up to a previous copy of the system software before an error has been introduced. Disks specifically configured and used in this manner are called software backup disks and sometimes referred to as warm spare drives.

The ABD feature provides the capability to recover a system from a remote monitoring station, such as a Switching Control Center (SCC), using a disk other

than MHD 0 or MHD 1 on DFC 0 or DFC 1, respectively. The ABD feature extends the bootstrap capability to MHD 2 through MHD 15 and requires the following MHD device ID assignment scheme:

Device ID #	DFC 0		DFC 1	
	SCSI Bus 0	SCSI Bus 2	SCSI Bus 1	SCSI Bus 3
0	MHD 0	MHD 2	MHD 1	MHD 3
1	MHD 4	MHD 6	MHD 5	MHD 7
2	MHD 8	MHD 10	MHD 9	MHD 11
3	MHD 14	MHD 12	MHD 15	MHD 13

With the ABD feature, trained personnel will no longer have to be present on site to recover an office using the SBD. The personnel at the SCC can select the SBD as the alternate boot disk and initiate a recovery from the EAI Page without having to manipulate the MHD hardware.

The ABD feature is an optional feature that is implemented in RTR Release 21.27 and later software releases and requires a 3B21D computer system equipped with a pair of UN597 MTTYCs. Refer to the MTTYC ABD Conversion chapter in 254-303-100, *Growth/Retrofit Tasks*, for procedures to convert from a UN583 MTTYC to a UN597 MTTYC or to convert a UN597 without the ABD feature to UN597 with the ABD feature. The ABD feature is not supported on the 3B20D computer, AM Work Station (AWS) or the Very Compact Digital Exchange (VCDX) platforms.

The primary benefit of the SBD and the ABD feature is reducing downtime during an outage. During a recovery scenario where the system needs to be recovered from backup, the use of SBDs can reduce the amount of downtime by avoiding the time to load the disk from tape. The ABD feature further reduces the downtime by eliminating the need for immediate on-site workforce specially trained to perform disk subsystem maintenance activity otherwise required to use the SBD.

If an error occurs that is undetected and propagated to the SBD, it is most likely also propagated to the backup tapes. As such, there is no benefit from having two SBDs of the exact same backup image. The age of the error cannot usually be determined; therefore, it is recommended that if two or more SBDs are equipped that one is used for routine backups and the others updated only after the system boots (or after successful completion of an off-line boot test, see Chapter 7, "Off-Line Boot").

For details using Software Backup Disk options for recovery, refer to the "Recovery Using Software Backup Disks" section.

Force Disk Selection Options

The FDS feature, which is available in *UNIX* RTR Operating System Release 6.4 and Release 21.5 and later, adds the capability of selecting application disks to

be restored at bootstrap. The feature selects which MHD of a duplex pair will be made active during the bootstrap. The system recovers with only one MHD of an active duplex pair.

The following FDS options are available:

- A straight selection – All duplex application disks on the same DFC are selected to be the active system boot disk.
- A cross-selection – All duplex application disks on the opposite DFC are selected to be the active system boot disk.

The selected disks would then be restored (made active) during bootstrap.

⇒ NOTE:

This feature affects only the duplex disks on DFC 0 and DFC 1.

The feature is activated by changing the *fds_activate* field (field 26 in Release 6.4 and field 27 in Release 21.5 and later) of the *btparm* form to “y” in the disk copy of the ECD. This field is set to “n” by default.

Once the feature is activated, the application disks are selected by the following procedure:

1. Select the boot disk by forcing a CU on-line (Command 10 or 11) and forcing a primary or secondary disk (Command 20 or 22).
2. Choose a straight or a cross-selection using the appropriate application parameter. Enter Command 42, “S” or “s” for straight selection, or Command 42, “X” or “x” for a cross-selection.
3. Boot at a level that loads the ECD from disk (Command 53 or 54).

Once the application disks are selected, no automatic disk reconfigurations will be done during bootstrap. If a selected disk cannot be restored, the mate MHD will NOT be tried. If a selected MHD is marked as essential in the ECD [field 5, “essential,” of the MHD Unit Control Block (UCB), set to “Y” or “ALWAYS”] and cannot be restored, the system will not recover. A “rolling phase” will occur. The only way to recover from the rolling phase is to clear the application parameter (Command 43) and boot (Command 53 or 54). If a selected disk is not marked as essential in the ECD and cannot be restored, the system will recover with neither MHD of the duplex pair active.

If a straight or a cross-selection is not made when booting, the default disk recovery strategy will be followed. The default strategy is to first attempt to restore the MHD of the duplex pair that was active before the bootstrap. If both or neither were active, then the MHD on the boot DFC will be tried first. If the first disk tried cannot be restored, the associated mate MHD will be restored.

DIOP Options

Faulty Unit Repair

If operating system disks are in DIOP, the following menu commands are available and can be initiated from the control/display portion of the MTTY and the Switching Control Center (SCC). (Refer to Chapter 12, "Craft Interface.")

- Format a disk.
- Verify tracks on a disk.
- Perform a special DIOP restore for a DFC.
- Output defect management tables for fixed media disks.
- Perform a special DIOP restore for a Small Computer System Interface Bus (SBUS).
- Load a disk from tape.

If the operating system disks are in DIOP, only the previously listed commands are available; diagnostics are not available.

System Restoration

When the faulty units have been repaired, system restoration can take place. The system must be booted selecting the appropriate boot device. All repairs must be complete before a bootstrap can be attempted. Performing a bootstrap before repairs are complete may cause system outage.

When the system goes into DIOP, a flag word is written into the writable microstore to indicate this state. While this flag is set, the system will honor only a manual request to bootstrap.

All of the writable microstore is reloaded from disk on every bootstrap. If the manually initiated bootstrap succeeds in reloading the writable microstore but the bootstrap attempt is not successful, the system would be out of DIOP.

The unsuccessful bootstrap attempt may have corrupted the main store, so system sanity may be lost. The following procedure must be followed for successful system restoration:

1. Select desired configuration forcing the repaired disk to be accessed on bootstrap.
2. Verify repaired devices are powered up.
3. Verify repaired devices are properly forced.
4. Perform a level 3 bootstrap (Emergency Action Page, Command 53).

Recovery Procedure

This procedure is used to determine whether manual intervention is required to recover the system and if so, to direct the craft person to the proper recovery procedures. If manual action is found to be necessary, the craft person should immediately follow local escalation procedures to contact the Lucent Technologies Regional Technical Assistance Center (RTAC) or Local Field Support (LFS), the application (this switching product) Customer Technical Support (CTS) or Development Field Support (DFS), and the Electronic Switching Action Center (ESAC) while continuing to follow the recovery procedures. The CTS will provide additional support, if necessary, to resolve the problem.

UNIX RTR operating system (3B20D/3B21D computer) initializations can be caused by failures in 3B20D/3B21D computer hardware, application hardware [usually those units connected via the DSCH link such as Common Network Interface (CNI) hardware], *UNIX* RTR operating system software, or application software. If manual intervention is necessary, analysis of the failure must be performed in order to select procedures that will recover the system while minimizing call processing downtime.

The key to this analysis is the information provided by Processor Recovery Messages (PRMs). A failure PRM printed at the beginning of every initialization attempt identifies the failure source. If you are not familiar with PRMs, refer to 254-303-103, *Processor Recovery Messages Guide*, before proceeding.

When looking for specific PRMs on the ROP, fields shown in **bold** print in this procedure should exactly match the corresponding PRM on the ROP. Other fields may contain different values.

Except where otherwise noted, all commands are Emergency Action Page commands.

Step	Procedure
1	Record the approximate time at which the first automatic 3B/RTR initialization occurred.
2	At the MTTY Receive-Only Printer (ROP), mark the occurrence of the first initialization on the ROP output.
3	At the MTTY ROP, look for the most recent failure PRM (first hexadecimal digit = F) which has the following format: PRM_a Fbcd xxxx xxxx xxxx xx xx xx

Step	Procedure
4	Was a failure PRM found in the previous step with field 'b' equal to E? If YES, application hardware or software is implicated. Go to the "Application Recovery Procedure" section and then continue. If NO, 3B20D/3B21D computer hardware or <i>UNIX</i> RTR operating system software is implicated. Continue to the next step.
5	Ensure all essential units are powered up including central office frames and equipment not part of the 3B20D/3B21D computer frame(s). Check power and fuses to the Moving Head Disks (MHDs), Disk File Controllers (DFCs), and Control Units (CUs).
6	If any of the units listed in Step 5 are powered down, restore power to the units in accordance with the local procedure.
7	If any of the units listed in Step 5 have blown fuses, replace the blown fuses.
8	Ensure all EAI forces and inhibits are clear by entering the following EAI poke: 14 (CLR EAI)
9	Are 3B20D/3B21D computer initialization failures still occurring (repeated new initializations and/or failure PRMs)? If YES, continue to the next step. If NO, go to the "Post Recovery Procedure" section and then continue.
10	Ensure highest level recoveries are being attempted by entering the following EAI pokes: 54 followed by y
11	Are 3B20D/3B21D computer initialization failures still occurring (repeated new initializations and/or failure PRMs)? If YES, continue to the next step. If NO, go to the "Post Recovery Procedure" section and then continue.

Step	Procedure
12	Attempt recovery using backup file-systems by entering the following EAI pokes: 30 followed by y
13	Ensure highest level recoveries are being attempted by entering the following EAI pokes: 54 followed by y
14	Are 3B20D/3B21D computer initialization failures still occurring (repeated new initializations and/or failure PRMs)? If YES , continue to the next step. If NO , go to the "Post Recovery Procedure" section and then continue.
15	Immediately follow local escalation procedures to contact your next level of technical support. Escalation should continue until RTAC or LFS and the application (this switching product) CTS or DFS organizations have been contacted. The Lucent Technologies field support engineers will provide additional support, if necessary, to resolve the problem.
16	At the MTTY ROP, look for the most recent failure PRM (first hexadecimal digit = F) or last successful (prior to START OF CU X RECOVERY message) PRM (first hexadecimal digit = E) which has the following format: PRM_a Fbcd xxxx xxxx xxxx xx xx xx or PRM_a Ebcd xxxx xxxx xxxx xx xx xx
17	Clear all forces and "Backup Root" by using the following EAI poke: 14 (CLR EAI)
18	Attempt to recover on CU 0: a. Force primary boot device by entering the following EAI poke: 20 b. Force CU 0 by entering the following EAI pokes: 10 followed by y

Step	Procedure
	c. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y d. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step. e. Attempt to recover the system using new configuration CU 0 with MHD 1 by entering the following EAI poke: 22 f. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step.
19	Attempt to recover on CU 1: a. Force primary boot device by entering the following EAI poke: 20 b. Force CU 1 by entering the following EAI pokes: 11 followed by y c. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y d. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step. e. Attempt to recover the system using new configuration CU 1 with MHD 0 by entering the following EAI poke: 22

Step	Procedure
	f. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step.
20	Quickly determine if all configurations result in the same failure. Select the most successful hardware configuration. If they are the same, select CU 0 and MHD 0 (EAI pokes 20 , and 10 followed by Y).
21	Attempt to recover using Inhibit Options: a. Inhibit Sanity Timer by entering the following EAI pokes: 24 followed by y b. Inhibit Software Checks by entering the following EAI poke: 36 c. Inhibit Hardware Checks by entering the following EAI poke: 34 d. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y e. Record failing PRM or last successful PRM.
22	Are 3B20D/3B21D computer initialization failures still occurring (repeated new initializations and/or failure PRMs)? If YES, go to Step 25. If NO, continue to the next step.
23	Evaluate the system. Inhibit options successful boot requires careful system evaluation. If the system will recover with some checks inhibited, it may still be taking errors that make the system unable to perform essential functions.

Step	Procedure
24	Is the system still unusable or unstable? If YES, continue to the next step. If NO, STOP. Contact technical assistance for further system evaluation and instructions.
25	Examine the failure data collected in Steps 18 and 19 (configuration options). Did all configurations exhibit the same failing signature by having the same failing or last successful PRM not including configuration bits? If YES, go to the "Recovery from Backups Procedure" section and execute the procedure beginning with Step 1. If NO, continue to the next step in this procedure.
26	If different configurations fail to recover with different signatures, the powerdown options should be used to isolate possible faulty equipment from the system. Follow the procedure described in the "Powerdown Options" section located earlier in this chapter.
27	Are 3B20D/3B21D computer initialization failures still occurring (the "Powerdown Options" section completed without success)? If YES, go to the "Recovery from Backups Procedure" section and execute the procedure beginning with Step 1. If NO, go to the "Post Recovery Procedure" section and then continue.

Recovery from Backups Procedure

Step	Procedure
1	First attempt to recover on all configurations using the Backup Root option: a. Clear EAI forces by entering the following EAI poke: 14 b. Select Backup Root by entering the following EAI poke: 30 c. Force primary boot device by entering the following EAI poke: 20 d. Force CU 1 by entering the following EAI pokes: 11 followed by y e. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y f. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to Step 1-g. g. Force CU 0 by entering the following EAI pokes: 10 followed by y h. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y i. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to Step 1-j. j. Force secondary boot device by entering the following EAI poke: 22

Step	Procedure
	k. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y l. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to Step 1-m. m. Force CU 1 by entering the following EAI pokes: 11 followed by y n. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y o. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step.
2	Does the EAI screen have fields 16, 28, and 29? If YES, go to the "Recovery Using Software Backup Disk and Alternate Boot Disk Options" section and execute the procedure beginning with Step 1. ⇒ NOTE: If the system recovery continues to fail when executing the "Recovery Using Software Backup Disk and Alternate Boot Disk Options" procedure, at some point, you will be referenced to return back to this procedure and start execution at Step 4. If NO, continue to the next step in this procedure.

Step	Procedure
3	Is this office equipped with Software Backup Disks (usually MHD 14 and 15)? If YES, go to the "Recovery Using Software Backup Disk and Alternate Boot Disk Options" section and execute the procedure beginning with Step 1.  NOTE: If the system recovery continues to fail when executing the "Recovery Using Software Backup Disk and Alternate Boot Disk Options" procedure, at some point, you will be referenced to return back to this procedure and start execution at Step 4. If NO, go to Step 5 in this procedure.
4	Did the system recover using the software backup disks? If YES, go to the "Post Recovery Procedure" section and then continue. If NO, continue to the next step.
5	The recovery from the backup tape strategy is to first attempt to recover using the backup from the newest session (the newest set of backup volumes or tapes). If that attempt fails, the process should continue using tape(s) from the next most recent backup session.
6	Choose the CU to be active during Load Disk from Tape (LDFT) procedure: If one CU is suspected to be faulty or will not boot as far as the other, then choose the best performing CU. If both CU 0 and CU 1 fail to boot with similar symptoms, then choose CU 0. Force the CU configuration using the selected "Boot CU" designated as 'X' in this procedure. - On the MTTY EAI Page, enter command 14 to clear the EAI. - On the MTTY EAI Page, enter command 1X (poke 10 or 11) to force the CU X active. - If the EAI Page responds with Force CU, enter y and continue to the next step.

Step	Procedure
7	Choose the MHD configuration to be used during the LDFT procedure. If both MHD 0 and MHD 1 are thought to contain similar bootable images, one MHD will be loaded from the tape at this time. Determine the MHD to be loaded using the following criteria: ▪ If backup tapes or software backup disks are significantly (6 months) out of date, seek technical assistance. Every attempt should be made to save the original data on at least one of the original system disks for use during post recovery operations. ▪ The last active MHD should not be considered for loading first unless it is known to be bad. This disk could contain information useful during post recovery operations. ▪ If any MHD was replaced (but not by a software backup disk) or otherwise rendered unusable, it should be selected to be loaded from backup tapes. ▪ If any MHD fails to boot and has Disk Driver PRMs indicating hardware errors, the MHD is faulty. DO NOT use any faulty MHDs for LDFT. However, if a replacement disk is available, replace the disk and use the new boot MHD for LDFT. ▪ If one system disk (MHD 0 or MHD 1) is suspected to be faulty or will not boot as far as the other, then choose it for LDFT. ▪ If both original disks fail with the same failing signature, then choose MHD 0. Force the CU-MHD configuration using the selected "Boot MHD" and "Boot CU". Refer to the Figure 3-1. On the MTTY EAI Page, enter command 22 to select secondary disk subsystem (CU 0 with MHD 1 or CU 1 with MHD 0). On the MTTY EAI Page, enter command 20 to select primary disk subsystem (CU 0 with MHD 0 or CU 1 with MHD 1).
8	If the disk selected is a replacement disk, then use the format option (EAI 42 i) with LDFT.

Step	Procedure
9	Is this office equipped using 9-track tape technology? If YES, go to the "Loading Single Volume Tapes (Formerly the Standard LDFT Procedure)" section and execute the procedure beginning with Step 3. If NO, continue to the next step in this procedure.
10	Is this office using a Digital Audio Tape (DAT) unit with a single multivolume tape for making backups (requires the 3B21D computer)? If YES, go to the "Loading Multivolume Tapes" section and execute the procedure beginning with Step 1. When you have finished executing the "Loading Multivolume Tapes" procedure, return to this procedure and continue to the next step. If NO, go to the "Loading Single Volume Tapes (Formerly the Standard LDFT Procedure)" section and execute the procedure beginning with Step 3. When you have finished executing the "Loading Single Volume Tapes" procedure, return to this procedure and continue to Step 11.
11	Did the system recover using the DAT backup tapes? If YES, go to the "Post Recovery Procedure" section. If NO, continue to the next step.
12	Immediately follow local escalation procedures to contact your next level of technical support. Escalation should continue until the Lucent Technologies application (this switching product) CTS or DFS and 3B/RTR Global CTS organizations have been contacted. The Lucent Technologies field support engineers will provide additional support, if necessary, to resolve the problem.
13	Choose the next set of backup volumes or tapes, and continue from Step 9 in this procedure.

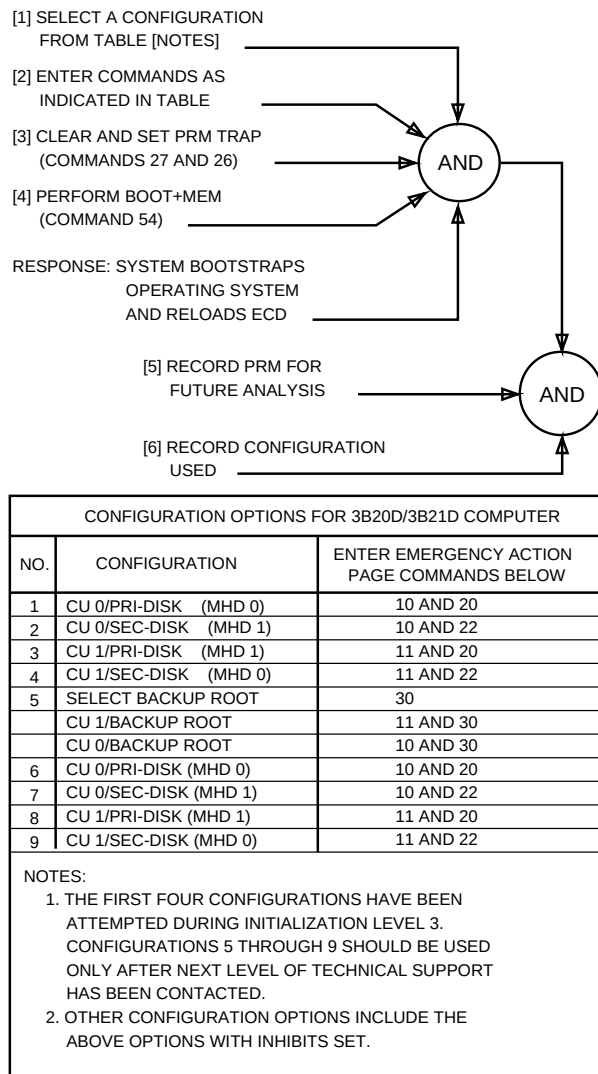


Figure 3-1. Recovery Procedure (System Configuration)

Application Recovery Procedure

This IP is **NOT** intended to replace or be used instead of the Lucent Technologies switching product recovery procedures. The application using the 3B20D/3B21D computer has provided detailed recovery procedures which provide specific guidelines and instructions which may help in the analysis and resolution of the outage and reduce the recovery time.

⇒ NOTE:

Please follow application recovery procedures if available.

Figure 3-2 is a step-by-step recovery procedure that should be followed when no application specific detailed recovery procedure is available.

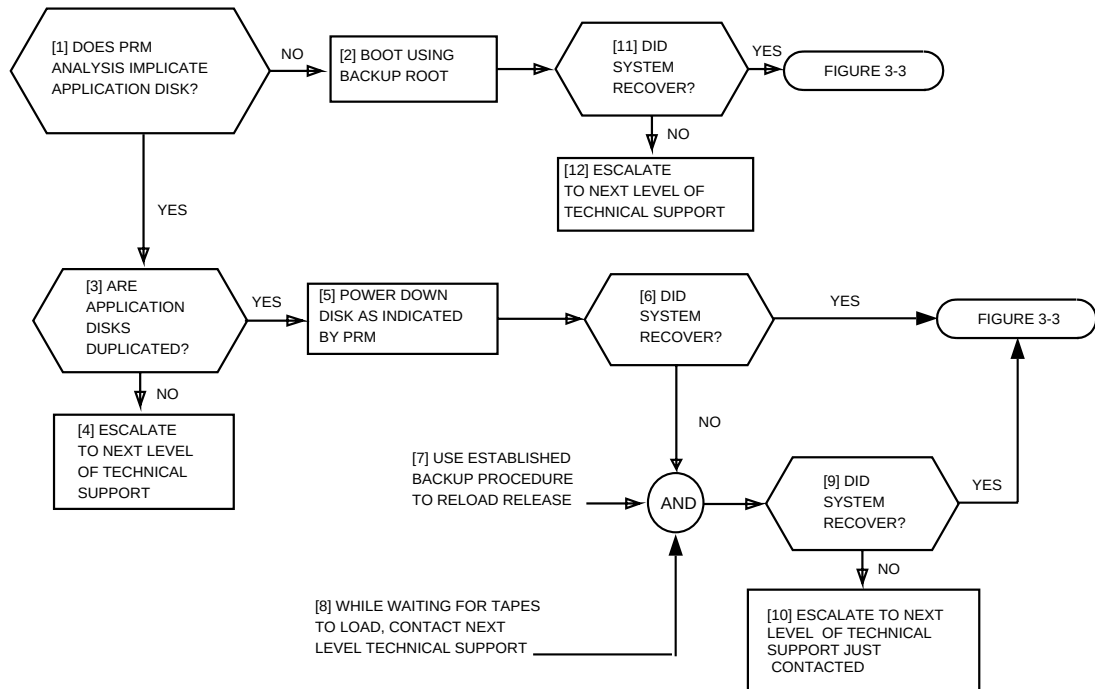


Figure 3-2. Recovery Procedure (Application Failure)

Post Recovery Procedure

Figure 3-3 is a step-by-step post recovery procedure that should be followed when no application specific post recovery procedure is available. Post recovery activity usually requires following an application recovery procedure to ensure system integrity and clearing of all troubles that led to the system outage as soon as possible. Escalation and consultation with highest levels of technical support are recommended.

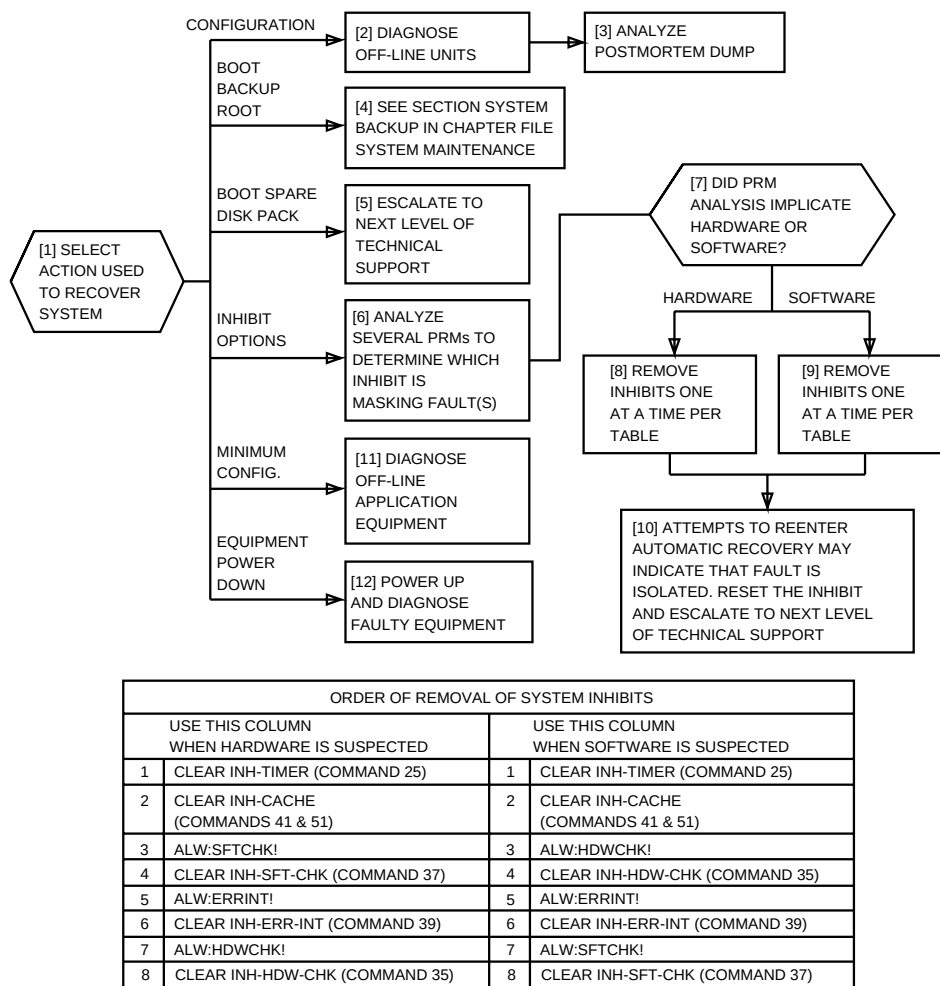


Figure 3-3. Recovery Procedure (Post Recovery)

Using Processor Recovery Messages

The PRMs are generated to indicated steps in processor recovery. For details on how to interpret PRMs and correct faults, refer to 254-303-103, *Processor Recovery Messages Guide*.

Responding to a Terminal Suspend

Terminal suspends occur in the following ways:

- When a single terminal or multiple terminals (but not all terminals), attached to the system fail to function, refer to the “One or More Terminals Faulty” section to clear the problem.
- When all terminals attached to the system fail to function, refer to the “All Terminals Do Not Function” section to resolve the problem.

One or More Terminals Faulty

Power cycle the affected terminal to see if it will recover. If power cycling the terminal(s) fails to restore functionality, a system problem is probable and the following checks need to be performed before taking any other action:



NOTE:

Do not perform a craft initialization.

Step	Procedure
1	Using a terminal that functions, verify the integrity of all mounted file systems with the AUD:FSBLK and AUD:FSLINK file system audits. If any of the file system checks fail, escalate the problem immediately to someone knowledgeable who deals with file system corruption. Any further action at this point may result in the loss of all terminals and access to the system.
2	Dump the <i>ERLOG</i> log file to verify that the system did not encounter any Library C (LIBC) faults. Check the ROP for any Process Manager (PMGR) or File Manager (FMGR) errors which may have preceded the terminal suspends. If any of these fault types were observed, suspect file corruption and/or file system damage. Do not take further recovery actions. Contact someone familiar with these faults.

If power cycling the terminal corrected the problem, power cycle any other terminals that are affected.

If power cycling the terminal does not help, refer to Chapter 2, "Maintenance Plan," for help in resetting and resolving terminal-related problems.

All Terminals Do Not Function

If all terminals fail to function, perform a craft initialization (EAI 42,3,15). If this fails to recover terminal functionality, a system initialization will be required. The following precautions should be exercised when attempting a system recovery from a total terminal suspend.

- Be prepared to perform a system reload if all boot combinations fail to restore system sanity. Since access to the system is unavailable, the source of the suspend is difficult to determine.
- Perform the initialization during low traffic periods, if possible.
- Ensure that the proper support organizations are available and are ready to help in case a failure does occur.
- Know what kind of data will be needed/collected before engaging a system initialization.
- Know how to create an off-line memory dump and the procedures required to collect valid data from this fault.

Power Interrupt (Fiber Optic Feature Only)

⇒ NOTE:

This feature is not available for the 3B21D computer.

Verify that the cabling connecting the port switch unit (CU 0) to the MTTY or ROP is a fiber optic cable. If fiber optic cable is part of the configuration, the cable at the MTTY end should have the 4131p type modem attached to it and the cable at the ROP end should have the 4131s type modem. On the CU 0 end of the connection, there should be two RS-232C modems installed on a patch panel. For the 3B20D computer only, the patch panel is located at level 12 in bay 0 (CU 0).

If a power interrupt causes the MTTY or ROP to lock up when power is applied, perform the following procedures:

MTTY Power Interrupt Recovery

Step	Procedure
1	Turn power switch on.
2	Press the EA DISP key.
3	Press the NORM DISP key.

ROP Power Interrupt Recovery

If ROP is in **LOCAL** mode:

Step	Procedure
1	Press the LINE key.
2	Press the LOCAL key.
3	Press the LINE key.

If ROP is in **LINE** mode:

Step	Procedure
1	Press the LOCAL key.
2	Press the LINE key.

Non-MTTY Suspend

If the MTTY is functioning properly but some other terminal is not functioning, clear the problem by performing the following procedure:

Step	Procedure
1	If a single terminal is suspended, power cycle the terminal not functioning to reinitialize the terminal handler.
2	If a single terminal is suspended at the MTTY, remove the affected Teletypewriter Controller (TTYC) from service (RMV:TTYC) and run diagnostics (DGN:TTYC). When All Tests Pass (ATP), restore the TTYC to service (RST:TTYC).
3	Initialize the craft interface (Emergency Action Page, Command 15) starting with level 2. Wait at least 5 minutes before trying anything else.
4	If not successful, escalate the terminal recovery to level 3.
5	Replace the terminal.
6	Remove the affected IOP controller from service (RMV:IOP) and run diagnostics (DGN:IOP). When diagnostics pass (ATP), restore the unit to service (RST:IOP;UCL).
7	If this procedure fails to restore the terminals, contact your technical assistance organization.

MTTY Suspend

If the MTTY is not functioning correctly (for example, if the cursor cannot be moved up and down), perform one of the following procedures:

KS-23554, L5 and L6 Video Terminal

Step	Procedure
1	The terminal can be self-diagnosed by power cycling the terminal. Refer to the <i>KS-23554, L5 and L6 Video Terminal Manual</i> for details.
2	Reset the terminal to a known, usable mode by pressing the Ctrl + Alt + F3 keys to enter setup.
3	Move the cursor to the Factory-Defaults function and press the RETURN key. The Factory-Defaults function will display a menu with NO and YES responses.
4	Move the cursor to the YES menu item and press the RETURN key.
5	After the terminal returns to the function selection menu, move the cursor to the Save function and press the RETURN key.
6	The Save function will display a menu with NO and YES responses. Move the cursor to the YES menu item and press the RETURN key.
7	After the terminal returns to the function selection menu, move the cursor to the Exit function and press the RETURN key. The terminal will return to operation as the MTTY in the "Terminal Only (1 session)" mode.

KS-23996, L1 and L5 Video Terminal (3B21D Computer Only)

Step	Procedure
1	The terminal can be self-diagnosed by power cycling the terminal. Refer to the <i>KS-23996, L1 and L5 Guide to Operations</i> for details.
2	Reset the terminal to a known, usable mode by pressing Ctrl + Alt + F3 keys to enter setup.
3	Press the F13 key.
4	Move the cursor to the Default Terminal function and press the RETURN key.
5	Press the F14 key, and enter y when prompted. The terminal will return to operation as the MTTY.

KS-22921 Video Terminal

Step	Procedure
1	The terminal can be self-diagnosed by power cycling the terminal. Refer to the <i>KS-22921, Video Terminal Manual</i> for details.
2	The terminal can be reset by pressing the SET-UP key; then press the RESET key to restart the terminal and reset the options to their preprogrammed values.
3	Check the options by pressing SET-UP ; then press the SET-UP A-B key.  NOTE: The options should be set to 1000 0011 0000 0X10 XXXX, parity equal to 7E.
4	To change the options, move the cursor to the position over the element to be changed and press the TOGGLE I/O key.
5	To change the speed setting, use the TRANSMIT SPEED and RECEIVE SPEED keys (both should be set to 9600).
6	When the options are correctly set, press the SHIFT bar and the S keys simultaneously to store the new options.

Teletype Model 4424 Terminal

Step	Procedure																								
1	The terminal can be self-diagnosed by power cycling the terminal. If the diagnostic does not pass, the INTERACTIVE key will not light. Refer to the <i>Dataspeed 4424 Display Terminal User's Guide</i> for further information.																								
2	The terminal can be reset by pressing the CONTROL key and the HOME key simultaneously.																								
3	Check the options by pressing the SET-UP key. The options should be set to: <table><tr><td>Speed selection</td><td>9600</td></tr><tr><td>Asynchronous/synchronous</td><td>ASYNC</td></tr><tr><td>Highlight definition</td><td>BLINK</td></tr><tr><td>Auto wrap</td><td>OFF</td></tr><tr><td>New line definition</td><td>LF=INDEX</td></tr><tr><td>Parity check</td><td>EVEN</td></tr><tr><td>Monitor keyboard</td><td>DISABLE</td></tr><tr><td>Display function</td><td>GROUP III</td></tr><tr><td>Receive control</td><td>DC3-STOP/DC1-START</td></tr><tr><td>Printer type</td><td>NONE</td></tr><tr><td>Response to printer alarm</td><td>BREAK-NO DISCONNECT</td></tr><tr><td>Blinking cursor</td><td>DISABLE</td></tr></table>	Speed selection	9600	Asynchronous/synchronous	ASYNC	Highlight definition	BLINK	Auto wrap	OFF	New line definition	LF=INDEX	Parity check	EVEN	Monitor keyboard	DISABLE	Display function	GROUP III	Receive control	DC3-STOP/DC1-START	Printer type	NONE	Response to printer alarm	BREAK-NO DISCONNECT	Blinking cursor	DISABLE
Speed selection	9600																								
Asynchronous/synchronous	ASYNC																								
Highlight definition	BLINK																								
Auto wrap	OFF																								
New line definition	LF=INDEX																								
Parity check	EVEN																								
Monitor keyboard	DISABLE																								
Display function	GROUP III																								
Receive control	DC3-STOP/DC1-START																								
Printer type	NONE																								
Response to printer alarm	BREAK-NO DISCONNECT																								
Blinking cursor	DISABLE																								
4	To change the options, press the down arrow ↓ key until the cursor reaches the desired option; then press the space bar to change the option.																								

If the MTTYC is not operating properly and data is not being transmitted to the MTTY (sanity indicator-number that increments from 0 to 7 on the MTTY screen not incrementing), perform the following procedure:

Step	Procedure
1	Make sure the other MTTY is in the standby mode. Switch the port switch to the other MTTY manually.
2	Perform the checks in the "Non-MTTY Suspend" section.
3	If another terminal is active on the system, use that terminal to remove, diagnose, and restore one of the MTTYCs.
4	Attempt a manual remove and restore of an IOP using the ROS/RST key on the CU power control pack.

Receive-Only Printer Suspend

If the ROP stops printing but the terminals work normally, perform the following procedure:

Step	Procedure
1	Power cycle the ROP to reinitialize the ROP handler.
2	Switch the associated ROP port switch to the other IOP.
3	Initialize the craft interface (Emergency Action Page, Command 15). Wait at least 5 minutes before trying anything else.
4	Remove, diagnose, and restore one of the MTTYCs; then port switch the ROP.
5	Activate the ROP self-test feature. (Refer to ROP manufacturer's manual for details.)
6	Replace the ROP.
7	If these steps fail to restore the ROP, contact your technical assistance organization.

Craft Initialization

The craft interface processes are executed and monitored by the User-Level Automatic Restart Process (ULARP). (Refer to the "Initialization Procedures" section in Chapter 10.)

Port Switch

The port switch manually forces the MTTY and ROP to either of the IOPs. Always try a port switch before concluding that a boot is necessary. Wait at least 5 minutes before trying anything else.

Creating a Dump Tape

If all other procedures fail to restore terminal access, the computer may have to be booted.

To assist your technical assistance organization in finding and correcting causes of terminal suspends and other system anomalies, make an off-line memory dump to tape after the system has been booted.

To boot the system and make an off-line memory dump to tape, perform the following procedure:

Step	Procedure
1	Install the tape on the tape drive.
2	Force the active CU on-line (Emergency Action Page, Command 12).
3	Enter the appropriate command at the MTTY for the type of tape drive installed in the office. For a <i>Keystone</i>* III—KS-23113 or SCSI devices, enter the following: In PDS, EXC:ENVIR:UPROC,FN"/bin/sh",ARGS("-c",\ "pio dd if=/dev/ofln of=/dev/mt00 bs=6144")! In MML, EXC:ENVIR:DATA,UPROC,FN="/bin/sh",ARGS="-c"- \ "pio dd if=/dev/ofln of=/dev/mt00 bs=6144"; For a <i>Keystone</i> II—KS-22762 or Kennedy 9000—KS-22091, enter the following: In PDS, EXC:ENVIR:UPROC,FN"/bin/sh",ARGS("-c",\ "pio dd if=/dev/ofln of=/dev/mt08 bs=6144")! In MML, EXC:ENVIR:DATA,UPROC,FN="/bin/sh",ARGS="-c"- \ "pio dd if=/dev/ofln of=/dev/mt08 bs=6144"; When the command completes execution, the computer responds with the following: EXC ENVIR UPROC COMPLETED

* Keystone is a registered trademark of Laser Magnetic Storage International Company.

Step	Procedure
4	Return the CUs to duplex by removing the force on-line (Command 13) and restoring the CU (<i>RST:CU</i>).
5	Send the tape, a copy of the sheet covering the attempts made to clear the suspend, and any other applicable information or ROP printouts to the application field support group.

Isolating Emergency Action Page Problems

The following procedure describes the actions to take to isolate the fault if the Emergency Action Page displayed on the MTTY goes blank or becomes unresponsive.

⇒ NOTE:

The terminal to CU MTTY interface is shown in Figure 3-4 (3B20D Model 1 computer), Figure 3-5 (3B20D Model 2 or Model 3 computer), Figure 3-6 (3B21D computer with a TN983 MTTYC), and Figure 3-7 (3B21D computer with a UN583 MTTYC).

Step	Procedure
1	At the MTTY: a. Verify power is applied. b. Verify terminal operation. (Some terminals execute an onboard diagnostic routine when the power is cycled or when a special purpose command is executed.) c. Verify that the 25-pin connector is properly connected to the terminal.
2	At the port switch: a. Verify power is applied. b. Verify that the cable on the terminal side of the port switch and both cables on the MTTYC side of the port switch are properly connected. c. Switch the port switch to the other port.
3	At the MTTYC: a. Verify power is applied. b. Verify cable connections. c. Run diagnostics. d. Replace the MTTYC circuit pack. The MTTYC paints the Emergency Action Page display on the terminal screen. A blank screen may indicate that the MTTYC is not functioning.

Step	Procedure
4	At the EAI circuit pack (3B20D Model 1 computer – TN11, 3B20D Model 2 and 3 computers – TN10, or 3B21D computer CC circuit pack – K LW31): a. Verify power is applied. b. Check cable connections. c. Replace the circuit pack. The EAI provides the real-time status information displayed in the PROCESSOR STATUS section of the Emergency Action Page (lines 1 and 2). If this information is not being displayed, the TN11 or TN10 or K LW31 may not be functioning.

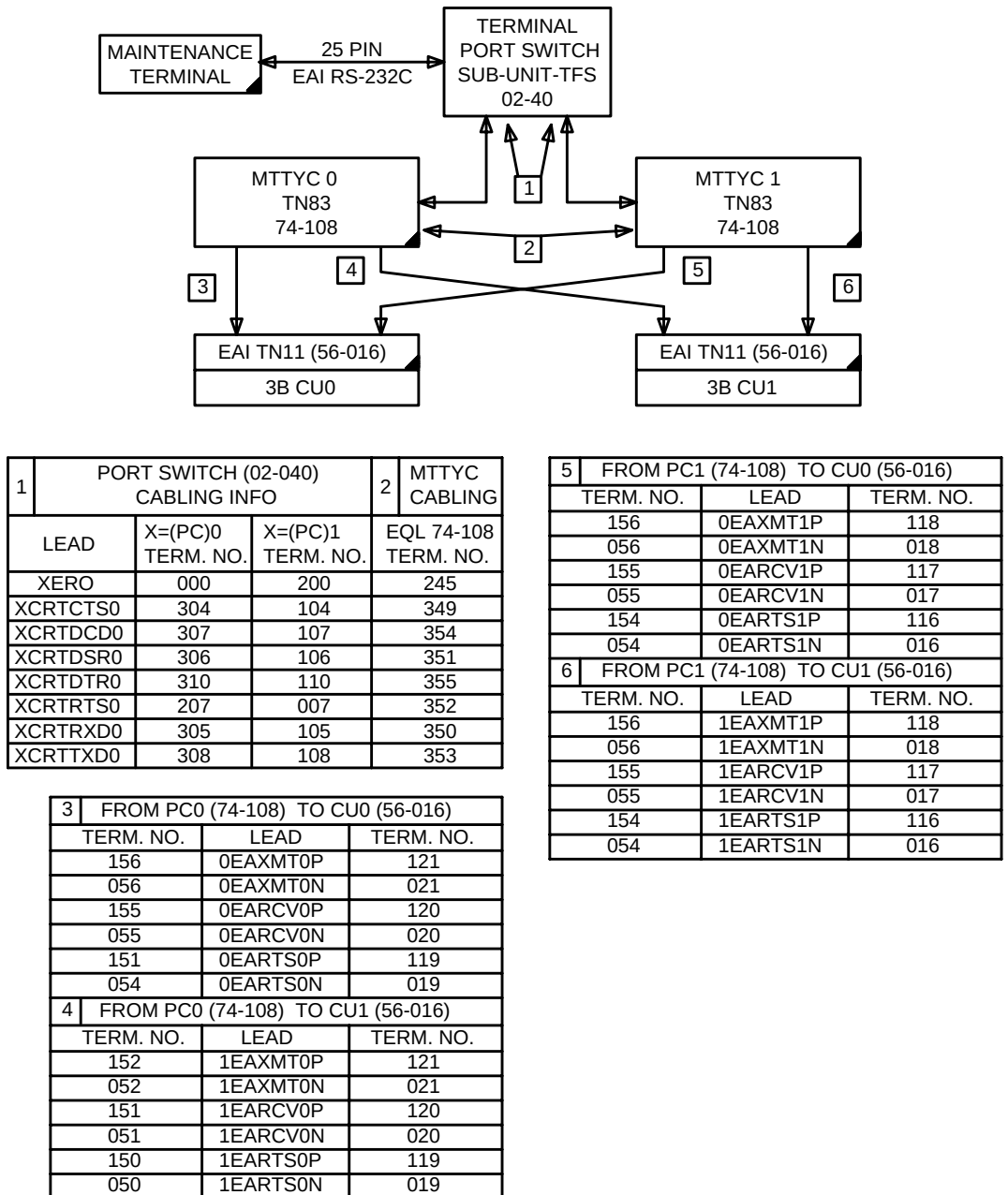


Figure 3-4. MTTY/3B20D Computer Interface (Model 1)

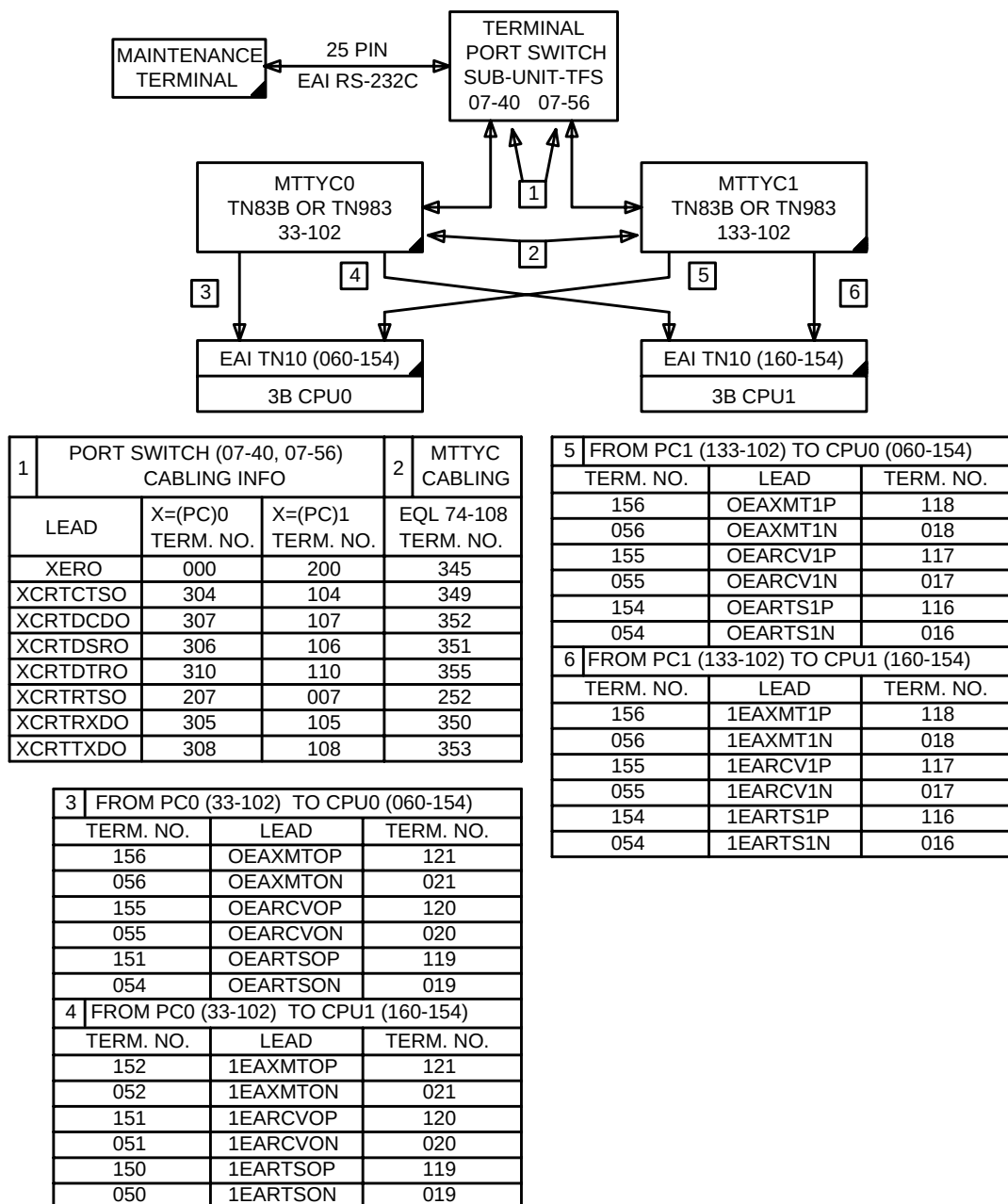


Figure 3-5. MTTY/3B20D Computer Interface (Model 2 or Model 3)

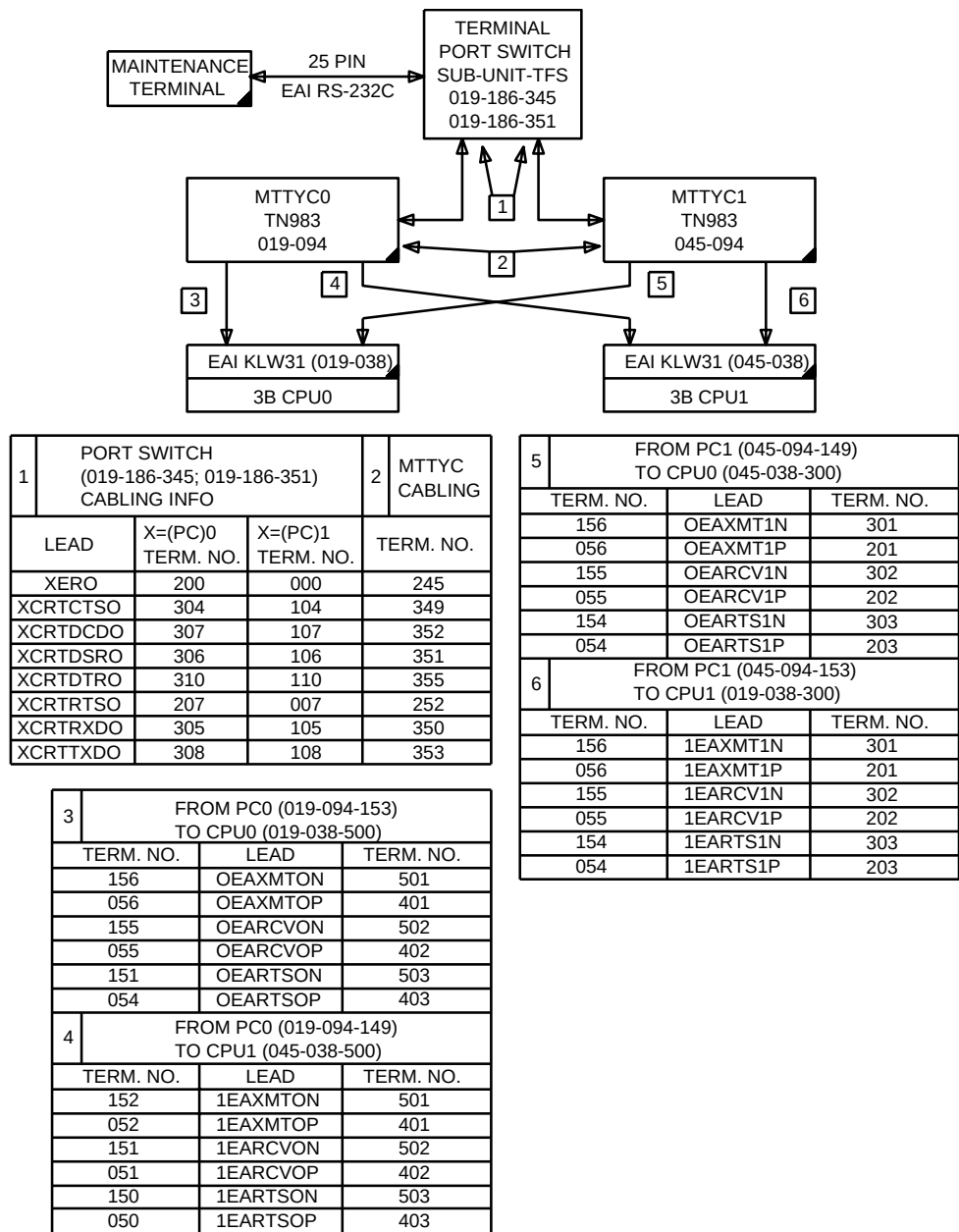


Figure 3-6. MTTY/3B21D Computer Interface with a TN983 MTTYC

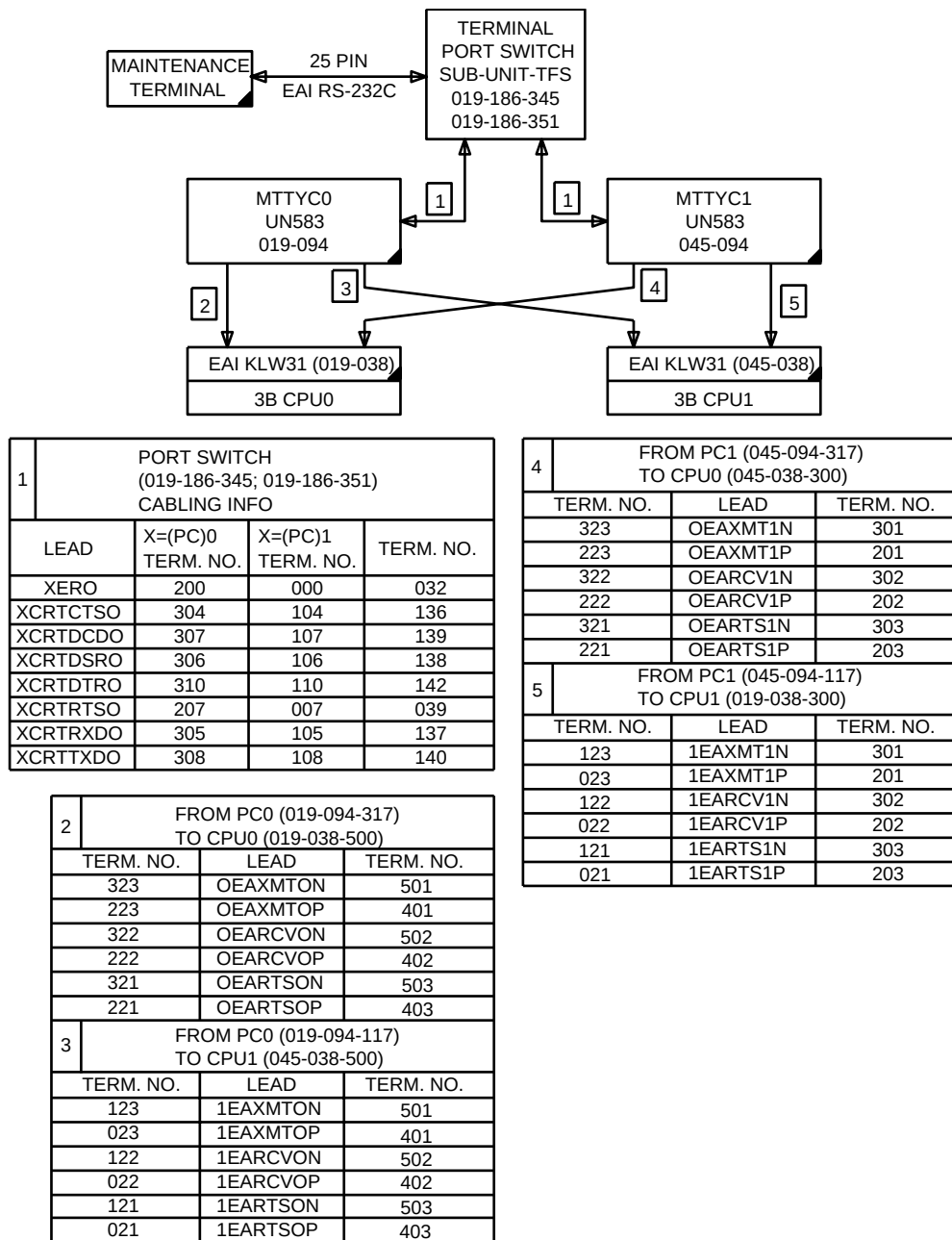


Figure 3-7. MTTY/3B21D Computer Interface with a UN583 MTTYC

Recovery Using Software Backup Disk and Alternate Boot Disk Options

The 3B20D/3B21D computer systems equipped with extra disk drives (usually MHD 14 and/or MHD 15) that are backups of the system disks (MHD 0 and MHD 1) can be used to manually recover an office with either a duplex system disk hardware failure or to recover by backing up to a previous copy of the system disk (before an error has been introduced). Disks specifically configured and used in this manner are called Software Backup Disks (SBDs) and sometimes referred to as warm spare drives.

The recovery from the Alternate Boot Disk (ABD) feature allows an SBD to be used as the alternate boot disk and initiate recovery from the EAI Page without having to manipulate the MHD hardware. In short, the ABD feature extends the bootstrap capability to the SBD.

The ABD feature is an optional feature that is implemented in RTR Release 21.27 and later software releases and requires a 3B21D computer system equipped with a pair of UN597 MTTYCs. The ABD feature is not supported on the 3B20D computer, AM Work Station (AWS), or the Very Compact Digital Exchange (VCDX) platforms.

The use of SBD does not take the place of making periodic and complete tape backups.

For details on recovering nonsystem disks, refer to Chapter 9, "File-System Maintenance."

For a picture of the EAI screen, refer to Chapter 12, "Craft Interface."

To recover from a duplex system disk outage in systems equipped with the Software Backup Disks (MHD 14 and/or MHD 15) or the Alternate Boot Disk option, perform the following procedure:

Step	Procedure
1	On the MTTY, push the EAI key to display the EAI screen. Does the EAI screen have fields 16, 28, and 29? If YES , go to the "Recovery Using Alternate Boot Disk" section. If NO , continue to the next step.

Step	Procedure
2	If there are "warm spare disks" SBDs (usually MHD 14 and MHD 15), used to back up the boot drives (MHD 0, MHD 1), choose the most recent SBD.
3	Switch identities between the drive selected and the boot disk. Refer to the "SBD Switching Identities Procedure" section for detailed instructions on how to switch identities.
4	If the new SBD has been configured as MHD 0, continue to the next step. If the new SBD has been configured as MHD 1, go to Step 6.
5	Attempt to recover with the new MHD 0: a. Force primary boot device by entering the following EAI poke: 20 b. Force CU 0 by entering the following EAI pokes: 10 followed by y c. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y d. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step. e. Force primary boot device by entering the following EAI poke: 22 f. Force CU 1 by entering the following EAI pokes: 11 followed by y g. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y

Step	Procedure
	h. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and go to Step 7.
6	Attempt to recover with the new MHD 1: a. Force primary boot device by entering the following EAI poke: 20 b. Force CU 1 by entering the following EAI pokes: 11 followed by y c. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y d. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to Step 6-e. e. Force primary boot device by entering the following EAI poke: 22 f. Force CU 0 by entering the following EAI pokes: 10 followed by y g. Request Boot+MEM full system initialization by entering the following EAI pokes: 54 followed by y h. Does the system recover? If YES, go to the "Post Recovery Procedure" section and then continue. If NO (recovery fails), record the failing PRM or last successful PRM and continue to the next step.

Step	Procedure
7	If the office is equipped with additional SBDs, then continue to the next step. Otherwise, go to the "Recovery from Backups Procedure" section and execute the procedure beginning at Step 4.
8	Switch identities between the other SBD and its boot disk. Refer to the "SBD Switching Identities Procedure" section. After switching identities, restart this procedure at Step 4.

SBD Switching Identities Procedure

To switch identities of a disk drive, proceed as follows:

Step	Procedure
1	For 340-MB disks: The DFC must be Out of Service (OOS) before removing or inserting the ID plug. No access of the drive must take place (for example, DUMP VTOC, VERIFY, and so forth) between the time the drive was removed from service and the time the ID plugs are changed. Remove both drives from service, exchange the ID plugs between the SBD and the system disk (usually 0 and 7), and go to Step 3.
2	For SCSI disks: The drive must be powered down before changing the setting of the ID switch. Remove power from both drives, toggle the settings of each drive's switch to that of the other drive, and power up both drives. The setting for the new MHD 0 or MHD 1 must be 0. Continue to the next step.
3	Return to the "Recovery Using Software Backup Disk and Alternate Boot Disk Options" section, Step 4.



CAUTION:

During the period that the drive IDs are swapped, SCAN POINT and SD POINT information and data (ROP messages such as POWER DOWN, and so forth) derived from the SCAN POINT information will NOT be swapped. This will lead to misinformation being output on the ROP (including the SCC).

Recovery Using Alternate Boot Disk

If there are Alternate Boot Disks backing up the boot drives (MHD 0, MHD 1), select the latest (newest) warm spare drive "Boot MHD" and use the following procedure to force the system configuration and boot using the selected "Boot MHD" designated as 'A'.

Step	Procedure
1	On the MTTY keyboard, push the EAI key to display the EAI screen.
2	Is the EAI Page displayed on the MTTY? If YES , continue to the next step. If NO , go to the "Isolating Emergency Action Page Problems" section or seek technical assistance. The ABD option cannot be used without a functioning EAI screen.
3	On the MTTY EAI Page, enter command 14 to clear the EAI.
4	On the MTTY EAI Page, enter command 10 to force the CU 0 active.  NOTE: If CU 0 is known to be faulty, enter command 11 on the MTTY EAI Page to select CU 1.
5	Did the EAI Page respond with Force CU? If YES , enter y and continue. If NO , continue to the next step.
6	On the MTTY EAI Page, enter command 28 to select the Boot MHD.
7	Did the EAI Page respond with Select boot device? (0-15) ? If YES , enter A and continue to the next step. If NO , go to Step 3 to retry the procedure.
8	Did the EAI Page respond with Set boot devices A/1 (y/n) ? If YES , enter y and continue to the next step. If NO , go to Step 10.

Step	Procedure
9	On the MTTY EAI Page, enter command 20 to select primary disk subsystem, and go to Step 12.
10	Did the EAI Page respond with Set boot devices 0/A (y/n) ? If YES , enter y and continue to the next step. If NO , return to Step 3.
11	On the MTTY EAI Page, enter command 22 to select secondary disk subsystem.
12	On the MTTY EAI Page, enter command 54 to boot the 3B20D/3B21D computers. (Yes, we really mean to enter a 54 .)
13	Did the EAI Page respond with Boot? (y/n) ? If YES , enter y and continue to the next step. If NO , return to Step 12.
14	Did the initialization complete? If YES , go to the "Post Recovery Procedure" section. If NO , continue to the next step.
15	On the MTTY EAI Page, enter command 30 to select backup file-systems.
16	On the MTTY EAI Page, enter command 54 followed by y to boot with this configuration.
17	Did the initialization complete? If YES , go to the "Post Recovery Procedure" section. If NO , continue to the next step.

Step	Procedure
18	Have you retried the selected ABD using CU 1? If YES, go to Step 19. If NO, retry the selected ABD using CU 1: - On the MTTY EAI, enter command 14 to clear the EAI. - On the MTTY EAI, enter command 11 followed by y to force the CU 1 active. - Go to Step 6 and retry the procedure.
19	If there are other Alternate Boot Disks backing up the boot drives (MHD 0, MHD 1), select the next warm spare drive "Boot MHD". Then, use this procedure starting at Step 1 to force the system configuration and boot using the selected "Boot MHD" designated as 'A'.

Recovering from DIOP

The purpose of these procedures is to recover the essential system disks after a system disk duplex failure. Once the disks are available, the procedures guide the craft in booting. The recovery strategy is as follows:

- Repair both of the essential disks (MHD 0 and MHD 1) using the DFC status page.
- Prepare the system MHDs so that more than one program version is available to improve the reliability of recovery at the time of boot.

⇒ NOTE:

When recovering from DIOP, consultation with Technical Assistance Groups is recommended.

The following is a list of comments concerning recovery from the DIOP procedures:

- These procedures assume that essential *UNIX* RTR operating system and application processes for call processing are locked in memory to allow call processing to continue. This assumes that when DIOP was entered, a REPT SIMPLEX PROCESSING COMPLETE message was printed or a REPT DIOP SIMPLEX UNSWAP FAILURE message was NOT printing every 5 minutes. If call processing has been lost, refer to the "Load Disk from Tape Procedure" section.
- While in the DIOP, **DO NOT** perform an initialization before both of the essential system disks are repaired and believed to be operational.
- The craft must be on site during the restoration from DIOP to ensure system recovery.
- Unessential application services, such as Automatic Message Accounting (AMA) recording, may be terminated during DIOP.
- Craft initialization (EAI cmd **15**) is not operational in DIOP.
- When backup tapes are loaded, a mismatch may be created between the newly booted 3B20D/3B21D computers resident application processes/data and the previously initialized application peripheral units. Technical assistance for the application should be called to resolve these mismatches.

- Serious problems can result by not adhering to the following sequence of procedures as detailed in the following sections.
 1. Verify DIOP Mode in Effect Procedure
 2. Prepare DFCs and MHDs for Recovery Procedure
 3. Load Disk from Tape Procedure
 4. Recovery Decision Procedure
 5. Boot Procedure
 6. SMD DFC Repair Guidelines
 7. SMD MHD Repair Guidelines
 8. SCSI DFC Repair Guidelines
 9. SCSI MHD Repair Guidelines.

Verify DIOP Mode in Effect Procedure

This procedure assumes that the switch is performing all essential activities (such as processing calls). An independent check (such as completing various test calls) should be made to verify that the application is performing essential activities. If essential activities are not working, this procedure should not be used; instead, use the procedure in the "Recovery Procedure" section.

Step	Procedure
1	At the MTTY, operate the NORMAL display key and enter Poke 120. The system response for <i>UNIX</i> RTR operating system releases is shown in Figure 3-8. ⇒ NOTE: While in DIOP only, DFC status page commands are operational. A representative printout, every 15 minutes on ROP, is as follows: <pre>DISK INDEPENDENT OPERATION MODE IN EFFECT DISK INDEPENDENT OPERATION MODE STATUS MHD 0 00S DISK INDEPENDENT OPERATION MODE STATUS MHD 2 ACT DISK INDEPENDENT OPERATION MODE STATUS DFC 0 ACT DISK INDEPENDENT OPERATION MODE STATUS MHD 1 00S ffff DISK INDEPENDENT OPERATION MODE STATUS MHD 3 ACT DISK INDEPENDENT OPERATION MODE STATUS DFC 1 ACT</pre> Where: <i>ffff</i> = FDLM or FDIOP

Step	Procedure
2	Is the DFC status page displayed on the MTTY? If YES , continue to Step 3. If NO , go to Step 4.
3	On the MTTY, is the DFC status page CURRENT LEVEL shown as FULL DLM or FULL DIOP ? If YES , go to the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , continue to Step 4.
4	Seek technical assistance. The system may be in DIOP or terminal suspend. If the system is in DIOP and terminal suspend, this procedure cannot help you. Reloading the system disk pair from tape (or software backup disk) will require an interruption in call processing for a full system recovery. If the switch is processing calls, WAIT before initializing the switch with a boot. Such a boot WILL terminate call processing. If the office is experiencing a duplex system disk outage, call processing cannot be restored until the disk problem is resolved.
5	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION			DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM	
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS				
CMD<_				120 - DFC STATUS PAGE				
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = FULL DIOP				
DFC-0	ACT	Y	FIRMWARE					
SBUS-0	ACT	Y	SBUS-2	ACT	Y	SBUS-3	ACT	Y
MHD-0	OOS	FDIO Y E	MHD-2	ACT	Y	MHD-3	OOS	N
MHD-4	ACT	Y	MHD-6	ACT	Y	MHD-7	ACT	Y
MHD-8	ACT	Y	MHD-10	ACT	Y	MHD-11	ACT	Y
MHD-12	ACT	Y	MHD-14	ACT	Y	MHD-15	ACT	Y
MT-1	ACT	Y						
CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	
600	INIT MHD	603	DUMP DEFECT	606	RST MT	677	SET MSG LVL	
601	VFY MHD	604	RST/RMV SBUS	607	DUMP BKTAPE	688	CLEAR BUFF	
602	RST DFC	605	RST MHD	620	LOAD MHD	699	ABORT CMD IP	
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <								

Figure 3-8. Poke 120 — DFC Status Page

Prepare DFCs and MHDs for Recovery Procedure

NOTE:

The strategy for recovery from DIOP will be to maximize the potential for a successful and quick boot. In brief, this is done by preparing each system MHD (and software backup disk, if equipped) for the boot attempt.

The DIOP LDFT will be used on at least one system MHD while in DIOP. This will be done even though the software backup disks (MHD 14/MHD 15) may exist. The age of data (on backup tapes and software backup disks) and subjective reliability of the data (on tape and disk) are the factors used to decide which tapes to load and which MHDs are to be loaded.

CAUTION:

*Guidelines in the "SMD DFC Repair Guidelines" and the "SMD MHD Repair Guidelines" sections should be followed when repairs must be done on DFCs, MHD 0, or MHD 1 that are Storage Module Drives (SMDs).
Guidelines in the "SCSI DFC Repair Guidelines" and the "SCSI MHD Repair Guidelines" sections should be followed when repairs must be done on DFCs, MHD 0, or MHD 1 that are SCSI.*

Step	Procedure
1	Ensure that DFC 0, DFC 1, MHD 0, and MHD 1 are powered up.
2	On MTTY display, restore the DFCs shown as out of service, if any. Enter command 602,x . Where: x = 0 or 1 ; DFC # shown as out of service on the MTTY display.
3	Were both DFCs restored to service? If YES , continue to the next step. If NO , correct the fault and continue. For repair guidelines, see the "SMD DFC Repair Guidelines" and "SMD MHD Repair Guidelines" sections.
4	Are any fault indicators lighted on either MHD 0 or MHD 1? If YES , correct the fault and continue. For repair guidelines, see the "SMD DFC Repair Guidelines" and "SMD MHD Repair Guidelines" sections. If NO , continue to the next step.

Step	Procedure
5	Is there one MHD on each DFC identified with a "0" ID plug (or "0" ID switch on SBUS 0 and SBUS 1) that is usable as MHD 0 and MHD 1? If YES, go to Step 10. If NO, continue to the next step.
6	If this office is not equipped with Software Backup Disks (SBDs), go to Step 9. Offices equipped with SBD (includes ABD) should use the SBD to replace the missing/faulty boot disk (MHD 0 or MHD 1). Power down both MHD 0 and MHD 14 (or MHD 1 and MHD 15).
7	Exchange the identities of the primary disk MHD 0 (or MHD 1) and the software backup disk MHD 14 (or MHD 15). The primary disk will take the identity of the software backup disk. The software backup disk will take the identity of the primary disk. If SMD, change MHD 0 (MHD 1) ID plug to 7 and change MHD 14 (MHD 15) ID plug to 0. If SCSI, change MHD 0 (MHD 1) ID switch to 3 and change MHD 14 (MHD 15) ID switch to 0. If a 3B21D computer, replace the new MHD 0 (MHD 1) into the correct location for the system disk at Equipment Location (EQL) 028-162 (or for MHD 1 EQL 053-162).
8	Power up MHD 0 and MHD 14 (or MHD 1 and MHD 15).
9	At this time, two disks are identifiable as MHD 0 and MHD 1 with "0" ID plugs/ID switches. On MTTY display, verify MHD 0. Enter command 601,0. On MTTY display, verify MHD 1. Enter command 601,1.
10	If either MHD does not verify successfully, repair or replace it to correct the fault. See one of the following sections for repair guidelines: "SMD DFC Repair Guidelines" "SMD MHD Repair Guidelines" "SCSI DFC Repair Guidelines" "SCSI MHD Repair Guidelines"  NOTE: DO NOT continue until MHD 0 and MHD 1 verification completes.

Step	Procedure
11	Obtain the most recent set of backup tapes.  NOTE: Offices with "cold spare" backups may prefer to select a recent shelf backup disk circuit pack. See the "Making a Backup Disk" section in Chapter 9 for more details.
12	If both MHD 0 and MHD 1 are thought to contain similar bootable images, one MHD will be loaded from the tape at this time. If both system MHDs are not bootable (that is, new drives, disks initialized, or a software backup disk was automatically configured), both MHDs will be loaded from the tape at this time. One MHD will have the most recent set of backup tapes. The other MHD will have the second most recent set of backup tapes. The MHDs to be loaded with the DIOP LDFT procedure depends on the circumstances (often unique) which lead to DIOP. Determine the first MHD to be loaded using the following criteria: ▪ If any MHD was replaced (but not by a software backup disk) or otherwise rendered unusable, it will be loaded from backup tapes. ▪ If the software backup disk is available and has been created recently, it will be used during the recovery strategy in the "Recovery Decision Procedure" section. If a software backup disk has been used to replace a system disk, do not reload it. Do, however, load the MHD on the DFC opposite the software backup disk. ▪ The MHD that had the highest incidence of errors. ▪ The last active MHD (marked "OOS FDL" on MTTY Page 120) should not be considered for loading first unless it is known to be bad. This disk would contain the most likely to boot disk image. ▪ If backup tapes or software backup disks are significantly (6 months) out of date, seek technical assistance. Every attempt should be made to save the original data on at least one of the original system disks for use during post recovery operations.
13	Use the DIOP LDFT procedure in the "Load Disk from Tape Procedure" section to reload an MHD selected in Step 12 with the set of tapes selected in Step 11.

Step	Procedure
14	If both system MHDs were not bootable (that is, new drives or disks initialized), the other MHD will be loaded at this time with the second most recent set of backup tapes. Was the other system MHD rendered unbootable? If YES, continue to the next step. If NO, go to the "Recovery Decision Procedure" section.
15	Obtain the next most recent set of backup tapes.
16	Use the DIOP LDFT procedure in the "Load Disk from Tape Procedure" section to reload the next MHD selected using the criteria listed in Step 12.
17	Go to the "Recovery Decision Procedure" section.

Load Disk from Tape Procedure

This procedure assumes that the technical support has selected which tapes to load and which MHD to load them on. If this is not true, return to the "Prepare DFCs and MHDs for Recovery Procedure" section. Installing office tapes* CANNOT be loaded while in DIOP.

The Tape-Only Program (TOP) tape will not be required for these load procedures.

⇒ NOTE:

Maintain a record of which backup tapes are loaded on each drive so that post recovery steps to resolve problems using "old" application data will be easier to resolve.

Step	Procedure
1	Mount text backup tape on the tape drive.
2	On MTTY Page 120, enter command 620,x y z ; for example, 620,0 /dev/mt00 b for <i>UNIX</i> RTR Operating System Release 6.3 and earlier 620,0 /dev/mt00 both for <i>UNIX</i> RTR Operating System Release 6.5 and later Where x = 0 , MHD 0 is to be loaded. = 1 , MHD 1 is to be loaded. y = /dev/mt00 for high-density tapes. = /dev/mt08 for low-density tapes. For <i>UNIX</i> RTR Operating System Release 6.3 and earlier, z = b for loading 3B20D/3B21D computers text and Office Dependent Data (ODD) tapes. = g for loading 3B20D/3B21D computers text only. = d for loading ODD only. For <i>UNIX</i> RTR Operating System Release 6.5 and later, z = both for loading 3B20D/3B21D computers text and ODD tapes.

* Tapes used for the initial office installation CANNOT be loaded while in DIOP, because they do not contain the ECD and the System Generation (SG) database.

Step	Procedure
	= gen for loading 3B20D/3B21D computers text only. = dbonly for loading ODD only. Response: Tape version PRM_0 E15x 5400 xxxx xxxx xx xx xx or LOAD MHD x FROM TD y STARTED Periodically while tape is being read: PRM_0 E15x 57xx xxxx xxxx xx xx xx or LOAD MHD x FROM TD y IN PROGRESS Response when the tape rewinds: PRM_0 E15x 58xx xxxx xxxx xx xx xx or LOAD MHD x FROM TD y MOUNT NEXT TAPE ⇒ NOTE 1: If there is more than one text backup tape, mount the next text tape in the sequence and enter 620,c (for <i>UNIX</i> RTR Operating System Release 6.3 and earlier) or 620,cont (for <i>UNIX</i> RTR Operating System Release 6.5 and later). ⇒ NOTE 2: Failures of the tape operation must be restarted with a 620,x /dev/mtxx b (not 620,c for <i>UNIX</i> RTR Operating System Release 6.3 and earlier nor 620,cont for <i>UNIX</i> RTR Operating System Release 6.5 and later). 3 Remove text backup tape from tape drive; mount ODD backup tape.

Step	Procedure
4	Enter 620,c (for <i>UNIX</i> RTR Operating System Release 6.3 and earlier). Enter 620,cont (for <i>UNIX</i> RTR Operating System Release 6.5 and later). Response: Tape version PRM_0 E15x 54xx xxxx xxxx xx xx xx or LOAD MHD x FROM TD y STARTED Periodically while tape is being read: PRM_0 E15x 57xx xxxx xxxx xx xx xx or LOAD MHD x FROM TD y IN PROGRESS Response when the tape rewinds: PRM_0 E15x 59xx xxxx xxxx xx xx xx or LOAD MHD x FROM TD y COMPLETED ⇒ NOTE: Failures of the tape operation must be restarted with a 620,x /dev/mtxx d (not 620,c for <i>UNIX</i> RTR Operating System Release 6.3 and earlier nor 620,cont for <i>UNIX</i> RTR Operating System Release 6.5 and later).
5	Was PRM_0 E15x 59xx xxxx xxxx xx xx xx (or LOAD MHD x FROM TD y COMPLETED) output? If YES, return to the “Prepare DFCs and MHDs for Recovery Procedure” section. If NO, continue to the next step.
6	Was PRM_0 E15x 58xx xxxx xx xx xx (or LOAD MHD x FROM TD y MOUNT NEXT TAPE) output after the tape rewind? If YES, mount the next tape and repeat the procedure beginning at Step 2. If NO, seek technical assistance or refer to 254-303-111, <i>PDS Output Messages Manual</i>, or 254-303-113, <i>MML Output Messages Manual</i>.

Recovery Decision Procedure

This procedure assumes that all system disks (and software backup disks, if equipped) have been repaired and loaded for the recovery attempt. It provides guidelines for determining when to change the configuration of the system (by swapping ID plugs/ID switches or using the Alternate Boot Disk feature) and when to proceed with the recovery attempt.

Step	Procedure
1	On the MTTY EAI, enter command 14 to clear the EAI.
2	On the MTTY EAI, enter command 12 to force the active CU online.
3	Is the Alternate Boot Disk (ABD) feature available (EAI screen has options 28 and 29)? If YES , go to Step 16. If NO , continue to the next step.
4	Is a software backup disk with a boot image equipped in this office? If YES , continue to the next step. If NO , go to Step 14.
5	For each DFC with a software backup disk, decide which disk (system or software backup) should be used first during the recovery attempt. The disk containing the most current data should normally be selected. Does the selected disk have an ID plug/ID switch labeled 0 on the control panel of the MHD? If YES , go to Step 13. If NO , continue to the next step.
6	Which software backup disk is to be switched to the system disk? If MHD 14 on DFC 0, continue to the next step. If MHD 15 on DFC 1, go to Step 11.
7	Power down both MHD 0 and MHD 14.
8	Exchange the identities of the primary disk (MHD 0) and the software backup disk (MHD 14). The primary disk will take the identity of the software backup disk. The software backup disk will take the identity of the primary disk. If SMD , change MHD 0 ID plug to 7 and change MHD 14 ID plug to 0 . If SCSI , change MHD 0 ID switch to 3 and change MHD 14 ID switch to 0 .

Step	Procedure																				
9	Power up MHD 0 and MHD 14.																				
10	Go to Step 14.																				
11	Power down both MHD 1 and MHD 15.																				
	 NOTE: MHD 15 on DFC 1 was selected to be switched to the system disk.																				
12	Exchange the identities of the primary disk (MHD 1) and the software backup disk (MHD 15). The primary disk will take the identity of the software backup disk. The software backup disk will take the identity of the primary disk. If SMD, change MHD 1 ID plug to 7 and change MHD 15 ID plug to 0. If SCSI, change MHD 1 ID switch to 3 and change MHD 15 ID switch to 0.																				
13	Power up MHD 1 and MHD 15.																				
14	The DFCs will recognize any change in the ID plug/ID switch arrangement at boot time. If there is an MHD that is still believed to contain a good boot image, that should be the first MHD to try. The next MHD should be the next oldest, and so on. On the EAI Page, enter the command to select the first MHD to be used for the recovery attempt. The disk containing the most current data should normally be selected. Enter the poke from the following table: <table><tr><th>Chosen MHD</th><th>Active CU</th><th>Active Disk</th><th>Poke</th></tr><tr><td>0</td><td>0</td><td>Primary</td><td>20</td></tr><tr><td>0</td><td>1</td><td>Secondary</td><td>22</td></tr><tr><td>1</td><td>0</td><td>Secondary</td><td>22</td></tr><tr><td>1</td><td>1</td><td>Primary</td><td>20</td></tr></table>	Chosen MHD	Active CU	Active Disk	Poke	0	0	Primary	20	0	1	Secondary	22	1	0	Secondary	22	1	1	Primary	20
Chosen MHD	Active CU	Active Disk	Poke																		
0	0	Primary	20																		
0	1	Secondary	22																		
1	0	Secondary	22																		
1	1	Primary	20																		
15	Was the selected MHD recovered from backup tapes? If YES, enter 30 on the EAI Page and go to Step 17. If NO, go to Step 17.																				

Step	Procedure

- 16 For offices with Alternate Boot Disks backing up the boot drives (MHD 0, MHD 1), the latest (newest) warm spare drive "Boot MHD" should be selected to initiate recovery from the EAI Page without having to manipulate the MHD hardware. Perform Steps 16-a through 16-f to force the system configuration and boot using the selected "Boot MHD" designated as 'A' in Step 16-b, Step 16-c, and Step 16-d.

- a. On the MTTY EAI, enter command **28** to select the Boot MHD.
- b. Did the EAI Page respond with Select boot device? (0-15) ?
If **YES**, enter **A** and continue to Step 16-c.
If **NO**, return to Step 1 to retry the procedure.
- c. Did the EAI Page respond with Set boot devices A/1 (y/n) ?
If **YES**, enter **y** and go to Step 16-e.
If **NO**, continue to Step 16-d.
- d. Did the EAI Page respond with Set boot devices 0/A (y/n) ?
If **YES**, enter **y** and continue to Step 16-e.
If **NO**, return to Step 1 to retry the procedure.
- e. To select the proper configuration enter the poke from the following table:

Chosen MHD	Active CU	Active Disk	Poke
EVEN	0	Primary	20
EVEN	1	Secondary	22
ODD	0	Secondary	22
ODD	1	Primary	20

- f. Was the selected MHD recovered from backup tapes?
If **YES**, enter **30** on the EAI Page and continue to the next step.
If **NO**, continue to the next step.

Step	Procedure
17	Using the following criteria, decide when to perform the actual boot. The boot should be performed as soon as possible or at some other specific time (such as midnight). ▪ Are application functions (such as AMA billing) that are not necessary still working? If so, how essential is the function to the customer? ▪ Are customers sensitive to short outages? ▪ Is on-site support available for the boot attempt? ▪ Have all preparations been made? (Refer to the “Prepare DFCs and MHDs for Recovery Procedure” section.) ▪ Is the time selected appropriate if the boot attempt is NOT successful? Application essential functions (such as call processing) will stop.
18	When the time to boot comes, go to the “Boot Procedure” section.

Boot Procedure

The object of this procedure is to successfully boot the 3B20D/3B21D computers with the most recent disk possible. If a recovery fails after trying all possible configurations, the "Recovery from Backups Procedure" should be used to continue the recovery process.

This procedure assumes that the craft has prepared the system for the recovery attempt. The following preparations are assumed:

- Both system MHDs (SMD MHDs with "0" ID plugs, SCSI MHDs with "0" ID switches) will verify and contain suspected good boot images.
- The CU and the MHD to be used for the first attempt at recovery have been forced active on the EAI Page.
- The proper time for attempting a system initialization (BOOT) has been determined.
- Before booting, review and understand the "Recovery Procedure" section of this IP.
- During Step 2 application specific boot parameters can be entered. Consultation with Lucent Technologies application CTS organizations is recommended prior to starting this procedure.

Step	Procedure
1	If the system is to be booted on an MHD that was restored from a backup source, the system initialization may perform better by using application specific parameters. Should an application parameter (designated X in this procedure) be used? If YES, enter command 42;X;54; to boot the 3B20D/3B21D computers. (Yes, we really mean to enter a 54.) If NO and if this is the first boot attempt, enter command 52 to boot; otherwise, enter 54.
2	Did the EAI Page respond with Boot? (y/n)? If YES, enter y and continue to the next step. If NO, continue to the next step.

Step	Procedure																				
3	Did the initialization complete? If YES , go to the "Post Recovery Procedure" section. If NO , continue to the next step.																				
4	If the selected MHD was not restored from backup tapes (that is, MHD was an original system disk or a software backup disk), was backup root tried? If YES , continue to the next step. If NO , enter 30 to force backup root and repeat the procedure beginning at Step 1.																				
5	Reaching this point means that the 3B20D/3B21D computers have not successfully booted on the chosen MHD. Have all the MHDs been tried using the forced CU? If YES , go to Step 8. If NO , continue to the next step.																				
6	Choose and boot on the next most recent vintage MHD. On the EAI Page, enter the command to select the DFC with the next MHD to be used for the recovery attempt. Enter the poke from the following table:																				
	<table><tr><th>Chosen MHD</th><th>Active CU</th><th>Active Disk</th><th>Poke</th></tr><tr><td>0</td><td>0</td><td>Primary</td><td>20</td></tr><tr><td>0</td><td>1</td><td>Secondary</td><td>22</td></tr><tr><td>1</td><td>0</td><td>Secondary</td><td>22</td></tr><tr><td>1</td><td>1</td><td>Primary</td><td>20</td></tr></table>	Chosen MHD	Active CU	Active Disk	Poke	0	0	Primary	20	0	1	Secondary	22	1	0	Secondary	22	1	1	Primary	20
Chosen MHD	Active CU	Active Disk	Poke																		
0	0	Primary	20																		
0	1	Secondary	22																		
1	0	Secondary	22																		
1	1	Primary	20																		
7	Was the selected MHD recovered from backup tapes? If YES , enter 30 on the EAI Page and return to Step 1. If NO , restart the procedure beginning with Step 1.																				

Step	Procedure
8	Has the other CU been tried? If YES , go to the "Recovery from Backups Procedure" section, Step 2. If NO , continue to the next step.
9	On the EAI Page, enter the command to force the other CU. If CU 0 is active, enter 11 . If CU 1 is active, enter 10 .
10	Restart the procedure beginning at Step 1.

SMD DFC Repair Guidelines

Step	Procedure
1	Does the DFC power up (assuming there is power to the power pack)? If YES , go to Step 8. If NO , replace the power pack (495F1 in slot 016).
2	Does the DFC power up now? If YES , go to Step 8. If NO , replace the power switch (TN3 in slot 074).
3	Does the DFC power up now? If YES , go to Step 8. If NO , continue to Step 4.
4	Install the original 495F1 and TN3 in the DFC. Unseat (pull partially out of the frame) the remaining DFC packs (TN19, UN55, TN70, TN69, and UN64) and attempt to power up.
5	Does the DFC power up now? If YES , continue to Step 6. If NO , seek technical assistance.
6	Remove the power, insert the circuit packs one at a time, and power the DFC up again to isolate the faulty pack and pack location.
7	Was the faulty pack isolated? If YES , replace it and continue to Step 8. If NO , seek technical assistance.
8	Restore the DFC using 602,x,UCL . Does the DFC restore? If YES , return to the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , replace DFC PIC microstore circuit pack (TN19 in slot 022) and DFC PIC circuit pack (UN55 in slot 044).
9	Does the DFC restore? If YES , return to the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , replace Bus Interface Controller (BIC) circuit pack (TN70 in slot 052).

Step	Procedure
10	Does the DFC restore? If YES , return to the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , replace D2SBS circuit pack (TN69 in slot 058).
11	Does the DFC restore? If YES , return to the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , seek technical assistance.

SMD MHD Repair Guidelines

Repairing essential system disks generally results in further degradation of the system (loss of additional active operational MHDs). Repair actions should NOT be performed without reading and understanding the following comments:

- A DFC must be removed from service and powered down whenever replacing an MHD or MHD cables.
- In order to remove the DFC from service, all MHDs on that DFC must be out of service. First remove the MHD power and then remove the DFC power by using the manual override button.
- Any MHDs removed from service (that is, while swapping ID plugs) cannot be returned to service while in DIOP.
- If an AMA disk is active, removing the MHD (or DFC) will terminate AMA recording on that MHD.
- The INIT command (**600,x**; x = MHD number) WILL ERASE all the data on the disk and reformat the disk. It should not be executed indiscriminately.
- When a 340-MB disk is replaced with a new drive, an INIT NEW command (**600,x,N**) must be used on the drive.

The failure of a maintenance command will result in DFC error completion codes being printed on the ROP. These codes may suggest the appropriate repair action.

REPT DFC x ERROR COMPL CODE X'ccxxxxxx MHD STATUS X'x06xxxxxx

Where: cc = Error codes from the following:

Disk Errors		
Code	Error	Action
81	PIC ram write error	Cables, ID Plug, Power
83	Seek error on disk	
85	No index detected on disk	
89	MHD status error	
8a	Uncorrectable data error	
8b	Uncorrectable data error	
8d	Disk boundary overflow	
8f	Head sector mismatch	
91	Wrong disk selected	
93	No disk selected	
94	Head positioning error	
95	Head positioning error	
97	Head select error	

The previous error codes can be resolved by initializing the disk (**600,x**) or replacing the disk drive.

DFC Errors		
Code	Error	Action
99	BIC parity error	Replace TN69 or TN70
9d	PIC error	Replace TN19 or UN55
9f	DMA setup error	Replace TN19, TN69, TN70, or UN55
a1	Early EOT from DMA	Replace TN19, TN69, TN70, or UN55
a3	No EOT from DMA	Replace TN19, TN69, TN70, or UN55
a7	PSDI FIFO parity error	Replace TN64
a9	PSDI FIFO underflow	Replace TN64 (possible drive problem)
ab	PSDI FIFO overflow	Replace TN19, TN64, or UN55
ad	PSDI ACK failure	
af	Extra data in PSDI	
b1	Disk data buffer error	
b3	Main store address error	
b5	Disk time out	Usually defective disk. Replace TN19 or UN55

Unless otherwise noted in the **Action** column, suspect a problem in the DFC (TN69, TN70, TN19, or UN55).

Defect Management Errors	
Code	Error
bb	Uncorrectable data error in manufacturers defect table
bd	Defect table overflow
bf	Command for defective disk
c1	Unable to create defect table on the disk
c3	Manufacturer specified more than maximum defects
c5	Uncorrectable data error on area that is to be error free
c7	Invalid data in defect table
c9	Invalid data in RAM
cb	Invalid manufacturers defect table
cd	Uncorrectable data error in defect table
cf	Unable to read defect table
d1	Unable to read manufacturers defect table
d3	Unable to write defect table on the disk

The Defect Management Error can be fixed by **600,x,N** to initialize the disk. Otherwise, the disk must be replaced since the defect table cannot be reloaded while in DIOP.

Autonomous response codes may also suggest the appropriate repair action.

REPT DFC x AUTONOMOUS RESP. code

Autonomous Response Error Codes		
Code	Error	Action
100	Onboard diag failure	DFC fault, replace TN19 or UN55 INIT disk or replace
200	Onboard exerciser failure	
300	Sysgen command completed	
400	Boot command completed	
500	DFC idle	
600	Illegal command	
700	High temp warning	

SCSI DFC Repair Guidelines

3B20D Computer

Step	Procedure
1	Does the DFC power up (assuming there is power to the power pack)? If YES , go to Step 8. If NO , replace the power pack (410AB).
2	Does the DFC power up now? If YES , go to Step 8. If NO , replace the power switch (TN5B).
3	Does the DFC power up now? If YES , go to Step 8. If NO , continue to the next step.
4	Install the original 410AB and TN5B in the DFC. Unseat (pull partially out of the frame) the remaining DFC packs (TN69, UN294, and TN2116) and attempt to power up.
5	Does the DFC power up now? If YES , continue to the next step. If NO , seek technical assistance.
6	Remove the power, insert the circuit packs one at a time, and power the DFC up again to isolate the faulty pack and pack location.

Step	Procedure
7	Was the faulty pack isolated? If YES , replace it and continue to the next step. If NO , seek technical assistance.
8	Restore the DFC using 602,x,UCL . Does the DFC restore? If YES , return to Step 3 in the “Prepare DFCs and MHDs for Recovery Procedure” section. If NO , replace DFC HA circuit packs (TN294 and UN2116).
9	Does the DFC restore? If YES , return to Step 3 in the “Prepare DFCs and MHDs for Recovery Procedure” section. If NO , replace D2SBS circuit pack (TN69).
10	Does the DFC restore? If YES , return to Step 3 in the “Prepare DFCs and MHDs for Recovery Procedure” section. If NO , seek technical assistance.

3B21D Computer

Step	Procedure
1	Does the DFC power up (assuming there is power to the power pack)? If YES , go to Step 5. If NO , replace the power pack (410AA), if equipped.
2	Does the DFC power up now? If YES , go to Step 5. If NO , replace the DFC A (either the UN373, UN373B, UN580, or UN580B circuit pack) equipped on your 3B21D computer that provides the power control function.
3	Does the DFC power up now? If YES , go to Step 5. If NO , replace the DFC B (TN2116), if equipped.
4	Does the DFC power up now? If YES , continue to the next step. If NO , seek technical assistance.
5	Restore the DFC using 602,x,UCL . Does the DFC restore? If YES , return to Step 3 in the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , replace the Host Adapter (HA) circuit packs (DFC B or TN2116).
6	Does the DFC restore? If YES , return to Step 3 in the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , replace the DFCA circuit pack (UN373).
7	Does the DFC restore? If YES , return to Step 3 in the "Prepare DFCs and MHDs for Recovery Procedure" section. If NO , seek technical assistance.

SCSI MHD Repair Guidelines

Repairing essential system disks generally results in further degradation of the system (loss of additional active operational MHDs). Repair actions should NOT be performed without reading and understanding the following comments:

- An SBUS must be removed from service (**604,x,RMV,UCL**) whenever replacing an MHD or SBUS cables.
- In order to remove the SBUS from service, all MHDs on that SBUS (not the DFC) must be out of service. First remove the MHD power and then remove the SBUS using **604,x,RMV,UCL** pokes.
- If a simplex NOMANRMV disk is on that SBUS, that DFC must be powered down.
- Any MHDs removed from service (that is, while swapping ID switches) cannot be returned to service while in DIOP.
- If the ID switch is changed, the drive **MUST** be power cycled.
- If an AMA disk is active, removing the MHD (or DFC) will terminate AMA recording on that MHD.
- The INIT command (**600,x**; x = MHD number) will erase all the data on the disk and reformat the disk. It should not be executed indiscriminately.
- When a SCSI disk is replaced with a new drive, an INIT command (**600,x**) must be used on the drive.
- If a 600-MB SCSI drive is replacing a 322-MB SCSI drive, ECD changes must be entered after system recovery.

The failure of a maintenance command will result in DFC error completion codes being printed on the ROP.

Loading Disk from Tape

The Load Disk from Tape (LDFT) program provides the capability to load software onto a boot disk from tape. LDFT, itself, resides on a tape and is executed from the Emergency Action Page via the LDTAPE-0 (55) or LDTAPE-1 (56) commands. Once the LDFT program is loaded, it determines whether the tape is a single volume tape or a multivolume tape. A single volume tape is distinguished from a multivolume tape by the contents of the tape. A single volume tape will consist of the LDFT program and no other loadable software. The software to be loaded is on separate tapes. The LDFT-only tape is known as the TOP tape. Multivolume tapes contain the TOP and at least one loadable volume of data containing loadable software. Volumes on a multivolume tape are analogous to the individual tapes used when loading single volume tapes. The **COPY:BKDISK** command is used to create the tapes in both single volume and multivolume formats; LDFT only reads the tapes and loads the contents to the disk.



NOTE 1:

LDFT should be used only as a last resort during a recovery situation. Using LDFT will cause complete loss of 3B/RTR system functionality. Guidelines for using LDFT are provided in the "Recovery from Backups Procedure" section.



NOTE 2:

LDTAPE-0 implies that IOP 0/DFC 0 and the MT device associated with this IOP/DFC are functional.

LDTAPE-1 implies that IOP 1/DFC 1 and the MT device associated with this IOP/DFC are functional.



NOTE 3:

Multivolume tapes require the use of a Digital Audio Tape (DAT) unit and are supported only on the 3B21D computer.



NOTE 4:

Application parameters to the Emergency Action Page Command 42 are shown in lower case throughout this section. However, the LDFT program is not case-sensitive and will accept any application parameter in the upper case.



NOTE 5:

LDFT should **NOT** be used to repair nonservice effecting file system damage such as individual files; instead, refer to Chapter 9, "File-System Maintenance."

Loading Single Volume Tapes (Formerly the Standard LDFT Procedure)

This section provides general instructions for loading data from tape to disk using single volume tapes.

To initiate the LDFT feature, use the following procedure:

Step	Procedure
1	Force a CU on-line (Emergency Action Page Command 10, 11).
2	Force the CU to use the primary or secondary MHD (Emergency Action Page Command 20, 22).
3	Mount the TOP tape onto a tape unit associated with LDTAPE-0 or LDTAPE-1.
4	Boot the LDFT program: If the TOP tape is mounted on tape unit 0, boot LDTAPE-0 (Emergency Action Page Command 55). If the TOP tape is mounted on tape unit 1, boot LDTAPE-1 (Emergency Action Page Command 56).
5	Observe that the following PRMs are alternately generated after the boot of the TOP tape: PRM:E1da-51 and PRM:E1da-99 PRM:E1da-51 instructs the user to mount the first tape that has disk data. PRM:E1da-99 is provided so that alternate messages are displayed on the Emergency Action Page.
6	The following tape sequences are supported: Sequence (a) assumes that the release and database information exist on separate tapes. Sequence (b) should be followed if the release and database information are on the same tape or if only the release information is being loaded. Sequence (c) should be followed if only the database information is being loaded. Notes 1 through 5 (describing disk formatting, tape version data, and PRMs) are provided at the end of this procedure.

Step	Procedure

a. Normal LDFT (release and database tapes).

- (1) Load the release tapes first. If the disk is to be formatted, refer to Note 1 at the end of this procedure. After each release tape is mounted, initiate the LDFT feature again as specified in Step 4 of this procedure. While loading data, several PRMs are generated as specified in Notes 2 and 3 at the end of this procedure.
- (2) After all the release data is loaded, PRM:E1da-59 indicates end of data and which disk partition will be used to boot the computer (digits 14 and 15: 0 = primary, nonzero = backup). Make a note of the disk partition for later use.
- (3) The computer is not to be booted after loading the release tapes; instead, the database tapes are loaded.
- (4) After each database tape is mounted, initiate the LDFT feature as specified in Step 4 of this procedure. While loading data, several PRMs are generated as specified in Notes 2 and 3 at the end of this procedure.
- (5) After all the database tapes are loaded, PRM:E1da-59 is alternately output with an All Seems Well (ASW) PRM:E1da-DD.
- (6) Select on the Emergency Action Page the noted disk partition.

If primary, Command 31
If backup, Command 30.

The computer should now be booted.

b. Release only or merge LDFT. The merge option combines both release data and database data into one sequence of tapes.

- (1) If the disk is to be formatted, refer to Note 1 at the end of this procedure. After each release tape is mounted, initiate the LDFT feature as specified in Step 4 of this procedure. While loading data, several PRMs are generated as specified in Notes 2 and 3 at the end of this procedure.
- (2) After all the release data is loaded, PRM:E1da-59 indicates end of data and which disk partition will be used to boot the computer (digits 14 and 15: 0 = primary, nonzero = backup). Make a note of the disk partition for later use.

Step	Procedure
	(3) Select on the Emergency Action Page the noted disk partition. If primary, Command 31 If backup, Command 30. The computer should now be booted. c. Database only LDFT. (1) The character "d" must be specified as the EAI application parameter (Command 42) prior to initiating LDFT to load the first database tape. (2) After each database tape is mounted, initiate the LDFT feature as specified in Step 4 of this procedure. While loading data, several PRMs are generated as specified in Notes 2 and 3 at the end of this procedure. (3) After all the database tapes are loaded, PRM:E1da-59 indicates end of data and which disk partition will be used to boot the computer (digits 14 and 15: 0 = primary, nonzero = backup). Make a note of the disk partition for later use. (4) Select on the Emergency Action Page the noted disk partition. If primary, Command 31 If backup, Command 30. The computer should now be booted. 7 To recover the system after loading the software, the "BOOT + MEM" command (54) must be executed.

⇒ NOTE 1:

The disk should be formatted prior to loading it with any data. To format a disk, mount the first release tape and request an application parameter (Command 42), parameter "i." Initiate the LDFT feature again. (Refer to Step 4 of this procedure.)

For SMD Only

Fixed media disks (340 MB) may have defects. Consequently, the DFC uses a defect table to map around these areas. If the defect table does not yet exist, the "i" disk format will fail and PRM:F1da-21 will be output. It is then necessary to request an application parameter (Command 42), parameter "n," with the first release tape still mounted. Initiate the LDFT feature again.

Step	Procedure

(Refer to Step 4 of this procedure.)

A progress PRM:E1da-88 is output showing the total number of tracks formatted so far. After the disk format is complete, the first release tape automatically begins to be read.

For SCSI Only

For SCSI disks, only the “i” parameter will be used for disk formatting. If the “n” is used with SCSI, a warning PRM:E1da-89 will be printed indicating an illegal parameter has been set. The “n” parameter will default to “i.”

A progress PRM:E1da-88 is output. After the disk format is complete, the first release tape automatically begins to be read.

⇒ **NOTE 2:**

The tape version data is output as PRM:E1da-54. During the reload, PRM:E1da-74 should be received after PRM:E1da-54. PRM:E1da-74 reports the LDFT program version. As the tape is read, a progress PRM:E1da-57 is output showing the total number of tape blocks read so far from the currently mounted tape.

⇒ **NOTE 3:**

If multiple tapes must be loaded, PRM:E1da-58 is output indicating the tape version used and the next tape sequence number.

⇒ **NOTE 4:**

Failing PRMs are generated as necessary. Refer to 254-303-103, *Processor Recovery Messages Guide*.

⇒ **NOTE 5:**

For separate generic and database tape sequences, the Volume Table of Contents (VTOC) on all tapes must match. If a common generic tape is being used, the database tape must have a common VTOC on it in order to boot the system.

Loading Multivolume Tapes

This section provides instructions for loading data from tape to disk using a multivolume tape.

The multivolume tape has the contents of the TOP tape and at least one additional volume. To load a volume to the boot disk, the user must enter a specific sequence of instructions via the application parameter command 42 on the Emergency Action Page.

To initiate the LDFT feature, perform the following procedure:

Step	Procedure
1	Force a CU on-line (Emergency Action Page Command 10, 11).
2	Force the CU to use the primary or secondary MHD (Emergency Action Page Command 20, 22).
3	Mount the multivolume tape onto a tape unit associated with LDTAPE-0 or LDTAPE-1.
4	Enter any preboot application parameters (Emergency Action Page Command 42) or clear the application parameter (Emergency Action Page Command 43).

Suggested preboot application parameters that could be entered are as follows:

“h” instructs LDFT to display volume information immediately after LDFT is loaded and executing.

The first parameter in any of the command sequences listed in Step 7 can also be entered but is not recommended.



CAUTION:

If any application parameter is displayed on the Emergency Action Page when the LDFT program is booted, the LDFT program will read that application parameter during its initialization and process it accordingly.

5	Boot the LDFT program: If the multivolume tape is mounted on tape unit 0, boot LDTAPE-0 (Emergency Action Page Command 55). If the multivolume tape is mounted on tape unit 1, boot LDTAPE-1 (Emergency Action Page Command 56).

Step	Procedure
6	Observe that the following PRMs are generated following the boot of the multivolume tape: PRM:E1da-30 followed by PRM:E1da-33 If the “h” application parameter was entered before booting, PRM:E1da-35 will be generated for each volume on the DAT before PRM:E1da-33.
7	As long as the 3B21D computer is not rebooted (Emergency Action Page Commands 51 through 56), the LDFT program remains active. PRM:E1da-33 is generated approximately every 15 seconds while the LDFT program waits for input commands from the Emergency Action Page. PRMs specifically for this feature and all LDFT failing PRMs are described in 254-303-103, <i>Processor Recovery Messages Guide</i>. The following command sequences are supported by LDFT: - To clear the EAI Application Parameter buffer, enter the following poke: Poke 43 Wait for PRM:E1da-33 to acknowledge this poke. - To dump the tape headers after LDFT has been loaded, enter the following poke: Poke 42 h After the header PRMs (PRM:E1da-35) are generated, wait for PRM:E1da-33 to acknowledge that LDFT is ready for the next application parameter to be entered. - To load the disk with a user-selected volume, the following sequence must be entered (see Notes 1 and 3): - Poke 42 s Wait for PRM:E1da-33 to acknowledge this poke. - Poke 42 <session number> Wait for PRM:E1da-33 to acknowledge this poke. - Poke 42 v Wait for PRM:E1da-33 to acknowledge this poke. - Poke 42 <volume number> PRM:E1da-36 will be generated to indicate whether the volume has been found on the tape. If it has, loading will begin immediately after this PRM is generated. After this task has been completed, LDFT returns to the PRM:E1da-33 wait loop.

Step	Procedure
	d. To format the disk before loading a user-selected volume, the following sequence must be entered (see Notes 1, 2, and 3): (1) Poke 42 i Wait for PRM:E1da-33 to acknowledge this poke. (2) Poke 42 s Wait for PRM:E1da-33 to acknowledge this poke. (3) Poke 42 <session number> Wait for PRM:E1da-33 to acknowledge this poke. (4) Poke 42 v Wait for PRM:E1da-33 to acknowledge this poke. (5) Poke 42 <volume number> PRM:E1da-36 will be generated to indicate whether the volume has been found on the tape. If it has, disk formatting and loading will begin immediately after this PRM is generated. After this task has been completed, LDFT returns to the PRM:E1da-33 wait loop. e. To load a disk with a database-only volume, the following sequence must be entered (see Notes 1 and 3): (1) Poke 42 d Wait for PRM:E1da-33 to acknowledge this poke. (2) Poke 42 s Wait for PRM:E1da-33 to acknowledge this poke. (3) Poke 42 <session number> Wait for PRM:E1da-33 to acknowledge this poke. (4) Poke 42 v Wait for PRM:E1da-33 to acknowledge this poke. (5) Poke 42 <volume number> PRM:E1da-36 will be generated to indicate whether the volume has been found on the tape. If it has, database loading will begin immediately after this PRM is generated. After this task has been completed, LDFT returns to the PRM:E1da-33 wait loop.
8	To recover the system after loading the software, the "BOOT + MEM" command (54) must be executed.

Step	Procedure

⇒ **NOTE 1:**

If a sequence is not followed as shown, an error PRM will be generated and the entire sequence will have to be reentered from the beginning.

⇒ **NOTE 2:**

A progress PRM:E1da-88 is output periodically while the disk is being formatted. After the disk format is complete, the user-selected volume automatically begins to be read and loaded onto the boot disk.

For SCSI disks, the "i" EAI Application Parameter is used for disk formatting. LDFT also supports the SMD-only "n" parameters, but defaults to the "i" parameter when "n" is used with SCSI. A warning PRM:E1da-89 will be printed indicating an illegal parameter has been set.

⇒ **NOTE 3:**

PRM:E1da-74 reports the LDFT program version. As the volume is read, a progress PRM:E1da-57 is output showing the number of tape blocks read from the volume being read. Once the volume has been loaded to disk, PRM:E1da-59, followed by All Seems Well (ASW) PRM:E1da-DD is output.

Contents

Introduction	4-1
Description of Error Reports	4-2
■ Error Log Files	4-2
■ Control Unit Error Messages	4-2
Memory History Log File	4-3
Error Interrupt Handler Log File	4-3
■ Automatic Postmortem Dumps	4-4
HEADING	4-5
INITIALIZATION MESSAGE	4-6
REQUESTED INITIALIZATION PARAMETERS	4-9
BOOT DEVICES	4-10
EMERGENCY ACTION INTERFACE BUFFER	4-10
TIMER	4-13
GENERAL REGISTERS	4-13
FAULTY CU REGISTERS	4-14
INTERRUPT STACK SAVED STATE	4-14
OFF-LINE REGISTERS	4-15
CACHE REGISTERS (3B21D Computer Only)	4-15
MAIN STORE REGISTERS	4-16
REAL-TIME CLOCK	4-16
■ Processor Recovery Messages	4-16
Error Report Analysis	4-18
■ Procedure 1 – Memory History File	4-26
■ Procedure 2 – On-Line Hardware – DMA Detected	4-40
■ Procedure 3 – On-Line Hardware – DMA Detected	4-41

Contents

■ Procedure 4 – On-Line Hardware – DMA Detected	4-42
■ Procedure 5 – On-Line Hardware Fault	4-43
■ Procedure 6 – Off-Line Hardware Failures	4-45
■ Procedure 7 – Software Fault	4-46
■ Error Interrupt Handler File	4-47
On-Line Hardware Faults	4-49
Off-Line Hardware Faults	4-52
On-Line Software Faults	4-53
■ Automatic Postmortem Dump Analysis	4-54
■ Memory Management Exceptions	4-73
■ Interpreting Memory Failures	4-74
■ Responding to Nondiagnosable Memory Errors	4-76
■ Responding to an Emergency Dump Partition	4-81
■ Processor Recovery Message Analysis	4-82
Register Layouts	4-82
■ Bit and Byte Numbering	4-83
■ Cache Registers (3B21D Computer)	4-84
■ Data Path Registers	4-91
■ DMA Channel Status Register	4-93
■ DMAC Status Register	4-94
■ DSCH Channel Status Register	4-101
■ Error Register	4-102
■ Error Register Description	4-104
3B20D Computer Error Register Description	4-105
3B21D Computer Error Register Description	4-107
■ Hardware Status Registers	4-122
■ Hardware Status Register Maintenance States	4-125
■ Interrupt Source Register	4-126
■ Main Store Controller Error Register 1 (3B20D Computer)	4-127
■ Main Store Controller Error Register 2 (3B20D Computer)	4-128
■ Main Store Controller Trapped Address Register (3B20D Computer)	4-129
■ Memory Registers	4-130
■ Microinterrupt Error Register	4-136
■ Microinterrupt Error Register 1	4-137
■ Processor Status Word Register	4-138
■ Pulse Point Register	4-140

Contents

■ Store Control Register	4-141
■ System Status Register	4-143

Introduction

This chapter gives an overview of the Control Unit (CU) error messages and automatic postmortem dumps for *UNIX*® Real-Time Reliable (RTR) Operating System Release 1 and later. It will help the user interpret the different types of error messages generated by the automatic fault recovery features of Release 1 and later. Specifically, the message data presented in this chapter is collected by the Error Interrupt Handler (EIH) and the Processor Initialization (PINIT) routine. The EIH is a kernel process and the PINIT is the first program to execute (in physical memory) after a system Maintenance Reset Function (MRF).

This chapter should be used together with the following information products (IPs):

- 254-303-110, *PDS Input Messages Manual*
- 254-303-111, *PDS Output Messages Manual*
- 254-303-112, *MML Input Messages Manual*
- 254-303-113, *MML Output Messages Manual*
- 254-303-103, *Processor Recovery Messages Guide*.

The Input Messages/Output Messages (IM/OM) manuals contain explanations of the many types of error messages that are sent to the spooler craft process. These messages are output when the system is in a stable state and capable of

performing normal processing activities. The error messages presented in this chapter are of this type. The information for these messages is collected at the time the error or recovery action takes place. When the system sufficiently recovers, the data is formatted and sent to the craft spooler process for output to the Receive-Only Printer (ROP).

In addition to the IM/OM manuals, the reader is encouraged to become familiar with 254-303-103, *Processor Recovery Messages Guide*. While error messages described in the IM/OM manuals are of use when the system has achieved normal processing capability, Processor Recovery Messages (PRMs) may be output in the face of the most severe system problems. Most system problems can be analyzed using the PRMs.

Description of Error Reports

This section describes the generation and content of the error messages and automatic postmortem dumps.

Error Log Files

There are three error log files:

- Memory Log File (*MEMLOG*)
- Error Interrupt Handler (EIH) Log File (*ERLOG*)
- Automatic Postmortem Dump Log File (*PMLOG*).

Entries in *MEMLOG* and *ERLOG* are generated by the output message REPT CU. Entries in *PMLOG* are generated following any Maintenance Reset Function (MRF) on the 3B20D and 3B21D computers. (An MRF means that the computer has experienced a system reset, that is, executing the CU microcode's Maintenance Reset Function.)

A sequence number is associated with each log file entry. All log file entries use the same sequence counter, so the order in which a set of entries occurs can be determined from the sequence numbers. This is useful when many errors lead to a processor initialization. Each entry also has a date and time stamp to relate log entries to other system printouts.

Control Unit Error Messages

Error interrupts are nonfatal errors detected by the computer, occurring in either the on-line or standby CU. (If preset thresholds are exceeded, error interrupts can escalate and cause a processor switch or an initialization.) When error interrupts occur, the associated information is written into *MEMLOG* and/or *ERLOG*. As stated previously, these log file entries are generated with the

REPT CU message which prints on the ROP and appears on the Maintenance Teletypewriter (MTTY) when an error interrupt has occurred. The REPT CU message shows the error interrupt type, fault correction information, and availability of supplemental data. The REPT CU message shows (via a sequence number in the message) that supplemental data is available. This supplemental data is in the form of an error log file entry.

The log file entry can be accessed by using the **OP:LOG** input message. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, and the sequence number given in the REPT CU message. Detailed procedures for accessing and interpreting *MEMLOG* and *ERLOG* file entries are described later in the "Error Report Analysis" section.

Memory History Log File

The *MEMLOG* file contains the REPT CU supplementary data particular to memory failures. Memory failure data should be analyzed using the procedure given in the "Error Report Analysis" section to determine the faulty pack. The faulty pack should be replaced, and diagnostics should then be run.

Error Interrupt Handler Log File

The *ERLOG* file contains supplementary error interrupt data detected by the EIH. The possible error sources are as follows:

- On-line hardware
- Off-line hardware
- Software.

The on-line and off-line processors are monitored for a variety of hardware faults. The fault types and corrective actions, as well as the direct memory access information, are described in the "Error Report Analysis" section.

Software faults, unlike most hardware faults, cannot generally be corrected by a user. However, to aid in reporting software error entry information to your technical assistance organization, these error types are also described in the "Error Report Analysis" section.

Software fault entries provide information about the state and identity of the program that encounters the fault. The program state at the time of the interrupt is shown in the section of the file entry called SAVE STATE. This information is in the form of the following register values: Program Address (PA) counter value, Processor Status Word (PSW) value, Primary Segmentation Base Register (PSBR) value, and Secondary Segmentation Base Register (SSBR) value.

NOTE:

The PSW register layout is in the "Processor Status Word Register" section. The "General Registers" section has more information.

After the register information, the Utility Identification (UID) and the process number are printed. These values together identify the program that encountered the fault.

Automatic Postmortem Dumps

Maintenance interrupts are a higher level interrupt than an error interrupt. They are caused by hardware or software sources and involve system initialization. This initialization triggers an output of a postmortem dump. The dump contains critical register and system data collected at the time the MRF occurred. The data provided in each entry consists of most of the critical hardware registers from both CUs. Some nonhardware type information dealing with the past and present configuration of the system is also provided.

The postmortem dump is useful in determining the reason for a particular initialization. This is possible because the hardware state of the computer is frozen (by the microcode) when the initialization takes place and later collected by a low-level recovery program called **PINIT**. The **PINIT** program collects the data saved by microcode, collects other relevant data not saved by microcode, and saves both for the user level program that will print the data when the system has achieved sanity.

The start of a system initialization is usually signaled by one of the following messages output on the ROP:

- REPT CU 0 MAINTENANCE INTERRUPT
- REPT CU 1 MAINTENANCE INTERRUPT
- REPT PHASE IN PROGRESS
- START OF CU 0 RECOVERY
- START OF CU 1 RECOVERY.

The above messages indicate that the CU has been initialized (reset), regardless of the severity. The severity of the initialization (also called the phase or MRF level) suggests the amount of initialization performed to achieve system sanity. This may be an initialization of all system programs (*UNIX* RTR operating system level 1) or a more serious initialization where new copies of programs and data are read in from the disk (levels 2, 3, and 4).

If a system MRF occurs but does not recover (initialize completely) then another MRF will occur and begin another initialization. Each subsequent MRF will be at a higher level of initialization or will try different combinations of essential hardware (CUs and DISKS).

The postmortem dump has a log file associated with it that contains a recent history of initializations that have taken place. The name of this log file directory is *PMLOG*, and it may be found in the */etc/log* directory along with the previously

discussed log files. Just as for *MEMLOG* and *ERLOG*, the current contents of the *PMLOG* can be printed using the **OP:LOG** command, the format of which is described in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

If printed automatically or by manual request, the format and information of a postmortem dump is the same. The reports are slightly different for the 3B20D and 3B21D computers. The postmortem dump is composed of the following sections.

- HEADING
- INITIALIZATION MESSAGE
- REQUESTED INITIALIZATION PARAMETERS
- BOOT DEVICES
- EMERGENCY ACTION INTERFACE BUFFER
- TIMER
- GENERAL REGISTERS
- FAULTY CU REGISTERS
- INTERRUPT STACK SAVED STATE
- OFF-LINE REGISTERS
- CACHE REGISTERS (3B21D Computer Only)
- MAIN STORE REGISTERS
- REAL-TIME CLOCK.

Examples of postmortem dumps are shown in the “Automatic Postmortem Dump Analysis” section.

HEADING

The two possible headings for a postmortem dump are MAINTENANCE INTERRUPT and POSTMORTEM DUMP. The decision on which heading to use is made on how long an initialization takes. Table 4-1 shows how the heading determination is made. The one exception to Table 4-1 is that a MAINTENANCE INTERRUPT accompanied by a panic dump request will be reclassified to a phase.

Table 4-1. Heading Determination

Memory Copied	Source	Phase Requested	Result
yes	*	*	postmortem dump
*	manual	*	postmortem dump
no	hardware	*	maintenance interrupt
no	sftw (A)†	yes	postmortem dump
no	sftw (A)†	no	maintenance interrupt
no	sftw‡	*	maintenance interrupt

* Do not care.

† Application (A) started initialization; UID for application processes starts at 0X400.

‡ UNIX RTR operating system started initialization.

Sequence Number: The sequence number for the postmortem dump accompanies the postmortem dump heading. The last two digits of the sequence number are contained in the All Seems Well (ASW) Processor Recovery Message (PRM) for each major process. PRMs are sometimes printed out of order. Using the sequence number in the PRMs, one can determine the correct order of PRM output.

Time: Also in this section is the TIME OF POSTMORT/INTERRUPT field. This is the time when the initialization took place, not the time when the postmortem dump was printed.

INITIALIZATION MESSAGE

The following fields appear in this section.

- SOURCE
- ON-LINE CU
- FAULTY CU
- UNIX RTR LEVEL
- APPLICATION LEVEL.

The purpose of the INITIALIZATION MESSAGE section is to provide an indication about which processor was on-line when the initialization occurred, which processor was involved in the initialization, and the severity of the initialization. The SOURCE field refers to the source of the initialization request; it does not suggest the source of the problem.

Possible values for the SOURCE field are the following:

- **p** shows the MRF was the result of a power clear followed by a Sanity Timer time-out.
- **m** shows a manual initialization request from the maintenance terminal.
- **s** shows the initialization was requested by software.
- **h** shows the initialization is either because a hardware error was detected or the Central Control (CC) microcode requested the MRF. In many cases, the microcode initiated MRFs will be accompanied by a PRM.

When the SOURCE field contains a **p**, the postmortem dump contains little additional information since the Sanity Timer (TIMERS ONLINE) indicates the recovery was initiated by a time-out. When a Control Unit (CU) initializes because of a powerup, it will not immediately boot. Instead, the microcode will indicate in low core that a power clear (the CU may have powerup) has occurred and then halt. If the other CU is not active or cannot clear the Sanity Timer of this CU, this CU will boot and the SOURCE will be set to **p**. For unexplainable "power clear" MRFs, suspect a faulty CC power switch circuit pack or faulty Special Register (SREG) (UN2 for the 3B20D computer and KLV31 for the 3B21D computer).

When the SOURCE field contains an **m**, the postmortem dump probably contains little additional information since the System Status Register (SSR) indicates the recovery was initiated by manual intervention [Emergency Action Interface (EAI) commands at the MTTY]. However, if a false manual request is indicated (SOURCE = **m**, but no requests were made), suspect a faulty EAI, SREG0, or either Maintenance Teletypewriter Controller (MTTYC) [in either Input/Output Processor (IOP)].

When the SOURCE field contains an **h**, a hardware initialization request is indicated. A hardware initialization may be because of Stop-and-Switch (SAS) class errors, microinterrupt with interrupts blocked, or sanity-timer firing. The following listing briefly describes the problems that result in hardware initializations.

- **Stop-and-switch class errors** — A severe hardware problem was detected causing an immediate switch to the mate processor. There is no program control over this type of recovery decision.
- **Microinterrupt with interrupts blocked** — One of two things may have happened:
 - A process blocked interrupt (SSR bit 18). During this interval, if a microinterrupt occurs, microcode will not be able to dispatch the EIH to resolve the error.
 - A level 15 kernel process was executing when a microinterrupt occurred. All the bits in the interrupt mask are asserted (no interrupts can be received). This prevents the microcode from dispatching EIH, and the result will be an MRF.

- **Sanity-timer firing** — The sanity timer was not serviced by the System Integrity Monitor (SIM) within the sanity time-out interval. The recovery actions are dictated by the hardware just as they are for SAS class errors.

To determine whether the hardware initialization was an SAS class error or the result of a microinterrupt, two registers should be checked. These registers are the Microinterrupt Error Registers (UER0 and UER1). The microinterrupt handler uses these registers to pass information about the interrupt to the EIH process. If the microinterrupt handler cannot dispatch the EIH process and initiates an MRF, the data in these registers will be preserved for the postmortem dump.

In some situations, all the hardware registers listed in the postmortem dump must be examined to show where the fault lies. There is no set procedure for examining all the registers to find a clue about the error. For example, if the error involves an inability to fetch data from the memory *MY-STORE-A*, program dispatching (dispatching an error handling routine) cannot occur since memory access is impossible. Therefore, the only alternative is for the hardware circuits that detected the error to take action and start the reset of the system.

When the SOURCE field contains an **s**, the initialization is because of a program request. The REQUESTED INITIALIZATION PARAMETERS section shows additional information about the program request. The ON-LINE CU field shows the CU on which the system was able to recover successfully. The FAULTY CU field will show the CU where the initialization was started. These two fields may not be the same if the problem occurred on one CU and the decision was made to switch to the second CU. The FAULTY CU field will always show the CU on which the first initialization was attempted.

The *UNIX* RTR LEVEL and APPLICATION LEVEL fields show the level or amount of initialization that was performed to restore system sanity. The *UNIX* RTR operating system level 1 is merely an initialization of all system programs. *UNIX* RTR operating system levels 2, 3, and 4 all involve the retrieval of data from the disk. The amount of data retrieved increases with the *UNIX* RTR operating system level. The application levels are used for application program initialization and do not affect the severity of the *UNIX* RTR operating system initialization.

REQUESTED INITIALIZATION PARAMETERS

Under the REQUESTED INITIALIZATION PARAMETERS section, the following fields appear:

- SOURCE
- PANIC CODE
- *UNIX* RTR LEVEL
- APPLICATION LEVEL
- PHASE OPTIONS
- UTILITY ID.

These fields should be used only when the SOURCE field shows that the initialization is because of a program request (**s**). For hardware or manual requests, this section is not used. These fields show the initialization level, just as in the INITIALIZATION MESSAGE section, but here the levels are because of a program request.

The INITIALIZATION MESSAGE section shows what amount of initialization was taken before the system achieved sanity. However, the initialization requests from software are cumulative. Software may request a *UNIX* RTR operating system level 1 initialization, but the system recovery actions may already dictate a higher recovery level. In these cases, the SOURCE fields in the INITIALIZATION MESSAGE section and the REQUESTED INITIALIZATION PARAMETERS section will not agree.

The UTILITY ID and PANIC CODE fields commonly identify the process requesting the initialization and show the reason for the request. When software is the source of the initialization, the problem is usually located by these two fields. Frequently, a process will print additional information either through the normal spooling mechanism or through PRMs before it requests the initialization. Certain *UNIX* RTR operating system processes have their own log files where error data is saved when errors are detected. In any case, there may be additional information about the error on the printer and in log files that should be inspected to get the maximum benefit from the *UNIX* RTR operating system error reporting mechanisms.

BOOT DEVICES

This section provides the previous and current active boot devices.

For releases prior to *UNIX* RTR Operating System Release 21.27, a “0” indicates boot device 0, a “1” indicates boot device 1, and a “b” indicates both 0 and 1.

For *UNIX* RTR Operating System Release 21.27 and later, the MHD numbers of the previous and current active boot devices are shown. An asterisk indicates the mate of the boot device is not active.

EMERGENCY ACTION INTERFACE BUFFER

This section presents information gathered from the EAI. The EAI is an intelligent interface for communicating between the maintenance terminal and the computer and for inputting manual requests for configuration and initialization. Also, the EAI can operate without the computer being sane.

The EAI buffer is an area of low physical memory containing the status (configuration and latest requests) of the EAI. This buffer is provided in this section of the postmortem dump. This buffer is loaded only on manual initializations and is saved on later automatic initializations. The state of the computer on the last manual initialization is shown in the EAI buffer. The individual settings are single-bit flags to show the options that are in effect. Table 4-2 lists and describes the parameter fields. Each parameter field contains eight single-bit flags, except as noted. (Refer to the “Bit and Byte Numbering” section for Byte 3 numbering.) Typically, a 0 shows the option is not in effect and a 1 shows the option is set. Bits that are not included in the table are not used.

Table 4-2. EAI Parameter Buffer Values

Parameter Field	Description
MICROCODE	Options used to control the microcode initialization sequence. Bit 7: Boot from tape Bit 6: Manual bootstrap request Bit 5: Tape unit selection
CONFIGUR	Options used to control the boot device and partition selection. Bit 7: Back up root file system Bit 6: <i>UNIX</i> RTR operating system minimum configuration Bit 3: EAINEW — new EAI information for application (<i>UNIX</i> RTR Operating System Release 6.3 and later) Bit 2: Manually initiated initialization Bit 1: Boot device that is controlled by DFC 1 was active on the previous boot. Bit 0: Boot device that is controlled by DFC 0 was active on the previous boot.
BDID PAIR	Bus Device ID (BDID) of the boot device pair (<i>UNIX</i> RTR Operating System Release 21.27 and later). Leftmost nibble is the BDID of the boot device controlled by DFC 0, and the rightmost nibble is the BDID of the boot device controlled by DFC 1.
INIT LVL	Initialization level requested.
APPL PAR	Application parameter value passed to the application by the initialization message when initializations occur, including application-only initializations. Also, specific options for the Load Disk from Tape (LDFT) feature.
INHIBIT	Options used to control various low-level inhibits which disable certain hardware and software checks. Bit 7: Blocks hardware checks. Prevents the detection of SAS class errors from causing a system initialization. Bit 6: Inhibits error interrupts. Prevents the reporting of error interrupts by the EIH. Bit 5: Inhibits cache. Prevents the use of the cache. Bit 4: Blocks software checks. Prevents SIM from phasing the computer.

Table 4-2. EAI Parameter Buffer Values (Contd)

Parameter Field	Description
PERIODIC	Options scanned every 15 seconds and, if found set, are cleared and the appropriate action taken. Bit 7: Initialize the craft interface system Bit 6: Initiate an application-only initialization

TIMER

This section contains the sanity timer values (ON-LINE and OFF-LINE). The sanity timer is a hardware timing register that must be serviced on a regular basis by the software as a check for system sanity. If the timer is not serviced, there is hardware circuitry that will start recovery actions. The timer protects against situations where either a hardware error has occurred that prevents the system from executing or where there is a loss of software sanity.

The timer is of interest on hardware requested initializations. A value of 40 or greater in the rightmost two digits of the timer shows a sanity timer time-out.

The system takes the following actions on sanity timer time-outs:

Time-Out	CU	Forced	Action
First	ONL	yes	MRF
First	ONL	no	SAS
First	OFL	*	MRF
Second	ONL	yes	MRF
Second	ONL	no	MRF
Second	OFL	*	MRF

* Do not care

When bit 6 in the TIMER section equals 1, the first time-out is indicated; when bit 7 equals 1, the second time-out is indicated. There are two exceptions:

- If the block hardware checks are asserted (SSR bit 23 = 0), no action is taken for the first time-out on the on-line CU.
- When SSR bit 11 = 0, no action is taken.

GENERAL REGISTERS

This section provides the values of the general registers at the time of the MRF. These values are taken from the CU where the MRF occurred. This section is sometimes useful when the source of the initialization is hardware (microcode started the initialization). The following registers should also be examined for hardware requested initializations.

- General register 9 (argument pointer)
- General register A (frame pointer)
- General register B (stack pointer).

One thing to note is that a stack overflow often involves the kernel stack where memory management is enabled. Here, the overflow is detected because a kernel process tries to address a location not within its address space. There is no such protection (detection) when a boot process overflows its stack. Instead, a boot process stack overflow is usually more destructive since memory management is not enabled and, therefore, cannot prevent destruction of text and system data. The boot process stack overflow is detected because the boot process has destroyed its own text or the data in low physical memory.

The other general registers are used as follows:

- General register 0 — used to return values from function calls
- General registers 0, 1, 2 — used by the compiler as scratch registers
- General registers 3 - 8 — used as register variables with register 8 being the first
- General register 12 - 15 — used by the microcode as scratch registers.

FAULTY CU REGISTERS

This section provides information that may be useful in deciphering both software and hardware requested initializations. These registers show the hardware state of the CU when the MRF occurred and are essential in the resolution of MRFs because of hardware problems. Items like the SSR and Error Register (ER) can also be of interest on software requested initializations. The Instruction Buffer (IB) can also be used to verify that an initialization was truly software requested. Here, the IB would be DD00DD00.

To be of use, this section must be viewed in a larger context that includes the entire postmortem dump. The FAULTY CU title of this section does not necessarily imply that the CU hardware is at fault. It simply shows the CU that was processing when the MRF occurred.

INTERRUPT STACK SAVED STATE

This section is useful only for certain postmortem dumps that indicate software (s) or hardware (h) as the initialization source. Specifically, when some recovery action takes place, it is often desirable to know the identity of the interrupted process.

There are two ways for a process to have its execution suspended and saved state placed on the interrupt stack. The first has already been mentioned—a system interrupt. The second is when a process executes an Operating System Trap (OST). A process may only trap to a higher level process, therefore the calling process is immediately suspended in a way similar to an interrupt occurring.

The saved state data varies depending upon the source of the initialization. Determine the initialization source by examining the SOURCE field of the POSTMORTEM DUMP.

- Source **p** The data is not useful for initializations caused by a power clear.
- Source **h** The data is useful only for initializations caused by microinterrupts, such as a kernel stack overflow. The overflow prevents the EIH from executing (no stack left for the handler to run on). The saved state is for the process that was executing when the system initialization occurred.
- Source **s** The data is useful when software is responsible for the initialization. Prior to requesting an MRF the software may also print a PRM which may provide information about why the phase was needed. When a process requests an initialization, it executes an OST to request that the kernel reset the system via an MRF. As a result of the request, the requesting process gives up control and its saved state is placed on the interrupt stack. The kernel then executes a privileged assembler instruction that resets the computer. When such a sequence of events occurs, the postmortem dump shows the kernel as the currently running process. The UID will be X'7FF and this section will identify the process that called the kernel to request the initialization.
- Source **m** On manual initialization requests, the postmortem dump seldom contains any useful information. Here, whatever process was at the top of the interrupt stack when the MRF occurred will be listed in this section.

OFF-LINE REGISTERS

This section provides the PA, PSW, SSR, and ER from the off-line CU. The off-line CU is the CU that was off-line when the postmortem data was collected.

CACHE REGISTERS (3B21D Computer Only)

This section contains information about the source of the cache error and other related information and applies only to 3B21D computers.

⇒ NOTE:

The first software release will display nine cache registers, namely CRA, CRB, CRC, CRD, CRE, CRF, CRG, CTAG, and CSTAT. The CTAG and CSTAT registers do not give any relevant information. Therefore, later software releases will display only the seven registers. The values displayed are saved by the CC microcode during initialization. For backward compatibility, machines equipped with the first version of the CC microcode will display the value 0x0BADBAD0 for CRF and CRG. For subsequent versions, the values displayed will be the correct values.

MAIN STORE REGISTERS

3B20D Computer — This section provides the error registers from the memory controlling hardware (main store controller). They are only useful in dealing with MRFs because of memory problems (errors A, B, C, D, or a store time-out). They provide additional detail about the state of the memory hardware at the time of a memory error. The first row of these registers is for main store controller 0, and the second row is for main store controller 1.

3B21D Computer — This section provides information from the memory control registers. This information is useful to determine the state of the 3B21D computer memory hardware at the time of a memory error. This section has three subsections.

- The first subsection provides the memory control registers.
- The second subsection provides the data path registers.
- The third subsection provides the various bank boundaries (this determines the size of the memory equipped).

REAL-TIME CLOCK

This section is saved by the computer microcode and is the value of the Real-Time Clock (RTC) at the time of the MRF. It is sometimes useful in determining the time between several system recovery actions. (The RTC is in milliseconds.) The RTC is a 32-bit register and, thus, cannot be directly translated to yield the date.

Processor Recovery Messages

The PRMs are generated at each critical step of an initialization and can, therefore, be used to monitor the progress of an initialization. Unlike normal system messages that are routed to the spooling process and require a stable operating environment, PRMs can be output even when the computer is unable to achieve recovery. For this reason, they become invaluable in determining the causes of severe system problems.

Because of the severe circumstances under which PRMs become most valuable, the amount of data they provide is kept to a minimum. The purpose of the PRM is to provide indications of the problem in an expedient manner. The PRMs are described in greater detail in 254-303-103, *Processor Recovery Messages Guide*. To illustrate the information a PRM provides, consider the following:

PRM_1 E511 E31D 001A 000 68 50 00

By looking up the PRM in 254-303-103, *Processor Recovery Messages Guide*, the following information can be extracted:

- The process that printed the PRM is the EIH.
- The function and source line number in the EIH where the problem occurred.
- The error encountered was an inability to write data to the off-line CU's microstore.
- The current level of initialization that the system is undergoing (*UNIX* RTR operating system level 1).
- The identity of the CU where the PRM originated (CU 1).
- The active boot disk is a Moving Head Disk (MHD) 0.
- The CUs are not forced.
- The system is currently running in the maximum configuration on the primary root partition.

A PRM can provide much needed information in the analysis of system problems. Therefore, when a problem with the system develops, one should collect the entire ROP output from the time preceding the onset of the problem until the system has fully recovered.

Error Report Analysis

This section gives the procedure for analyzing the various types of error reports and the actions that should be taken to correct the fault. Use the following procedure to interpret a particular error interrupt log file entry:

Step	Procedure
1	Extract the supplemental data for the error from the log file by executing the OP:LOG command on the <i>MEMLOG</i> log file with the KW parameter. Pass the sequence number indicated in the REPT CU output message as the argument to the KW parameter. For detailed information regarding the use of this command, refer to the appropriate manual page in 254-303-110, <i>PDS Input Messages Manual</i>, or use the craft shell's help capability feature. The messages in <i>ERLOG</i> are not of a set format. Different errors require different information to be printed. Hardware errors produce messages where the contents of various hardware registers are provided. Software errors on the other hand provide the saved state of the process involved in the error. There are different approaches to using each type of message to determine the source of the problem. Refer to the "On-Line Hardware Faults," "Off-Line Hardware Faults," and "Software Faults" sections for specific details on these procedures. Steps 2 and 3 apply to all types of EIH reported errors including those in <i>MEMLOG</i>.
2	Classify the error as either a hardware or software error. To accomplish this, look at the REPT CU message. Refer to 254-303-111, <i>PDS Output Messages Manual</i>, or 254-303-113, <i>MML Output Messages Manual</i>, to interpret the message.

Step	Procedure
3	Refer to Table 4-3 for a brief description of the error. Included in the table is the fault and error log associated with the error. The fault type indicates the class of failure and is described in the "Error Interrupt Handler File" section. Also listed in Table 4-3 are general actions to take. Procedure references found in the table heading "Action to Be Taken" are indicated by a number within square brackets "[]" and have the following meanings: [1] See section "Procedure 1 – Memory History File" [2] See section "Procedure 2 – On-Line Hardware – DMA Detected" [3] See section "Procedure 3 – On-Line Hardware – DMA Detected" [4] See section "Procedure 4 – On-Line Hardware – DMA Detected" [5] See section "Procedure 5 – On-Line Hardware Fault" [6] See section "Procedure 6 – Off-Line Hardware Failures" [7] See section "Procedure 7 – Software Fault" If you have trouble locating a procedure reference, scan the Table of Contents for Chapter 4 under the heading "Error Report Analysis."

In many cases, the resolution of a problem requires analysis of several log files, PRMs, and other system output. The error entry by itself may not be enough. To effectively analyze a problem, the system output should be collected for a period of time prior to and after the system has recovered. The intent is to preserve system error message sequence and continuity. This is sometimes vital to the resolution of a problem.

Additional information is found in the text sections immediately following the procedures.

Table 4-3. Error Interrupt Handler Faults

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'2†	EAI reset type error threshold exceeded	On-line hardware	An excessive number of errors has occurred on the EAI hardware on the CC board. A reset of the EAI hardware will be attempted.	Switch to the other processor and diagnose the CC.
X'3†	EAI switch type error threshold exceeded	On-line hardware	An excessive number of errors has occurred on the EAI hardware on the CC board. A previous reset of the EAI hardware was not sufficient to clear the error. A soft switch will be attempted.	If the attempted soft switch failed, then restore and switch to the other processor. Diagnose the faulty CC.
X'8	A my store error C [MEMLOG]	On-line hardware	A noncorrectable parity error occurred on the currently running CU.	If the processor switched as a result of the error, run diagnostics on the processor encountering the fault. [1]
X'B	Invalid maintenance channel order [ERLOG]	Off-line hardware	The maintenance channel is faulty or maintenance channel protocol has been violated.	If the error occurs repeatedly, run diagnostics on the off-line processor. If no problem is found, run diagnostics on the on-line processor. [6]
X'C	Restore hardware error [MEMLOG]	Off-line hardware	The memory controller of the off-line processor encountered a store error A condition.	Indicates a problem with the controller hardware, not the memory. The implicated controller should be diagnosed and replaced if found to be faulty. [1]
X'D	Other store correctable memory failure [MEMLOG]	Off-line hardware	A correctable parity error (other store D) occurred in the mate processor. This type of error can occur only if the mate processor is in the standby state.	Run diagnostics if the mate processor is removed from a standby state. [1]

* The X' represents the REPT CU [1, 0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

† For the 3B21D computer only.

Table 4-3. Error Interrupt Handler Faults (Contd)

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'E	Other store noncorrectable memory failure [MEMLOG]	Off-line hardware	A noncorrectable memory error (other store C) occurred in the mate processor. This type of error can occur only if the mate processor is in the standby state.	Run diagnostics if the mate processor is removed from a standby state. [1]
X'F	Other store time-out error [ERLOG]	Off-line hardware	The memory controller of the mate processor did not respond to an access.	Check the update cables on both processors and run diagnostics on the off-line processor. [6]
X'10	Input/Output (IO) channel error [ERLOG]	On-line hardware	The faulty unit encountered a fatal error condition. The channel status and channel address are provided.	Examine the channel status register layout in the "Register Layouts" section to determine the error type and the appropriate action to be taken. [5]
X'11	3/6 main channel IO response error [ERLOG]	On-line hardware	More than one channel responded to an IO address.	Run diagnostics on the processor encountering the fault. [5]
X'12	3/6 main channel sequencing error [ERLOG]	On-line hardware	More than one IO pulse point was active in the command to the channel.	Run diagnostics on the processor encountering the fault. [5]
X'13	Parity divert error [ERLOG]	On-line hardware	A named unit returned a word with bad parity when the instruction expected good parity.	Run diagnostics. [5]
X'14	On-line correctable memory failure [MEMLOG]	On-line hardware	A correctable memory failure occurred in the on-line processor (my store D).	If the processor encountering the fault is removed, run diagnostics on it. [1]
X'15	Protection violation [ERLOG]	Software	A program tried to execute from a segment that did not have execute permission or tried to write into a segment that did not have write permission.	Forward the software log entry information to your technical assistance organization. [7]

* The X' represents the REPT CU [1,0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

Table 4-3. Error Interrupt Handler Faults (Contd)

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'16	Virtual Address Out of Range (VORA) [ERLOG]	Software	A program attempted to access memory with virtual address values exceeding 64 megabytes.	Forward the software log entry information to your technical assistance organization. [7]
X'17	Out-of-range memory reference [ERLOG]	Software	A program tried to access unequipped memory.	Forward the software log entry information to your technical assistance organization. [7]
X'18	Out-of-range memory reference (other store) [ERLOG]	Software	The off-line main memory considered the address to be out of range while the on-line memory did not. This usually indicates a problem with the off-line main memory.	Check and replace the fuses in the off-line memory (if necessary). Run diagnostics on off-line memory. [7]
X'19	Privileged instruction violation [ERLOG]	Software	The program tried to execute one of the special instructions without a corresponding privilege bit set in the PSW.	Forward the software log entry information to your technical assistance organization. [7]
X'1A	Bad alignment on memory reference [ERLOG]	Software	The program executed an instruction and used the wrong type of address (for example, the execution of a full-word instruction with a half-word address).	Forward the software log entry information to your technical assistance organization. [7]
X'1B	Illegal operand [ERLOG]	Software	One of the operands specified in an instruction is invalid for the particular opcode.	Forward the software log entry information to your technical assistance organization. [7]
X'1C	Illegal instruction [ERLOG]	Software	The data in the instruction buffer did not correspond to a valid instruction.	Forward the software log entry information to your technical assistance organization. [7]

* The X' represents the REPT CU [1, 0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

Table 4-3. Error Interrupt Handler Faults (Contd)

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'1D	Illegal stack switch [ERLOG]	Software	The program attempted to switch to a private stack or attempted to switch to the kernel stack when it was already running on the kernel stack.	Forward the software log entry information to your technical assistance organization. [7]
X'1E	Spurious interrupt [ERLOG]	Software	The EIH was entered and did not find an error set.	If possible, the processors should be switched so that the faulty CU can be diagnosed. In any case, forward the results of the diagnostic and the ERROR log to the responsible engineer through the technical assistance organization.
X'20	My store A during Direct Memory Access (DMA) [ERLOG]	On-line hardware	The Direct Memory Access Controller (DMAC) encountered a fatal main memory problem during a DMA transfer.	Run diagnostics on the unit indicated in the <i>MEMLOG</i> entry. [2]
X'21	My store B during DMA [MEMLOG]	On-line hardware	The DMAC used an out-of-range address.	Run diagnostics on the unit indicated in the <i>MEMLOG</i> entry. [1]
X'22	DMA detected noncorrectable parity failure [MEMLOG]	On-line hardware	Noncorrectable memory error (my store C) occurred in the on-line processor during DMA.	If the processors switched as a result of the error, run diagnostics on the processor encountering the fault. [1]
X'25	Bad parity on the Direct Memory Access Input/Output (DIO) bus [ERLOG]	On-line hardware	The DMAC received bad parity from the named CHANNEL ADDRESS device on data destined for the main memory during a DMA transfer.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]

* The X' represents the REPT CU [1,0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

Table 4-3. Error Interrupt Handler Faults (Contd)

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'26	Bad parity on DMA read [ERLOG]	On-line hardware	The DMAC detected bad parity on its internal bus. This can be caused by a DMAC failure or a request from a device that has not had its DMAC Random Access Memory (RAM) block initialized.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]
X'27	DMA RAM parity failure [ERLOG]	On-line hardware	The DMAC detected bad parity over the RAM address register. This occurs only on the Lucent Technologies 3B20D Model 1 computer.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]
X'29	DIO ready failure [ERLOG]	On-line hardware	The named channel did not return a READY signal to the DMAC.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [4]
X'2A	DIO-ASW failure [ERLOG]	On-line hardware	The named channel did not return all-seems-well to the DMAC.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [4]
X'2B	DIO acknowledge failure [ERLOG]	On-line hardware	Either more than one or no channel responded to the DMAC.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [4]
X'2C	Central Control Input/Output (CCIO) data parity failure [ERLOG]	On-line hardware	The command received by the DMAC on a Programmed Input/Output (PIO) operation had bad parity.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]
X'2D	CCIO-command error [ERLOG]	On-line hardware	A second programmed IO order was sent to the DMAC before the first one was completed.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]

* The X' represents the REPT CU [1,0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

Table 4-3. Error Interrupt Handler Faults (Contd)

Interrupt Type No.*	Fault [Log File]	Fault Type	Description	Action to Be Taken [Procedure No.]
X'2E	DMA-ROM parity failure [ERLOG]	On-line hardware	Bad parity was detected in the DMAC read only memory. This can be caused by other fatal errors occurring in the DMAC.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [3]
X'2F	DMA-address error [ERLOG]	On-line hardware	During the virtual to physical address translation, the DMAC found that the requested IO access was not permitted to physical memory, or during the data transfer, the data would have been written beyond the end of a segment.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [4] Analyze the <i>ERLOG</i> output and the implicated driver.
X'30	DMA-channel request error [ERLOG]	On-line hardware	A device on a DMA channel requested service but the DMAC RAM block for the device did not have its valid bit set, or, a channel on the DMAC requested service but no device on that channel had a request pending.	Run diagnostics on the unit indicated in the <i>ERLOG</i> entry. [4]
X'32	Spurious interrupt [ERLOG]	On-Line hardware	EIH was dispatched to handle an error but no error bit in the error registers was set.	Run diagnostics on the CU in which the error occurred and forward the <i>ERRLOG</i> and ROP output to the responsible engineer.
X'33	Bad operating system [ERLOG]	Software	A bad operating system trap was detected by the kernel.	Forward the software log entry information to your technical assistance organization. [7]

* The X' represents the REPT CU [1,0] ERROR INTERRUPT portion of the error message. The interrupt type number is in hexadecimal and appears in the REPT CU output message.

Procedure 1 – Memory History File

The memory history file is described in the “Control Unit Error Messages” section. Access and interpret the *MEMLOG* file entry using the following procedure:

Step	Procedure
1	Run Main Store Controller (MASC) diagnostics including demand phases. If a failure occurs, replace the faulty board indicated in the <i>ANALY:TLPFILE</i> output message.  NOTE: Steps 2 and 3 are for the 3B20D computer only.
2	Enter: In PDS, OP:MEMERRS:DATE a[-b]! In MML, OP:MEMERRS:DATA,DATE=a[&&b]; Where <i>a</i> is starting date and <i>b</i> is ending date. <i>a</i> and <i>b</i> are of the form yy mmyy mmddyy. Analyze the memory error data for each MASC (0 and 1) on each CU (0 and 1) to determine if there is a pattern that implicates boards in a particular CU and MASC. Replace the boards that can be clearly identified as faulty.
3	Move the boards that are suspected faulty to another location. Note the time of movement and new locations of the suspected faulty boards. Track the movement of memory errors to confirm faulty boards. If Steps 2 and 3 are inconclusive, continue to next step.
4	Refer to the “Interpreting Memory Failures” section for additional information. Also refer to the REPT CU output message in 254-303-111, <i>PDS Output Messages Manual</i>, or 254-303-113, <i>MML Output Messages Manual</i>.

Step	Procedure
5	Extract the supplemental data for the error from the <i>MEMLOG</i> log file by executing the OP:LOG command on the <i>MEMLOG</i> log file with the KW parameter. Pass the sequence number indicated in the REPT CU output message as the argument to the KW parameter as follows: In PDS, OP:LOG;LG "MEMLOG":KW "a"! In MML, OP:LOG:MEMLOG:DATA,KW=a; Where <i>a</i> is the sequence number. For detailed information regarding the use of this command, refer to the appropriate manual page in 254-303-110, <i>PDS Input Messages Manual</i>, or 254-303-112, <i>MML Input Messages Manual</i>, or use the craft shell's help capability feature.
6	Check the first or second line of the <i>MEMLOG</i> log file entry to determine the type of memory problem. The error message will be one of the following: (a) NONCORRECTABLE PARITY ERROR ON MEMORY REFERENCE <i>a</i> (X'8) (b) MEMORY PARITY FAILURE <i>a</i> (X'14) (c) OTHER STORE REFRESH PARITY ERROR <i>a</i> (X'D) (d) OTHER STORE READ PARITY ERROR <i>a</i> (X'E) (e) OTHER STORE HARDWARE ERROR <i>a</i> (X'C) (f) MY STORE B DURING DMA <i>a</i> (x'21) (g) NONCORRECTABLE PARITY ERROR ON DMA <i>a</i> (X'22). The <i>a</i> in each of the error messages is the sequence number indicated in the REPT CU output message. ⇒ NOTE: The numbers in parentheses at the end of each of the messages represent the error interrupt type numbers (in hexadecimal) from the REPT CU message. These do not print out in the log file entry. They appear only in the REPT CU message and are shown here for reference purposes only.

Step	Procedure
7	Interpret the error message(s) as follows: A NONCORRECTABLE PARITY ERROR indicates that a location in memory has encountered a double-bit error. This error type is the most likely to damage running programs. Error messages (a), (b), (c), and (d) of Step 6 indicate a memory array if there is correlation between the errors; for example, the same failing location. If errors (a) through (d) occur in several memory arrays or if the error is (e), the MASC is the likely cause. Isolated, uncorrelated events usually do not require action, but records should be kept of the occurrences.
8	Examine the register information. Many error reports have physical addresses supplied. If supplied, refer directly to Tables 4-4 through 4-8 instead of using this procedure. ▪ 3B20D Computer — Although the Trapped Address Register (TAR) and the MASC Error Registers (ERs) are printed for both controllers, examine only the information for the faulty unit. To determine the unit on which the error occurred, check the FAULTY UNIT designation in the <i>MEMLOG</i> file entry. If neither controller is reported faulty, examine all information for both units. Most 3B20D computers have only one memory controller; memory controller 0. ▪ 3B21D Computer — Replace the memory board (KLW32, KLW40, KLW48, KLW64, KLW128 or KLW256).
	⇒ NOTE: Steps 9 through 15 are for the 3B20D computer only.
9	Check the second digit from the left of the MASC 0 TAR for validity (must be 8, 9, A, B, C, D, E, or F). If the MASC 0 Trapped Address (TA) is invalid (invalid if = FFFFFFFF), repeat the check for MASC 1. ⇒ NOTE: For systems with the Extended Main Memory (EMM) feature, the procedure in Steps 10, 11, 13, 14, and 15 should be followed. For systems with the Very Large Main Memory (VLMM) feature, the procedure in Steps 10, 12, 13, 14, and 15 should be followed. If the system does not have the EMM or VLMM feature, the procedure in Steps 10, 13, 14, and 15 should be followed.

Step	Procedure

- 10 Convert the TAR to get the faulty address and identify the faulty memory array slot. A conversion is necessary because the address bits in this register do not yield a contiguous address. The bit layout is as follows:

0100 0000 0111 1111 1111 1111 1111 1100
↑

The bit layout shows 1 bit (second leftmost bit) separated by zeros from the lower 23 bits. The conversion procedure is performed to shift the second leftmost bit seven positions to the right so that the 24 bits of the address are contiguous.

The following procedure may be used to convert the TAR:

- (a) If the TN28 or TN56 is equipped, check the leftmost digit in the register.

If digit is 0, 1, 2, or 3, then the base address is 0000000.

If digit is 4, 5, 6, or 7, then the base address is 0800000.

If digit is 8, 9, A, or B, then the base address is 1000000.

If digit is C, D, E, or F, then the base address is 1800000.

If the TN14 is equipped and the failure is in MASC 1, the base address is also 800000.

In all other cases, the base address is 000000.

- (b) Replace (or drop) the leftmost two digits of the TAR with zeros.
- (c) Look at the third leftmost digit of the TAR. If this digit is seven or less, then it should not be changed. Otherwise, it should be converted as follows:

Original	New Value
8	0
9	1
A	2
B	3
C	4
D	5
E	6
F	7

The reason for this conversion is that the third leftmost digit has only 3 bits of value. The most significant bit of this nibble is not part of the address, and therefore, it is discarded.

- (d) Refer to Tables 4-4 through 4-7, depending on the type of memory array (TN14, TN28, TN56, or TN2012), to locate the faulty memory array slot. Refer to Example 1.

Step	Procedure

Example 1

TAR	Range	Faulty Address
183FFFCC	LOW	3FFFCC
583FFFCC	HIGH	3FFFCC
19C14568	LOW	414568
D9C14568	HIGH	414568

- **TN14 Memory Array:** Refer to Table 4-4 to locate the faulty memory array slot for TN14. A fully equipped main memory using TN14 will require the use of two memory controllers (0 and 1). The LOW and HIGH address ranges mean MASC 0 (controller 0) and MASC 1 (controller 1), respectively. Each controller in this case will support only address ranges of 000000 through 7FFFFC. The combined capacity of both controllers is 16 MB (8 MB each). Some examples for TN14 memory arrays are as follows:

Range	Faulty Address	Slot	Controller	Table Used
LOW	3FFFCC	7	0	4-4
HIGH	3FFFCC	7	1	4-4
LOW	414568	8	0	4-4
HIGH	414568	8	1	4-4

- **TN28 and TN56 Memory Arrays:** There is only one memory controller capable of addressing the full 16-MB address range for TN28 and TN56 arrays. In this case, the HIGH/LOW address range designation no longer means MASC 1 and MASC 0. It should be incorporated into a full 24-bit address. This is done by adding 800000 to the converted address if the address range was found to be HIGH. After this addition, use Tables 4-5 and 4-6 to locate the faulty memory array slot for TN28 and TN56 memory arrays, respectively. Some examples for TN28 memory arrays are as follows:

Range	Faulty Address	Slot	Table Used
LOW	3FFFCC	3	4-5
HIGH	BFFFCC	11	4-5
LOW	414568	4	4-5
HIGH	C14568	12	4-5

Step	Procedure

Some examples for TN56 memory arrays are as follows:

Range	Faulty Address	Slot	Table Used
LOW	3FFFCC	1	4-6
LOW	414568	2	4-6
HIGH	C14568	6	4-6

- **TN2012 Memory Arrays:** MASC 1 (controller 1) is not associated with these arrays. The trapped address with these arrays can range from 0x0 to 0x1FFFFFFC for a 32-MB system in *UNIX* RTR Operating System Release 1 and from 0x0 to 0x3FFFFFFC for a 64-MB system in *UNIX* RTR Operating System Release 6. Any system equipped with more than 32 MB of memory must have TN2012 arrays (4 MB/array) installed. If the system is running on *UNIX* RTR Operating System Release 6, continue to Step 11.

- 11 **EMM Only** — Convert the TAR to get the faulty address and identify the faulty memory array slot.

For *UNIX* RTR Operating System Release 1 only — for systems equipped with more than 16 MB of memory, the conversion procedure is different than the procedure for systems equipped with 16 MB or less of memory. The reason for the difference is the additional address bit required for an address range of 16 through 32 MB.

The address bits of the TAR are not contiguous. The bit layout is as follows:

```
1100 0000 0111 1111 1111 1111 1111 1100
↑↑
```

The bit layout shows 2 bits (leftmost 2 bits) separated by zeros from the lower 23 bits. The conversion procedure is performed to shift the leftmost 2 bits seven positions to the right so that the 25 bits of the address are contiguous.

Step	Procedure

The following procedure should be used to translate the TAR:

- (a) Observe the third leftmost digit of the TAR. If the value of this digit is 0, 1, 2, 3, 4, 5, 6, or 7, continue to Step 11(b). Otherwise, convert this digit as follows:

Present Value	New Value
8	0
9	1
A	2
B	3
C	4
D	5
E	6
F	7

The purpose of this step is to eliminate the high-order bit of the third leftmost nibble, since this bit is not part of the address.

- (b) Observe the leftmost digit of the TAR. The two most significant bits of this nibble are the two most significant bits of the address. These bits must be extracted and concatenated with the low three significant bits of the third leftmost significant digit of the TAR [converted in Step 11(a)]. The following list is used to extract the 3 bits of address from the leftmost nibble of the TAR:

Leftmost Nibble	Converted Value
0	0
1	0
2	0
3	0
4	8
5	8
6	8
7	8
8	10
9	10
A	10
B	10
C	18
D	18
E	18
F	18

Step	Procedure
(c)	The next step of the address conversion involves the concatenation of the results derived in Steps 11(a) and 11(b). This can be done by a simple hexadecimal addition of the results of the two previous steps. As a check, ensure that the result of Step 11(a) is a value between 0 and 7 and the result of Step 11(b) is 0, 8, 10, or 18. After the addition, the resulting value has a range of 0 through 1F hexadecimal.
(d)	Incorporate the result of the addition into the TA. To do this, the three most significant nibbles of the TAR are set to zero and the addition result becomes the most significant 5 bits of the TA. The resulting TA can now be used to determine the memory array slot that contains the failing location. For systems equipped with TN56 boards and more than 16 MB of memory, refer to Table 4-6 to determine the slot number. For systems equipped with the TN2012 boards, refer to Table 4-7 to determine the slot number. Examples 2 and 3 illustrate some address conversions.

Example 2

A system equipped with more than 16 MB of main memory cannot use memory arrays smaller than the TN56 (2 MB each). There is only one memory controller required for the entire maximum memory size of 32 MB. For this reason, the only memory slot look-up table that is used is Table 4-6. The last addressable word of such a configuration is 0X1FFFFFFC.

Faulty Address	Converted Address	Slot (Decimal)	Table Used
3F312345	0312344	1	4-6
4F312345	0B12344	5	4-6
4F812345	0812344	4	4-6
8F312345	1312344	9	4-6
9F312345	1312344	9	4-6
DF312345	1B12344	13	4-6
FF012345	1812344	12	4-6

Step	Procedure

Example 3

This is the same example as Example 2 except that TN2012 arrays (4 MB each) are assumed. For the TN2012 arrays, refer to Table 4-7 to determine the slot number of the faulty array. The last addressable word of such a configuration is 0X1FFFFFC.

Faulty Address	Converted Address	Slot (Decimal)	Table Used
3F312345	0312344	0	4-7
4F312345	0B12344	2	4-7
4F812345	0812344	2	4-7
8F312345	1312344	4	4-7
9F312345	1312344	4	4-7
DF312345	1B12344	6	4-7
FF012345	1812344	6	4-7

- 12 **VLMM Only** — Convert the TAR and the Memory Error Register 2 (MER2) to a TA as outlined below, and then, refer to Table 4-7, to determine the faulty memory array slot. For *UNIX* RTR Operating System Release 6, both registers must be used to determine the TA.

The TA bits 2 through 22 are in the TAR bits 2 through 22.

The TA bit 23 is located in the TAR bit 30.

The TA bit 24 is located in the TAR bit 31.

The TA bit 25 is located in the MER2 bit 23.

Concatenating these bits in the proper order yields the TA which must be used with Table 4-7 to determine the faulty array slot number. The last 2 bits of the TA are always zero since the TA is a *word* address.

The bit layout of the TA bits in the registers is given as follows, where the address bits are indicated by an *x* and the nonaddress bits are indicated by zeros.

Bit#:	3	2	2	2	1	1			
TAR:	xx00	0000	0xxx	xxxx	xxxx	xxxx	xxxx	xxxx	xx00
MER2:	0000	0000	x000	0000	0000	0000	0000	0000	0000

Step	Procedure

The TA must be formed (converted) by concatenating the noncontiguous address bits as follows:

	3	2		2		2		1		1							
Bit#:	1	8		4		0		6		2		8		4		0	
TA:	0000	00xx		xxxx		xxxx		xxxx		xxxx		xxxx		xxxx		xx00	

The TA conversion is completed when the TAR bits 30 and 31 and the MER2 bit 23 are correctly concatenated with the TAR bits 2 through 22 as indicated. Refer to Table 4-7 to determine the faulty slot number for the TN2012 array. Some examples of the TA conversion are given in Example 4.

Example 4

Slot numbers are determined using Table 4-7. Raw TAR and the MER2 values are also given.

TAR Value	MER2 Value	Converted Address	Slot (Decimal)	Table Used
3F312345	1289abcd	2312344	8	4-7
3F312345	1249abcd	0312344	0	4-7
4F312345	1200abcd	0B12344	10	4-7
4F312345	1259abcd	0B12344	2	4-7
DF512345	12A9abcd	3D12344	15	4-7
DF512345	1279abcd	1D12344	7	4-7
9F812345	12B9abcd	3012344	12	4-7
9FF12345	12C9abcd	3712344	13	4-7

- 13 Steps 1 through 12 locate the memory array shown in the *MEMLOG* file. A single occurrence of a fault does not imply that a particular memory array is faulty. Diagnostics should be run on the main memory to verify that a hard failure (always fails) exists before a memory array is replaced. If a hard failure exists, the memory array should be replaced. Even though an error report message indicates a fault, if memory diagnostics do not reveal a problem, the memory array should not be immediately replaced. A better approach is to swap the array with another under the same controller (noting the array positions). If the same array fails in its new position, it should then be replaced. If the failures continue in the same position as before the swap, the problem is most likely with the controller. In this case, consider swapping or replacing the suspect controller.

Step	Procedure
14	Run MASC diagnostics including demand phases. ▪ If a failure occurs, replace faulty pack indicated in Step 13. ▪ If no failures occur, rotate pack identified in Step 13 to another location and track the movement, confirming faulty board. ▪ If the error flag is a 23 and no faults were detected during diagnostics, return the unit to service.
15	Refer to the “Interpreting Memory Failures” section for additional information.

Table 4-4 relates to the 3B20D Model 1 computer with two main store controllers.

Table 4-4. Address Conversion Table (TN14 Memory Array)

Converted Address	Range	Faulty Slot	Controller
000000 – 07FFFC	LOW	0	0
080000 – 0FFFFC	LOW	1	0
100000 – 17FFFC	LOW	2	0
180000 – 1FFFFC	LOW	3	0
200000 – 27FFFC	LOW	4	0
280000 – 2FFFFC	LOW	5	0
300000 – 37FFFC	LOW	6	0
380000 – 3FFFFC	LOW	7	0
400000 – 47FFFC	LOW	8	0
480000 – 4FFFFC	LOW	9	0
500000 – 57FFFC	LOW	10	0
580000 – 5FFFFC	LOW	11	0
600000 – 67FFFC	LOW	12	0
680000 – 6FFFFC	LOW	13	0
700000 – 77FFFC	LOW	14	0
780000 – 7FFFFC	LOW	15	0
000000 – 07FFFC	HIGH	0	1
080000 – 0FFFFC	HIGH	1	1
100000 – 17FFFC	HIGH	2	1
180000 – 1FFFFC	HIGH	3	1
200000 – 27FFFC	HIGH	4	1
280000 – 2FFFFC	HIGH	5	1
300000 – 37FFFC	HIGH	6	1
380000 – 3FFFFC	HIGH	7	1
400000 – 47FFFC	HIGH	8	1
480000 – 4FFFFC	HIGH	9	1
500000 – 57FFFC	HIGH	10	1
580000 – 5FFFFC	HIGH	11	1
600000 – 67FFFC	HIGH	12	1
680000 – 6FFFFC	HIGH	13	1
700000 – 77FFFC	HIGH	14	1
780000 – 7FFFFC	HIGH	15	1

Table 4-5. Address Conversion Table (TN28 Memory Array)

Converted Address	Faulty Slot
000000 – 0FFFFC	0
100000 – 1FFFFC	1
200000 – 2FFFFC	2
300000 – 3FFFFC	3
400000 – 4FFFFC	4
500000 – 5FFFFC	5
600000 – 6FFFFC	6
700000 – 7FFFFC	7
800000 – 8FFFFC	8
900000 – 9FFFFC	9
A00000 – AFFFFC	10
B00000 – BFFFFC	11
C00000 – CFFFFC	12
D00000 – DFFFFC	13
E00000 – EFFFFC	14
F00000 – FFFFFC	15

Table 4-6. Address Conversion Table (TN56 Memory Array)

Converted Address	Faulty Slot
000000 – 1FFFFC	0
200000 – 3FFFFC	1
400000 – 5FFFFC	2
600000 – 7FFFFC	3
800000 – 9FFFFC	4
A00000 – BFFFFC	5
C00000 – DFFFFC	6
E00000 – FFFFFC	7
1000000 – 11FFFFC	8
1200000 – 13FFFFC	9
1400000 – 15FFFFC	10
1600000 – 17FFFFC	11
1800000 – 19FFFFC	12
1A00000 – 1BFFFFC	13
1C00000 – 1DFFFFC	14
1E00000 – 1FFFFFC	15

Table 4-7. Address Conversion Table (TN2012 Memory Array)

Converted Address	Faulty Slot
000000 – 3FFFFC	0
400000 – 7FFFFC	1
800000 – BFFFFC	2
C00000 – FFFFFC	3
1000000 – 13FFFFC	4
1400000 – 17FFFFC	5
1800000 – 1BFFFFC	6
1C00000 – 1FFFFFC	7
2000000 – 23FFFFC	8
2400000 – 27FFFFC	9
2800000 – 2BFFFFC	10
2C00000 – 2FFFFFC	11
3000000 – 33FFFFC	12
3400000 – 37FFFFC	13
3800000 – 3BFFFFC	14
3C00000 – 3FFFFFC	15

Table 4-8. Address Conversion Table (3B21D Computer Memory Pack)

Bank Number	SIMM Number
0	1
1	1
2	2
3	2
4	3
5	3
6	4
7	4

Procedure 2 – On-Line Hardware – DMA Detected

This procedure is valid only for the following output message:

REPT CU [1,0] ERROR INTERRUPT [20] [a]

Where *a* is the sequence number.

The following is an example of an output message:

```
12 REPT ERSLOT
MY STORE A DURING DMA X'124
ONLINE CU: 0
FAULTY UNIT: DFC 0

DMA ADDRESS: 23 DMA STATUS REGISTER: 00F07D7E
DMA CHANNEL STATUS REGISTER: 00008A47
CHANNEL ADDRESS: 25 CHANNEL STATUS REGISTER:00000000
DMA RAM: 0008FF00 00A6FFF8 0008FA0C 8602BC00
DMA STORE ADDRESS REGISTER: 0002E420
DMA STORE DATA REGISTER: FB02B800
DMA INCREMENTOR: 0008FA10 DMA DIO DATA REGISTER: 000008E4
DMA BUFFER REGISTER: 20000600 DMA MASK REGISTER: 00000000
REAL TIME CLOCK: 3B03B33F
```

Step	Procedure
1	Extract the <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where <i>a</i> is the sequence number.
2	If <i>FAULTY UNIT</i> indicates a main store problem (for example, <i>CU 0 MASC 0</i>), a message may also have been written to <i>MEMLOG</i> . Extract the <i>MEMLOG</i> file entry using the same sequence number and follow the instructions outlined in Procedure 1.
3	Run diagnostics on the unit indicated in the <i>FAULTY UNIT</i> of <i>ERLOG</i> or <i>MEMLOG</i> . If the problem cannot be corrected with standard diagnostics phases, request the appropriate demand phases.

Procedure 3 – On-Line Hardware – DMA Detected

This procedure is valid for the following output message:

REPT CU [1,0] ERROR INTERRUPT [21,25,26,27,28,2C,2D,2E] [a]

Where a is the sequence number.

The following is an example of an output message:

```
12 REPT ERSLOT
DMA RAM PARITY FAILURE X'124
ONLINE CU: 0
FAULTY UNIT: DFC 0

DMA ADDRESS: 23 DMA STATUS REGISTER: 00F06D7F
DMA RAM: 00000000 80000000 0015AD00 007AFF3A
DMA STORE ADDRESS REGISTER: 0002E420
DMA STORE DATA REGISTER: FB02B800
DMA INCREMENTOR: 0008FA10 DMA DIO DATA REGISTER: 000008E4
DMA BUFFER REGISTER: 20000600 DMA MASK REGISTER: 00000000
REAL TIME CLOCK: 3B03B33F
```

Step	Procedure
1	Extract the <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where a is the sequence number.
2	Request diagnostics on the unit indicated as the <i>FAULTY UNIT</i> .

Procedure 4 – On-Line Hardware – DMA Detected

This procedure is valid only for the following output message:

REPT CU [1, 0] ERROR INTERRUPT [29, 2A, 2B, 2F, 30] [a]

Where a is the sequence number.

The following is an example of an output message:

```
12 REPT ERSLOT
BAD PARITY ON DMA READ X'124
ONLINE CU: 0
FAULTY UNIT: DFC 0

DMA ADDRESS: 23 DMA STATUS REGISTER: 00F06D7F
DMA CHANNEL STATUS REGISTER: 00008A47
CHANNEL ADDRESS: 25 CHANNEL STATUS REGISTER: 00000000
DMA RAM: 0008FF00 00A6FFF8 0008FA0C 8602BC00
DMA STORE ADDRESS REGISTER: 0002E420
DMA STORE DATA REGISTER: FB02B800
DMA INCREMENTOR: 0008FA10 DMA DIO DATA REGISTER: 000008E4
DMA BUFFER REGISTER: 20000600 DMA MASK REGISTER: 00000000
REAL TIME CLOCK: 3B03B33F
```

Step	Procedure
1	Extract <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where a is the sequence number.
2	If the message line indicates a channel problem (that is, DIO), check the <i>CHANNEL ADDRESS</i> in the <i>ERLOG</i> file.
3	Run diagnostics on the unit indicated in the <i>FAULTY UNIT</i> of <i>ERLOG</i> or <i>MEMLOG</i> . If the problem cannot be corrected with standard diagnostic phases, request the appropriate demand phases. If channel problems are indicated, run direct memory access channel demand phases.

Procedure 5 – On-Line Hardware Fault

This procedure is valid only for the following output message:

```
REPT CU [1, 0] ERROR INTERRUPT [10, 11, 12, 13] [a]
```

Where a is the sequence number.

The following is an example of an output message:

```
23 REPT ERSLOT  
IO CHANNEL ERROR    X'34  
ONLINE CU: 0  
FAULTY UNIT: DMA 0  
HSR: 00005A00
```

Step	Procedure
1	Extract <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where a is the sequence number.
2	Determine the indicated faulty channel by using Table 4-9 and the third and fourth digits of the Hardware Status Register (HSR).
3	Determine which channels are equipped in your computer.
4	If the address or HSR corresponds to a channel in the table but the channel is not equipped, the Equipment Configuration Database (ECD) may contain an incorrect entry. Use Recent Change and Verify (RC/V) to check the database for channel equipage. If the channel is equipped but not specified in the ECD, the <i>UNIX</i> RTR operating system will ignore that channel.  NOTE: If the channel is specified in the ECD but subsequently removed from the CCIO bus, the computer will not boot.

Step	Procedure
5	If the HSR converts to a valid channel (equipped), check the cabling for proper connection and request diagnostics on that channel (specify Phase 40 and a helper unit — DFC 0 or 1 or IOP 0 or 1).
6	If an error occurs on one processor and not the other, request diagnostics on the processor that indicated a fault.

Table 4-9. HSR/Channel Conversion Table

Fourth Digit	Third Digit				Faulty Channel
1	C	D	E	F	0
2	C	D	E	F	1
3	4	5	6	7	2
3	8	9	A	B	3
4	C	D	E	F	4
5	4	5	6	7	5
5	8	9	A	B	6
8	C	D	E	F	10
9	4	5	6	7	11
9	8	9	A	B	12
A	8	9	A	B	13
A	4	5	6	7	14
B	0	1	2	3	15
C	4	5	6	7	16
C	9	9	A	B	17
D	0	1	2	3	18
E	0	1	2	3	19

Procedure 6 – Off-Line Hardware Failures

This procedure is valid only for the following output message:

REPT CU [1, 0] ERROR INTERRUPT [0B, 0F] [a]

Where a is the sequence number.

The following is an example of an output message:

```
15 REPT OTHER STORE HARDWARE ERROR X'19
ONLINE CU: 1
FAULTY UNIT: CU 0 MASC 0
MAIN STORE CONTROLLER 0
TRAPPED ADDRESS REGISTER: 18000001
ERROR REGISTER 1: 00000000
ERROR REGISTER 2: 400b4041
MAIN STORE CONTROLLER 1 (MAY NOT BE CONFIGURED)
TRAPPED ADDRESS REGISTER: ffffffff
ERROR REGISTER 1: ffffffff
ERROR REGISTER 2: ffffffff
```

Step	Procedure
1	Extract <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where a is the sequence number.
2	Note the location of the suspected faulty circuit pack.
3	If the error flag is a 0B and is an isolated failure, no action is necessary. If not 0B, continue to Step 4.
4	Check the update cables for proper connection.
5	Request diagnostics for the off-line main store. Enter: In PDS, DGN:CU[1,0],MASC[1,0],TLP! In MML, DGN:CU=[1,0],MASC=[1,0]:DATA,TLP;
6	If standard diagnostic phases do not indicate a failure, request demand phases for MASC 33, 34, 36, 39, 40, 93, and 95.

Procedure 7 – Software Fault

This procedure is valid only for the following output message:

```
REPT CU [1,0] ERROR INTERRUPT [15,16,17,18,19,1A,1C,1D] [a]  
where a is the sequence number.
```

The following is an example of an output message:

```
23 REPT BAD ALIGNMENT ON MEMORY REFERENCE X'18  
ONLINE CU: 1  
FAULTY UNIT: CU 0 CC 0
```

SAVE STATE

```
PA: 00000024 PSW: 44934345 PSBR: 7E055C0A SSBR: 7E055C0A
```

GENERAL REGISTERS

```
0: 00141B80 1: 00000010 2: 0000008C 3: 006A0018  
4: 00000000 5: 00647854 6: 00000002 7: 00000001  
8: 00020001 9: 006A00B0 A: 005A0164 B: 006A0168  
C: 00000000 D: 00000000 E: 00000000 F: 00000000  
UTILITY ID: 00000000 PROCESS NUMBER: 00010037
```

Step	Procedure
1	Extract <i>ERLOG</i> file entry. Enter: In PDS, OP:LOG;LG "ERLOG":KW "a"! In MML, OP:LOG:ERLOG:DATA,KW=a; Where a is the sequence number.
2	If the REPT CU message prints out randomly, record the information (message, date, time, unit, and so forth). This data is helpful if another interrupt of the same type occurs.

Step	Procedure
3	In the output message, the Program Address (PA), Processor Status Word (PSW), UID, and process number are the clues to a repeating software problem. However, identified software problems cannot normally be resolved at the office level. Therefore, complete the following: (a) Note any unusual activity that preceded the interrupt. (b) Save the output message. (c) Perform an off-line dump, if requested by your technical assistance organization. (d) Forward all information to the next appropriate level.
4	Repeated random interrupts of this type may indicate a hardware problem. If a hardware problem is suspected, request the appropriate demand diagnostics.

Error Interrupt Handler File

The *ERLOG* contains a history of all error interrupts handled by EIH that are not associated with the main memory. Among the errors handled are Input/Output (IO) errors, DMA errors, other CU hardware errors, and software errors. Each error message is preceded by the header REPT ERSLOT. (Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.)

This section is divided into the following three parts:

- On-Line Hardware Faults
- Off-Line Hardware Faults
- On-Line Software Faults.

The reason for these subsections is that they correspond to the bits in the Interrupt Source (IS) register.

IS Bit	Fault Classification
0	On-line hardware
1	Off-line hardware
2	On-line software

Depending on which of these bits is asserted, the EIH will gain control of the computer at a predetermined error handling routine. The locations of the error handling routines are placed in the interrupt vector table on a system initialization or on completion of a CU restore (bit 1 only).

In most instances when an error occurs, a message providing details about the error is printed on the ROP. Consider, as an example, the following software generated error:

```
08 REPT ERSLOT      ← for Release 6.3 and later
08 REPT ILLEGAL INSTRUCTION
ILLEGAL INSTRUCTION  X'17  ← for Release 6.3 and later
ONLINE CU:  CU 0
FAULTY UNIT:  CU 0
TIME OF ERROR:  02/15/86 17:08:55

                                SAVE STATE
PA: 007A3F82   PSW: 4AE5FC90      PSBR: FE05AE0A   SSBR: FE05AE0A

                                GENERAL REGISTERS
0: 00000000    1: 000440A0      2: 00000000      3: 005C0018
4: 005C00B0    5: 005C4200      6: 00004000      7: 005C4214
8: 0004403C    9: 006A0694      A: 006A06D0      B: 006A06D0
C: 00000005    D: 00000000      E: 005C4278      F: 00000000
UTILITY ID: 000000C0   PROCESS NUMBER: 00000009
REAL TIME CLOCK: F1D4740A
```

The message is written to a log file (*ERLOG*) in addition to being printed on the ROP. Having a log file of messages serves the following purposes:

- If the message is lost and never printed on the ROP, it will be preserved in the log file.
- Having a file containing the error messages allows one to access more than one message at a time.

On-Line Hardware Faults

The on-line hardware error handling capabilities of the EIH process can be divided into the following two classes:

- Memory errors – discussed in the “Memory History Log File” section.
- All remaining on-line hardware related errors – discussed in this section.

The class of on-line hardware faults that can be handled is limited to DMA detected errors and IO channel errors.

There are three models of 3B20D computers with different DMACs. The 3B20D Model 1 computer DMA has four circuit packs (UN36, UN37, and two UN35s). The 3B20D Model 2 and 3 computers have one circuit pack (UN46). To the 3B20D computer software using the DMA, there are no apparent functional differences other than the number of DMA channels supported. There are differences, however, in the initialization and error handling procedures.

Because of these differences, the error reports printed on certain DMA errors differ slightly between the 3B20D Model 1, 2, and 3 computers. The format for the 3B21D computer is the same as that for the 3B20D Model 2 and 3 computers.

In the following examples, instances of the log file entries are given for all DMAC models. Generally, a CHANNEL DATA REGISTER entry in the log file indicates the 3B20D Model 2 and 3 computers or the 3B21D computer DMA. THE DMA CHANNEL STATUS REGISTER applies only to the 3B20D Model 1 computer. For 3B20D Model 2 and 3 computers and the 3B21D computer, this information is contained in the DMA STATUS REGISTER and the CHANNEL DATA REGISTER. (Refer to Tables 4-28 through 4-30.)

3B20D Model 1 Computer Fault Example

```
16 DIO - ACKNOWLEDGE FAILURE  X'137
ONLINE CU:  CU 0
FAULTY UNIT:  DFC 0
TIME OF ERROR:  07/02/86 10:12:09
DMA ADDRESS:  23    DMA STATUS REGISTER:  00F07D7F
DMA CHANNEL STATUS REGISTER:  00008A47
CHANNEL ADDRESS:  25    CHANNEL STATUS REGISTER:  00000049
DMA RAM:  0008FF00  00A6FFF8  0008FA0C  8602BC00
DMA STORE ADDRESS REGISTER:  0002E420
DMA STORE DATA REGISTER:  FB02B800
DMA INCREMENTOR:  0008FA10    DMA DIO DATA REGISTER:  000008E4
DMA BUFFER REGISTER:  20000600    DMA MASK REGISTER:  00000000
REAL TIME CLOCK:  F286F902
```

3B20D Model 2 and 3 Computers and 3B21D Computer Fault Examples

```
43 DIO - ACKNOWLEDGE FAILURE    X'AC
ONLINE CU:  CU 0
FAULTY UNIT:  IOP 0
TIME OF ERROR:  06/30/86 14:43:06
DMA ADDRESS:  23    DMA STATUS REGISTER:  DFFDC4EF
CHANNEL DATA REGISTER:  00000305
CHANNEL ADDRESS:  25    CHANNEL STATUS REGISTER:  00000049
DMA RAM:  00130C00  00000000  00130C00  841F86C1
DMA STORE ADDRESS REGISTER:  0002E420
DMA STORE DATA REGISTER:  FB02B800
DMA INCREMENTOR:  0008FA10    DMA DIO DATA REGISTER:  000008E4
DMA BUFFER REGISTER:  20000600    DMA MASK REGISTER:  00000000
REAL TIME CLOCK:  0B647FDE
```

The previous examples represent typical nonfatal DMA errors; that is, those that implicate a device. Fatal errors are generally attributed to the DMA hardware and result in a system initialization.

In the 3B20D Model 2 and 3 computers and the 3B21D computer, if the DMA status register indicates any error except the DIO errors, then the channel data register has no meaning. If there are no errors other than the DIO errors and the low three bits of the channel data register are a five, then the DIO error bits in the DMA status register are valid. If the DMA status register shows no errors, then channel data register values one through four in the low three bits indicate the error source.

The DMA information provided in lines four through eight is described in Table 4-10.

The following is an example of an on-line hardware fault log entry for an error that does not involve the DMAC:

```
23 3/6 MAIN CHANNEL IO RESPONSE ERROR    X'13
ONLINE CU:  CU 0
FAULTY UNIT:  CU 0 CC 0
TIME OF ERROR:  06/25/86 15:12:29
HSR: 00000400
```

In this example, the problem is not with the DMA. More than one channel responded when one channel was addressed. There may be a problem with the channel or the error may be transient. A single IO response error will not cause the system to switch to a new processor or prevent it from functioning. Diagnostics should be run on the implicated CU. The HSR layout is shown in Table 4-36 and may be useful in identifying the faulty channel. General fault correction procedures are given in Table 4-3.

Table 4-10. Direct Memory Access Information

Log Entry Designation	Description
DMA address	IO address of the DMAC encountering the fault.
DMA controller status register	DMA status register. Refer to Tables 4-28 and 4-29 for layouts of the register.
DMA channel status register	Status of the channel under the DMAC that encountered the fault. Refer to Table 4-27 for the layout of the register. This register is in the DMA (3B20D Model 1 computer only).
Channel data register (3B20D Models 2 and 3 computers)	Value stored in the DMA channel data register when the error occurred. It contains supplemental error information.
Channel address	IO address of the channel the DMA believed caused the problem.
Channel status register	Status of the DMA channel. This register is internal to the channel (UN9). Refer to Table 4-31 for the layout of the register.
DMA RAM	The four-word entry in the DMA RAM for the device that failed.
DMAC store data register	Internal DMAC register.
DMA incrementer	Internal DMAC register.
DMA buffer	Internal DMAC register.
DMA store address register	Internal DMAC register.
DMA mask register	Internal DMAC register.
DMA DIO data register	Internal DMAC register.

Off-Line Hardware Faults

Off-line hardware faults are of the following classes:

- Memory related
- Violation of maintenance channel protocol.

The system must be in the Active/Standby (ACT/STBY) configuration before any of the errors in the off-line CU can be detected. When the system is not in the ACT/STBY configuration, the EIH process detaches from off-line interrupts and the memory of the off-line is not being updated.

When the system is in the ACT/STBY configuration, the EIH process attaches to off-line interrupts and the off-line memory is placed in the update mode. When the update circuit to the off-line CU is enabled, every memory read and write in the on-line CU's memory is also performed via the update circuit to the off-line memory. This allows the hardware to check both the off-line and on-line memories for parity errors and read/write completion while at the same time keeping both memories identical in content.

The act of memory update is sufficient to ensure that the off-line memory controller and memory are functional. Any remaining hardware not associated with memory update is not checked by the self-check hardware, nor is it handled by the interrupt handler process. The remaining hardware is routinely tested by the audit process in the active CU, on the processor soft switch, and by automatic diagnostics.

The types of errors which can occur in the standby CU are the following:

Error Type	Description
X'B	Invalid maintenance channel order
X'C	Other store hardware entry
X'D	Single-bit parity error (OTHSERD)
X'E	Multiple-bit parity error (OTHSERC)
X'F	Memory controller time-out (other store time-out).

The errors occur in the off-line CU but are detected by the on-line CU. There is one additional error that signals a problem in the off-line memory but is detected as an on-line error. This error is the other-store-out-of-range error (X'18). This error may occur if the off-line CU is equipped with less memory than the amount the on-line CU is actively using. Detection of the error is on a system access in the on-line CU, and the error is classified as a software error.

General fault interpretation and correction procedures are given in Table 4-3. These procedures may be used regardless of whether a parity error is from the on-line or off-line CU. As an example, consider the other store time-out error given below. This error signals a problem with the memory controller hardware or the path to the other CU's memory. There is no register data provided since this is a gross problem where the off-line memory is simply not responding. The result of this error is that the off-line CU will be removed from the standby state and placed in the Out of Service (OOS) state.

```
15 REPT ERSLOT
OTHER STORE TIMEOUT ERROR X'19
ONLINE CU: CU 1
FAULTY UNIT: CU 0 MASC 0
TIME OF ERROR: 06/18/86 17:25:47
REAL TIME CLOCK: 78f50c0f
```

On-Line Software Faults

In the 3B20D/3B21D computers, software errors are of the following basic types:

- Errors detected by the hardware that cause the EIH process to be dispatched.
- Errors detected by the microcode during the execution of a processor instruction.

The software fault entry of the EIH process is attached to IS bit 2 of the IS register. A software error asserts this bit and causes the software fault routine to gain control of the computer. The microcode also asserts IS bit 2 when it detects an error. The result is identical to that when IS bit 2 is asserted by the hardware. On a microcode detected error, an additional step is performed that is not performed if the error is detected by the hardware. This step is to leave a pattern in one of the temporary registers for the EIH. This allows the error handler to determine the type of error encountered by the microcode. This is not a problem when the hardware detects the error since the error register has this information.

When a process performs some action that causes a software error, its execution is suspended and the EIH gains control. The state of the suspended process is placed on the interrupt stack. When the error handler determines the error type, it also clears out the error indications in the error register and temporary registers used to pass data by microcode.

To document the error, data must be collected about the error and the process that caused it. This information is readily available from the process' saved state on the interrupt stack. The saved state consists of the PA, PSW, PSBR, SSBR, and general register. A UID and process number are also determined at this time by accessing the process control block of the offending process. Once the data is collected, it is placed in an area of physical memory where at some later time a user-level process will extract this data and print it on the ROP and the *ERLOG* file.

The EIH process performs one last function before giving up control of the computer. A request is made to the operating system to fault the process responsible for the error. The next time the faulty process is allowed to execute, it will resume execution in its fault entry. There it must decide what initialization/action is needed to become functional again.

The following example is a typical error message, which was generated when a process tried to execute an instruction that could not be translated (instruction does not exist). The information in this output message is sufficient to identify the process causing the error and the location in the process (via the PA) where the problem exists.

```
08 ERSLOT
ILLEGAL INSTRUCTION    X'17
ONLINE CU:  0
FAULTY UNIT: CU 0
TIME OF ERROR: 02/15/86 17:08:55

                SAVE STATE
PA: 007A3F82  PSW: 4AE5FC90      PSBR: FE05AE0A  SSBR: FE05AE0A

                GENERAL REGISTERS
0: 00000000  1: 000440A0      2: 00000000      3: 005C0018
4: 005C00B0  5: 005C4200      6: 00004000      7: 005C4214
8: 0004403C  9: 006A0694      A: 006A06D0      B: 006A06D0
C: 00000005  D: 00000000      E: 005C4278      F: 00000000
UTILITY ID:  000000C0  PROCESS NUMBER:  00000009
REAL TIME CLOCK:  F1D4740A
```

Automatic Postmortem Dump Analysis

The generation and content of automatic postmortem dumps are described in the "Automatic Postmortem Dumps" section. Automatic postmortem dumps are labeled either POSTMORTEM DUMP or MAINTENANCE INTERRUPT. Each type is placed in the log file *PMLOG*. In the case of a REPT CU MAINTENANCE INTERRUPT *d* message, the sequence number *d* can be used in the **OP:LOG** command to extract the entry from *PMLOG*. For the postmortem case, the sequence number is embedded in the DD PRM (All Seems Well PRM) message issued during processor initialization. (Refer to the "Processor Recovery Message Analysis" section.)

Access and analyze an automatic postmortem dump as follows:

Step	Procedure
1	Generate a printout of the dump by executing the OP:LOG command on the <i>PMLOG</i> log file. Use keyword or TIME and DATE ranges to select appropriate entries. For detailed information regarding the use of this command, refer to the appropriate manual page in 254-303-110, <i>PDS Input Messages Manual</i>, or 254-303-112, <i>MML Input Messages Manual</i>, or use the craft shell's help capability feature.
2	Analyze the INITIALIZATION MESSAGE section of the dump to determine the source of the initialization. ▪ If the SOURCE field designation is shown as a <i>p</i>, the postmortem dump contains little additional information since the Sanity Timer (TIMERS ONLINE) indicates the recovery was initiated by a time-out. No further analysis is necessary. When a Control Unit (CU) initializes because of a powerup, it will not immediately boot. Instead, the microcode will indicate in low core that a power clear (the CU may have powerup) has occurred and then halt. If the other CU is not active or cannot clear the Sanity Timer of this CU, then, this CU will boot and the SOURCE will be set to p. For unexplainable "power clear" MRFs, suspect a faulty CC power switch circuit pack or faulty Special Register (SREG) (UN2 for the 3B20D computer and KLW31 for the 3B21D computer). ▪ If the SOURCE field designation is shown as an <i>m</i>, the postmortem dump probably contains little additional information since the System Status Register (SSR) indicates the recovery was initiated by manual intervention [Emergency Action Interface (EAI) commands at the MTTY]. Therefore, no further analysis is necessary, and Step 3 may be performed if desired. However, if a false manual request is indicated (SOURCE = m, but no requests were made), suspect a faulty EAI, SREG0, or either Maintenance Teletypewriter Controller (MTTYC) [in either Input/Output Processor (IOP)]. ▪ If the SOURCE is an <i>s</i>, complete Steps 4 through 8. ▪ If the SOURCE is an <i>h</i>, complete Steps 7 and 8.

Step	Procedure
	The INITIALIZATION MESSAGE also indicates which CU was on-line, the faulty CU, the <i>UNIX</i> RTR operating system initialization level, and the application initialization level. Follow local practices for using this information.
3	Analyze the EAI buffer (eaipar). The buffer describes the EAI input parameter buffer for inputting manual requests to <i>UNIX</i> RTR operating system software from the craft interface terminal. The single bit defined for the parameters under EAI BUFFER can be found in Table 4-2. For <i>UNIX</i> RTR Operating System Release 6.3 and later, all parameters are from the previous manual initialization (except the APPL PAR and the EAINEW bit in the configuration field). The APPL PAR is from the current manual request. The EAINEW bit is set by the boot program on the current manual request. It is reset by the Application Integrity Monitor (AIM) when it has reacted to the manual initialization or at the end of the initialization interval by the EIH.
4	Complete this step only for software requested initializations (SOURCE s). Analyze the REQUESTED INITIALIZATION PARAMETERS section of the dump, and forward the information to your technical assistance organization. This section repeats the source, <i>UNIX</i> RTR operating system initialization level, and application initialization level information presented in the initialization message. In addition, a panic code, the phase options, and the UID are given. Use these two pieces of information to determine the process that invoked the phase OST and the problem that it encountered by using the following procedure: (a) Locate the UID value in Table 4-11. The process name is in parentheses below the ID value. (b) For the appropriate UID, locate the panic code in Table 4-11. The problem is identified in the column to the right of the panic code. Figures 4-1 and 4-2 illustrate a software initialization caused by the disk driver (UID = C1) because of a failed limp mode initialization (panic code = 6).

Step	Procedure
5	Check for acceptable values. The BOOT DEVICE fields indicate the previous and current active boot devices. For releases prior to <i>UNIX</i> RTR Operating System Release 21.27, acceptable values for these fields are 0 (boot device 0), 1 (boot device 1), and b (both boot devices). For <i>UNIX</i> RTR Operating System Release 21.27 and later, acceptable values for the BOOT DEVICE fields are MHD numbers in the range of 0x0 through 0xf or an asterisk. An asterisk indicates the mate of the boot device was not active at boot.
6	If the source of the fault was software (refer to Step 1), forward the information in the postmortem dump to your technical assistance organization. If the source was <i>m</i>, disregard the GENERAL REGISTERS information.
7	Analyze the FAULTY CU REGISTERS section of the dump. Although the saved state information is mainly of interest if the source is <i>h</i>, the System Status Register (SSR) and ER may also be of interest on software-requested initializations. (Follow local practices for using the saved state information.) The saved state information is stored in 16 different registers. The layouts of most registers are shown in the "Register Layouts" section and are arranged in alphabetical order by register name. Registers not shown in the "Register Layouts" section have functions that are explained by the register name and contain 32 bits of information. Table 4-12 lists and defines the acronym of each register and provides additional information. If an error is not obvious, each timer register may be checked for a sanity time-out. The timer registers are normal counters (active 1) and are periodically reset. The registers are printed for each side relative to the current configuration; therefore, the off-line timer may be checked for an SAS error. The sanity time-out bits are bits 6 and 7 in the timer register. Bit 6 going active produces an SAS error, while bit 7 going active causes an MRF on that side. If the processor is not active (SSR bit 16 = 0), then bit 6 going active does not cause an SAS error.

Step	Procedure

- 8 If the ER in the microcode save state registers section indicates that a MYSERA or a my store time-out occurred (bit 6 = 0 or bit 7 = 0, respectively), the MAIN STORE REGISTERS provide additional information about the error. Follow local practices for using this information. The 3B20D computer main store register acronyms are as follows:

Acronym	Description
ER01	Main store controller 0, error register 1
ER02	Main store controller 0, error register 2
ER11	Main store controller 1, error register 1
ER12	Main store controller 1, error register 2
TAR0	Trapped address register 0
TAR1	Trapped address register 1

The main store register acronyms for the 3B21D computer are as follows:

Acronym	Description
MC_CMD	Memory Control Command Register
MC_STAT	Memory Control Status Register
MC_ERADD	Memory Control Error Address Register
MC_ER	Memory Control Error Register
DP_CMD	Data Path Command Register
DP_STAT	Data Path Status Register
DP_ERDAT	Data Path Error Data Register
BANK0	Memory Control Bank 0 Address Boundary Register
BANK1	Memory Control Bank 1 Address Boundary Register
BANK2	Memory Control Bank 2 Address Boundary Register
BANK3	Memory Control Bank 3 Address Boundary Register
BANK4	Memory Control Bank 4 Address Boundary Register
BANK5	Memory Control Bank 5 Address Boundary Register
BANK6	Memory Control Bank 6 Address Boundary Register
BANK7	Memory Control Bank 7 Address Boundary Register

Figures 4-3 and 4-4 show a postmortem dump for a hardware source initialization caused by a microinterrupt; in this case, a page fault (UER1 equals eight) occurs while interrupts are blocked. The interrupt mask is all "1s"; therefore, no interrupts can fire. This is a typical example of kernel stack overflow. Figures 4-5 and 4-6 show a manually requested initialization for releases prior to *UNIX* RTR Operating System Release 21.27.

Figures 4-7 and 4-8 show a manually requested initialization for *UNIX RTR* Operating System Release 21.27 and later.

```

                                MAINTENANCE INTERRUPT X'39
TIME OF INTERRUPT:  MON JUN 28 21:39:16 1986

                                INITIALIZATION MESSAGE
SOURCE:  s ONLINE CU: 1  FAULTY CU: 1
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 1

                                REQUESTED INITIALIZATION PARAMETERS
SOURCE:  s          PANIC CODE: 6
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 000000C1

                                BOOT DEVICES
PREVIOUS: 0          CURRENT: 0

                                EAI BUFFER (eaipar)
MICROCODE: 40      INIT LVL: 04      INHIBIT: 00
CONFIGUR: 41      APPL PAR:          PERIODIC: 00

                                TIMERS
ONLINE: 57192B33      OFFLINE: 8180131D

                                GENERAL REGISTERS
0: 00000000      1: 000000C1      2: 0000001F      3: 00000000
4: 00141780      5: 00000040      6: 00000006      7: 00000000
8: 04000000      9: 006A1274      A: 006A12B8      B: 006A12D8
C: D8A7D8A7      D: 00000000      E: 00000000      F: 00000000

                                FAULTY CU REGISTERS
PA: 00000580      SAR: 00000580      SSR: 7F5AFFBA      PSW: 0AF58098
SCR: FFFD07FF      ER: FFFFDFFF      IB: DD00DD00      SIR: 5400A180
PPR: 00000000      HSR: 00009409      SDR: 0000057E      IM: FFC1FFDA
IS: FFFFFFFF      UER: 4FBDFDFF      UER1: 006A10F4      CDR: 00000000

                                INTERRUPT STACK SAVED STATE
PA: 0000E8A0      PSW: 4AE59090      PSBR: FE01EC0A      UTILID: 000000C1
0: FFFFFFFF      1: 00000050      2: 0000001F      3: 00000000
4: 00141780      5: 00000040      6: 00000006      7: 00000000
8: 04000000      9: 006A1220      A: 006A1264      B: 006A1264
C: 006A119C      D: 00000005      E: 00021000      F: 006A1264

                                OFFLINE REGISTERS
PA: 0000B804      PSW: 00F50003      SSR: 2F69DF73      ER: 4FEFFFF8

                                MAIN STORE REGISTERS
ER01: 00000000      ER02: 206B0000      TAR0: 0597FEFD
ER11: 00000000      ER12: 206AD000      TAR1: 41F7FEFD

REAL TIME CLOCK: 013CB85C
```

Figure 4-1. 3B20D Computer Postmortem Dump (Software Requested Initialization) - Releases Prior to Release 21.27

```

                                MAINTENANCE INTERRUPT X'39
TIME OF INTERRUPT:  MON JUN 28 21:39:16 1986

                                INITIALIZATION MESSAGE
SOURCE:  s  ONLINE CU: 1  FAULTY CU: 1
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 1

                                REQUESTED INITIALIZATION PARAMETERS
SOURCE:  s  PANIC CODE: 6
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 000000C1

                                BOOT DEVICES
PREVIOUS: 0  CURRENT: 0

                                EAI BUFFER (eaipar)
MICROCODE: 40  INIT LVL: 04  INHIBIT: 00
CONFIGUR: 41  APPL PAR:  PERIODIC: 00

                                TIMERS
ONLINE: 57192B33  OFFLINE: 8180131D

                                GENERAL REGISTERS
0: 00000000  1: 000000C1  2: 0000001F  3: 00000000
4: 00141780  5: 00000040  6: 00000006  7: 00000000
8: 04000000  9: 006A1274  A: 006A12B8  B: 006A12D8
C: D8A7D8A7  D: 00000000  E: 00000000  F: 00000000

                                FAULTY CU REGISTERS
PA: 00000580  SAR: 00000580  SSR: 7F5AFFBA  PSW: 0AF58098
SCR: FFFD07FF  ER: FFFDFDFF  IB: DD0DD000  SIR: 5400A180
PPR: 00000000  HSR: 00009409  SDR: 0000057E  IM: FFC1FFDA
IS: FFFFFFFE  UER: 4FBFDFFF  UER1: 006A10F4  CDR: 00000000
EAI: 00000000

                                MAIN STORE REGISTERS
MCMD: 400115fe  MSTAT: 6005af08  MERADD: 041e8810  MER: 00000000
DCMD: 00000003  DSTAT: 041e8810  DERDAT: 00000002  MMR: f378a00a
BANK0: 00010000  BANK1: 00020010  BANK2: 00024020  BANK3: 00028024
BANK4: 00028028  BANK5: 00028028  BANK6: 00028028  BANK7: 00028028

                                CACHE REGISTERS
CRA: fffc0840  CRB: fffc0117  CRC: ffff256b  CRD: ffff000d
CRE: fffc1000  CRF: fffc2001  CRG: fffc2003

                                INTERRUPT STACK SAVED STATE
PA: 0000E8A0  PSW: 4AE59090  PSBR: FE01EC0A  UTILID: 000000C1
0: FFFFFFFF  1: 00000050  2: 0000001F  3: 00000000
4: 00141780  5: 00000040  6: 00000006  7: 00000000
8: 04000000  9: 006A1220  A: 006A1264  B: 006A1264
C: 006A119C  D: 00000005  E: 00021000  F: 006A1264

                                OFFLINE REGISTERS
PA: 0000B804  PSW: 00F50003  SSR: 2F69DF73  ER: 4FEFFFF8

```

Figure 4-2. 3B21D Computer Postmortem Dump (Software Requested Initialization) - Releases Prior to Release 21.27

```

                                MAINTENANCE INTERRUPT X'33
TIME OF INTERRUPT:  MON JUN 14 11:29:16 1986

                                INITIALIZATION MESSAGE
SOURCE:  h ONLINE CU: 1  FAULTY CU: 1
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 1

                                REQUESTED INITIALIZATION PARAMETERS
SOURCE:                                PANIC CODE:  0
UNIX RTR LEVEL: 0  APPLICATION LEVEL: 0
PHASE OPTIONS:  0
UTILITY ID: 00000000

                                BOOT DEVICES
PREVIOUS:  b      CURRENT:  b

                                EAI BUFFER (eaipar)
MICROCODE: 40    INIT LVL: 04    INHIBIT:  00
CONFIGUR:  43    APPL PAR:      PERIODIC: 00

                                TIMERS
ONLINE: 84310C24    OFFLINE: 81803B0D

                                GENERAL REGISTERS
0: 0062BD00    1: 0062BD00    2: 0000001F    3: 00000000
4: 00362F40    5: 00000040    6: 0FF50098    7: 00000000
8: 20000000    9: 006A3FF0    A: 006A4028    B: 006A4028
C: D8A7D8A7    D: 00000000    E: 00000000    F: 00000000

                                FAULTY CU REGISTERS
PA: 0000010E    SAR: 006A402C    SSR: 5F5EFFFF    PSW: 4FF504D0
SCR: FFFB6FFF    ER: BFFFDFFF    IB: 76017000    SIR: 7000DEDE
PPR: 00000000    HSR: 00009402    SDR: 00000000    IM: FFFFFFFF
IS: FFFFFFFF    UER: FFFFFFFF    UER1: 00000008    CDR: 00000000

                                INTERRUPT STACK SAVED STATE
PA: 00002ECC    PSW: 0BF52090    PSBR: FE0116E4    UTILID: 00000000
0: 0062BD00    1: 0062BD00    2: 0000001F    3: 00000000
4: 00362F40    5: 00000040    6: 0FF50098    7: 00000000
8: 20000000    9: 006A3FF0    A: 006A3FF0    B: 006A3FFC
C: 000002F4    D: 00000004    E: 00000000    F: 006A3FF0

                                OFFLINE REGISTERS
PA: 0000B804    PSW: 00F50003    SSR: 2F69DFF3    ER: 5FEFFFF8

                                MAIN STORE REGISTERS
ER01: 00000000  ER02: 206A9000  TAR0: 0597FEFD
ER11: 00000000  ER12: 206A6000  TAR1: 41F7FEFD

REAL TIME CLOCK: 0126E034

Note:  Notice kernel stack overflow.
```

Figure 4-3. 3B20D Computer Postmortem Dump (Hardware Source Caused by Microinterrupt) - Releases Prior to Release 21.27

```
POSTMORTEM DUMP   X'6d
TIME OF POSTMORT: 01/06/93 16:27:42

INITIALIZATION MESSAGE
SOURCE: h ONLINE CU: 0      FAULTY CU: 1
UNIX RTR LEVEL: 2    APPLICATION LEVEL: 1

REQUESTED INITIALIZATION PARAMETERS
SOURCE: PANIC CODE: 0
UNIX RTR LEVEL: 0    APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 00000000

BOOT DEVICES
PREVIOUS: b CURRENT: 0

EAI BUFFER (eaipar)
MICROCODE: 40      INIT LVL: 04    INHIBIT: 00
CONFIGUR: 07      APPL PAR: PERIODIC: 00

TIMERS
ONLINE: 35724300    OFFLINE: 73910a00

GENERAL REGISTERS
0: 11111111 1: 00000000 2: 00000031 3: ffffffffcb
4: 00000000 5: 4ff52490 6: 00000000 7: 00021a44
8: 000801e0 9: 006a05b4 A: 006a05ec B: 006a05ec
C: 00000000 D: 00000000 E: 00000000 F: 00000000

FAULTY CU REGISTERS
PA: 00007e7a SAR: 041e8800 SSR: af3bef0a PSW: 00f50000
SCR: 7febaf00 ER: 1fffffff IB: 51080000 SIR: 0000001c
PPR: 00000000 HSR: 00008c08 SDR: 9f3bef0a IM: ffffffff
IS: ffffffff UER: 1fff7fff UER1: 041e8800 CDR: 44550080
EAI: 000000ff

MAIN STORE REGISTERS
MCMD: 00000000 MSTAT: 00000000 MERADD: 00000000 MER: 00000000
DCMD: 00000000 DSTAT: 00000000 DERDAT: 00000000 MMR: 00000000
BANK0: 00000000 BANK1: 00000000 BANK2: 00000000 BANK3: 00000000
BANK4: 00000000 BANK5: 00000000 BANK6: 00000000 BANK7: 00000000

CACHE REGISTERS
CRA: 11111111 CRB: 00000000 CRC: 00000031 CRD: ffffffffcb
CRE: 00000000 CRF: fffc2001 CRG: 4ff52490

INTERRUPT STACK SAVED STATE
PA: 0000262a PSW: 0df520d5 PSBR: 0004e950 UTILID: 0000000a
0: 00140380 1: 000263cb 2: 00000000 3: 00000001
4: 00140600 5: 00000000 6: 00000000 7: 00140380
8: 0000000d 9: 006a0350 A: 006a038c B: 006a038c
C: 00000000 D: ffffffff E: 00000380 F: 00000034

OFFLINE REGISTERS
PA: 00007e7a PSW: 00f50000 SSR: af3bef0a ER: 1fffffff

REAL TIME CLOCK: 2c92a788
```

Figure 4-4. 3B21D Computer Postmortem Dump (Hardware Source Caused by Microinterrupt) - Releases Prior to Release 21.27

```

                                POSTMORTEM DUMP          X'3C
TIME OF POSTMORT:  FRI JAN  2 11:06:55 1986

                                INITIALIZATION MESSAGE
SOURCE: m ONLINE CU: 0  FAULTY CU: 0
UNIX RTR LEVEL: 1  APPLICATION LEVEL: 1

                                REQUESTED INITIALIZATION PARAMETERS
SOURCE:                PANIC CODE: 0
UNIX RTR LEVEL: 0  APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 00000000

                                BOOT DEVICES
PREVIOUS: 0  CURRENT: 0

                                EAI BUFFER (eaipar)
MICROCODE: 40  INIT LVL: 04  INHIBIT: 00
CONFIGUR: 41  APPL PAR:      PERIODIC: 00

                                TIMERS
ONLINE: 5974312D  OFFLINE: 96560B18

                                GENERAL REGISTERS
0: 00000001  1: 00647F9A  2: 00660004  3: 0000048C
4: 00000003  5: 00000000  6: 00660684  7: 00661620
8: 00660A04  9: 006A0284  A: 006A102C  B: 006A102C
C: D8A7D8A7  D: 00000000  E: 00000000  F: 00000000

                                FAULTY CU REGISTERS
PA: 0000F79E  SAR: 0000F79E  SSR: 5F5EBF3B  PSW: 00F500D9
SCR: FFFDEBAB  ER: BFFFFFFF  IB: 33680F40  SIR: 0F403368
PPR: 00000000  HSR: 00009408  SDR: 00000000  IM: 0001FBDA
IS: FFFFFFFF  UER: 4FBFFFFF  UER1: 006A1080  CDR: 00000000

                                INTERRUPT STACK SAVED STATE
PA: 00000020  PSW: 0FF500D0  PSBR: FE0116E4  UTILID: 00000000
0: 006A1000  1: 00000000  2: 00000000  3: 00000000
4: 00000000  5: 00000000  6: 00000000  7: 00000000
8: 00000000  9: 00000000  A: 00000000  B: 00000000
C: 00000000  D: 00000000  E: 00000000  F: 006A1000

                                OFFLINE REGISTERS
PA: 0000B804  PSW: 00F50003  SSR: 2F69D5B2  ER: 5FFFFFF8

                                MAIN STORE REGISTERS
ER01: 00000000  ER02: 206FD000  TAR0: 0597FEFD
ER11: 00000000  ER12: 206FA000  TAR1: 4AF7FEFD

REAL TIME CLOCK: 0140E160
```

**Figure 4-5. 3B20D Computer Postmortem Dump (Manual Request) - Releases
Prior to Release 21.27**

```
POSTMORTEM DUMP  X'4
TIME OF POSTMORT: 01/08/93 16:14:43

INITIALIZATION MESSAGE
SOURCE: m ONLINE CU: 1      FAULTY CU: 1
UNIX RTR LEVEL: 2    APPLICATION LEVEL: 1

REQUESTED INITIALIZATION PARAMETERS
SOURCE:      PANIC CODE: 0
UNIX RTR LEVEL: 0    APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 00000000

BOOT DEVICES
PREVIOUS: 0 CURRENT: 1

EAI BUFFER (eaipar)
MICROCODE: 40      INIT LVL: 04    INHIBIT: 00
CONFIGUR: 41      APPL PAR:      PERIODIC: 00

TIMERS
ONLINE: 02043001    OFFLINE: 46473c00

GENERAL REGISTERS
0: a00d2923 1: 00000010 2: 00000113 3: 00000258
4: 0000a800 5: a00d2923 6: 00000004 7: 00000008
8: 00000400 9: 0000de84 A: 0000dec0 B: e3757c80
C: d8a7d8a7 D: 00000000 E: 00000000 F: 00000000

FAULTY CU REGISTERS
PA: 000029a0 SAR: 00000144 SSR: db1a8b8a PSW: 00f50000
SCR: ffeff700 ER: 0fffffff IB: dd00dd00 SIR: 00000048
PPR: 00000000 HSR: 0000fc90 SDR: df0acb8a IM: ffffffff
IS: ffffffff UER: ffffffff UER1: ffffffff CDR: 00000000
EAI: 000000ff

MAIN STORE REGISTERS
MCMD: 400115fe MSTAT: 6005af08 MERADD: 041e8810 MER: 00000000
DCMD: 00000003 DSTAT: 041e8810 DERDAT: 00000002 MMR: fe78a00a
BANK0: 00010000 BANK1: 00020010 BANK2: 00024020 BANK3: 00028024
BANK4: 00028028 BANK5: 00028028 BANK6: 00028028 BANK7: 00028028

CACHE REGISTERS
CRA: fffc0840 CRB: fffc0117 CRC: ffff256b CRD: ffff000d
CRE: fffc1000 CRF: fffc2001 CRG: fffc2003

INTERRUPT STACK SAVED STATE
PA: ffc2ffff PSW: ff9bffff PSBR: de027dff UTILID: 00000001
0: a0ff0800 1: c5ff0e00 2: 00fa0000 3: 00f80000
4: ff87ffff 5: ff87ffff 6: fd00dfff 7: ff88ffff
8: 00ff0c00 9: 00ff3300 A: 00ef0000 B: 00fd0000
C: ffc5ffff D: ff00ffff E: fd004bff F: f70078ff

OFFLINE REGISTERS
PA: 0000061e PSW: 00f5000d SSR: ab2b8d0b ER: 0fffffff9

REAL TIME CLOCK: e37570fa
```

**Figure 4-6. 3B21D Computer Postmortem Dump (Manual Request) - Releases
Prior to Release 21.27**

```
POSTMORTEM DUMP    X'54
TIME OF POSTMORT:  04/20/98  11:31:37

INITIALIZATION MESSAGE
SOURCE: m ONLINE CU: 1      FAULTY CU: 1
UNIX RTR LEVEL: 4    APPLICATION LEVEL: 1

REQUESTED INITIALIZATION PARAMETERS
SOURCE:      PANIC CODE: 0
UNIX RTR LEVEL: 0    APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 00000000

BOOT DEVICES
PREVIOUS: * / 1    CURRENT: * / 1

EAI BUFFER (eaipar)
MICROCODE: 40      INIT LVL: 04    INHIBIT: 00
CONFIGUR: 46      APPL PAR:      PERIODIC: 00
BDID PAIR: 00

TIMERS
ONLINE: 88144c23    OFFLINE: 82260c2a

GENERAL REGISTERS
0: 400b2523  1: 00000010    2: 00000002    3: 00000111
4: 0000a800  5: 400b2523    6: 00000000    7: 00000001
8: 00000400  9: 006a023c    A: 006a002c    B: 006a0030
C: d8a7d8a7  D: 00000000    E: 00000000    F: 00000000

FAULTY CU REGISTERS
PA: 00010c96    SAR: 00010c96    SSR: 5b5eab38    PSW: 00f540d9
SCR: fbfdbf32    ER: bfffffff    IB: 92139213    SIR: 92135408
PPR: 00000000    HSR: 0000fc90    SDR: 00000000    IM: 0001fbda
IS: ffffffff    UER: ffffffff    UER1: ffffffff    CDR: 08110811

INTERRUPT STACK SAVED STATE
PA: 00000020    PSW: 0ff500d0    PSBR: 3f012a95    UTILID: 0000000a
0: 006a0000  1: 00000000    2: 00000000    3: 00000000
4: 00000000  5: 00000000    6: 00000000    7: 00000000
8: 00000000  9: 00000000    A: 00000000    B: 00000000
C: 00000000  D: 00000000    E: 00000000    F: 006a0000

OFFLINE REGISTERS
PA: 00010ffc    PSW: 00f50001    SSR: 3b298d31    ER: 0fffffff

MAIN STORE REGISTERS
ER01: 00000000    ER02: 00652000    TAR0: 0597fefb
ER11: 00000000    ER12: 0064f000    TAR1: 41f7fefb

REAL TIME CLOCK: 74fa3b0e
```

Figure 4-7. 3B20D Computer Postmortem Dump (Manual Request) - Release 21.27 and Later

```
POSTMORTEM DUMP   X'21a
TIME OF POSTMORT: 04/16/98 14:05:22

INITIALIZATION MESSAGE
SOURCE: m ONLINE CU: 0      FAULTY CU: 0
UNIX RTR LEVEL: 4    APPLICATION LEVEL: 1

REQUESTED INITIALIZATION PARAMETERS
SOURCE: PANIC CODE: 0
UNIX RTR LEVEL: 0    APPLICATION LEVEL: 0
PHASE OPTIONS: 0
UTILITY ID: 00000000

BOOT DEVICES
PREVIOUS: 0 / 9    CURRENT: * / 9

EAI BUFFER (eaipar)
MICROCODE: 40    INIT LVL: 04    INHIBIT: 00
CONFIGUR: 47    APPL PAR:    PERIODIC: 00
BDID PAIR: 04

TIMERS
ONLINE: 7768bc37    OFFLINE: 02074c00

GENERAL REGISTERS
0: a40d2923  1: 00000010  2: 00000113  3: 00000258
4: 0000a800  5: a40d2923  6: 00000004  7: 00000008
8: 00000400  9: 006a04d8  A: 006a002c  B: 006a0030
C: d8a7d8a7  D: 00000000  E: 00000000  F: 00000000

FAULTY CU REGISTERS
PA: 00010c96    SAR: 00010c96    SSR: db1ebb7b    PSW: 00f500d9
SCR: fffdaf00    ER: 4fffffff    IB: 92139213    SIR: 92135408
PPR: 00000000    HSR: 0000fc00    SDR: 00000000    IM: 0000fbda
IS: ffffffff    UER: ffffffff    UER1: ffffffff    CDR: 00000000
EAI: 00000000

MAIN STORE REGISTERS
MCMD: 40019dfe    MSTAT: 200aaf08    MERADD: 041e8810    MER: 00000000
DCMD: 0000002f    DSTAT: 10101004    DERDAT: 53000000    MMR: 42788808
BANK0: 00010000    BANK1: 00020010    BANK2: 00030020    BANK3: 00040030
BANK4: 00040040    BANK5: 00040040    BANK6: 00040040    BANK7: 00040040

CACHE REGISTERS
CRA: fffc0000    CRB: fffc0000    CRC: fffc0000    CRD: fffc0000
CRE: fffc0000    CRF: fffc0000    CRG: fffc0001

INTERRUPT STACK SAVED STATE
PA: 00000020    PSW: 0ff500d0    PSBR: 0004fa54    UTILID: 0000000a
0: 006a0000  1: 00000000  2: 00000000  3: 00000000
4: 00000000  5: 00000000  6: 00000000  7: 00000000
8: 00000000  9: 00000000  A: 00000000  B: 00000000
C: 00000000  D: 00000000  E: 00000000  F: 006a0000

OFFLINE REGISTERS
PA: 00010ffc    PSW: 00f50000    SSR: ab2b8d7a    ER: 0fffffff

REAL TIME CLOCK: 17672321
```

Figure 4-8. 3B21D Computer Postmortem Dump (Manual Request) - Release 21.27 and Later

Table 4-11. Process/Problem Identification

Utility ID	Panic Code	Description
X'0 Kernel Boot (KBOOT)	1	Kboot cannot create a system process.
	2	Bad kernel process Dispatcher Control Table (DCT) pointer.
	3	Bad kernel process, Process Control Block (PCB) segment.
	4	Bad segment case.
	5	Bad supervisor process DCT pointer.
	6	Bad supervisor process PCB segment.
	7	Cannot send message to process manager.
	8	Cannot send message to file manager.
	9	Cannot send <i>pcreate</i> message.
	A	Bad boot segment ID.
	B	Bad kernel message segment ID.
	C	Bad kernel DCT segment ID.
	D	Bad kernel interrupt stack segment ID.
	E	Bad port table segment ID.
	F	Bad process ID.
	10	Cannot create Protected Application Segments (PAS).
	11	Cannot fill in segment table entry.
	12	PAS is too large.
	13	Cannot create a PUBLIC LIBRARY work segment.
	14	Boot process segment list full.
	15	Boot process segment list overlay.
	16	Invalid root directory block.
X'1 Memory Manager (MMGR)	17	No boot shared library.
	18	Invalid boot shared library inode.
	19	Cannot allocate boot shared library segment.
	1A	Disk read time-out.
	1B	DMA SG failure.
	1C	Disk read failure.
	1D	Cannot apply segment name.
	1E	Boot time database is missing.
	1F	Could not allocate a hole descriptor block segment.
	20	Not enough swappable memory.
	21	Could not allocate a PAS segment for a boot process because there are no PAS segments.
	22	Kernel PCB of a boot process does not have enough entries to accommodate either PAS or ECD segments.
	6E	Function to make space on disk failed.
	6F	Invalid return code from function to load segments.
	70	Failed reclaiming swap space.
	71	Unable to retry failed IO.
	72	Message drained out of sequence.
	73	Ran out of swap space by boot process.
	74	Ran out of entries describing free swap space on the swap device (HDBs).
	75	Unexpected fault.
	7F	Swap space damaged.

Table 4-11. Process/Problem Identification (Contd)

Utility ID	Panic Code	Description
X'2 Capability Manager (CMGR)	80	Capability manager unexpected fault.
X'3 Utility Manager (UMGR)	50	Utility manager unexpected fault.
X'4 File Manager (FMGR)	1 2 4 5 6 7 8 9 C D 10 11 12	No buffer segment. Could not read root super block. Could not read root inode. Cannot allocate initialization message to SIM. Could not connect to ECD. Stack switch failure. Bad mount table. Illegal block device. Mount point not found in mount table. Bad inode disk address. Too many nontask faults. Too many task faults. Too many kfault faults.
X'9 (SCHEDULER)	37 38 3A	IO error loading essential/no-term process. SCHEDULER unexpected fault. SCHEDULER received second a software fault within monitored interval.
X'A (Kernel)	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 1D	Excess number of illegal interrupts. Bad segment descriptor entry pointer. Faulting kernel. Unexpected interrupt. Unattached process during interrupt. Dispatching to a nonkernel process. Bad segment descriptor entry pointer. Faulting kernel. Null DCT pointer. Invalid branch to zero. Nonidentifiable kernel fault. Special process fault. Fault while handling supervisor OST. Fault while handling channel OST. Fault while handling kernel process OST. Fault in kernel audit. Null DCT pointer.
X'25 (KDAPDRVR)	40	Kernel Display Application (KDAP) driver special process unexpected fault.

Table 4-11. Process/Problem Identification (Contd)

Utility ID	Panic Code	Description
X'8C Little Boot (LBOOT)	1	Bus Interface Controller (BIC) for the boot Disk File Controller (DFC) initialization failed.
	4	Peripheral Interface Controller (PIC) for the boot DFC initialization failed.
	5	Failed to find the big boot file entry in the disk Volume Table Of Contents (VTOC).
	7	Failed to initialize the DMAC Random Access Memory (RAM).
	A	Failed to find an entry in the disk VTOC.
	B	Disk job timed out.
	10	Disk job failed.
	20	Page table initialization routine called with an invalid parameter.
	21	Channel error during programmed IO SG command to disk.
	22	Disk job failed.
X'8D Processor Init (PINIT)	2	Enabled hardware checks with an error source present.
	8	A <i>UNIX</i> RTR operating system level 3 phase will be called because the up-to-date off-line memory is not accessible after a CU SAS.
	9	Memory copy failed after a SAS.
X'8E Big Boot (BBOOT)	1	Failed to initialize the BIC for the boot DFC.
	2	Failed to initialize the PIC for the boot DFC.
	3	Failed to initialize the PIC for the boot DFC.
	4	Failed to find the disk VTOC entry for the file being loaded.
	5	Failed to find the boot file or a Boot Time Database (BTDB).
	6	Failed to find the boot file or a BTDB.
	7	Memory copy failed after a SAS.
	9	Boot file data on disk is no good.
	A	Failed to access <i>btparm</i> record from incore ECD.
	10	Boot ECD on disk is no good.
	20	Page table initialization routine called with an invalid parameter.
	21	Channel error during a programmed IO SG command to disk.
	22	Disk job failed.
	31	ECD: MASC 0 UCB not found for one or both CUs.
X'C0 IOP Driver (IODRV)	32	ECD: MASC 0 UCB has unknown memory array type for one or both CUs.
	33	ECD: MASC 1 UCB has unknown memory array type for one or both CUs.
	53	Boot image format invalid for release.
	63	Disk job timed out.
	1	Encountered a fault while handling a fault.
	2	Excessive software faults.
	3	Excessive unrecoverable errors while processing a fault.
	4	Excessive unrecoverable errors while processing a fault where IOPs were affected but not implicated.
	A	User requested, via the EAI page on MTTY, a <i>UNIX</i> RTR operating system level 0 application level 0 initialization.

Table 4-11. Process/Problem Identification (Contd)

Utility ID	Panic Code	Description
X'C1 Disk Driver (DKDRV)	1	CONFIG failed a <i>rpterr</i> call - deljob.c.
	2	CONFIG failed a <i>rpterr</i> call - jobrsp.c.
	3	Message buffer allocation failed - kevent.c.
	4	Disk driver software fault detected - kfault.c.
	5	CONFIG failed a <i>rpterr</i> call - kfault.c.
	6	Limp mode initialization failed - jobrsp.c.
	7	Limp mode initialization failed - deljob.c.
	8	Limp mode initialization failed - kfault.c.
	9	CONFIG failed a <i>rpterr</i> call - kfault.c.
	A	Limp mode initialization failed - kfault.c.
	B	Limp mode enable interrupt failed - kfault.c.
	C	CONFIG failed a <i>rpterr</i> call - kfault.c.
	D	Limp mode initialization failed - kfault.c.
	E	Nested faults have been detected - kfault.c.
	F	Fault rate is excessive - kfault.c.
X'83 Error Interrupt Handler (PCPEIH)	2	EIH failed to attach to the ECD.
	10	Channel error.
	20	DMA detected fatal hardware fault in on-line main store.
	21	DMA detected out-of-range memory reference.
	22	DMA detected noncorrectable parity failure in main store.
	26	DMA error: bad parity on read.
	27	DMA error: DMA RAM parity failure.
	28	DMA error: DMA incrementer parity failure.
	2C	DMA error: CCIO data parity failure.
	2D	DMA error: CCIO command failure.
	2E	DMA error: DMA Read-Only Memory (ROM) sequencer parity failure.
	4D	Boot disk (MHD) or SCSI Bus (SBUS) off-line or unavailable.
	51	ECD audit failure.
	55	Hard switch called with CU forced off-line.
	7E	ECD audit found errors in UCB of device type DMA, Dual Serial Channel (DSCH), or SCH.
	C4	Fault FLT_CMA remove request.
	C5	Fault FLT_CMB remove request.
	C6	Fault FLT_CMC remove request.
	C7	Fault FLT_CMD remove request.
	C8	Fault FLT_CMAM manual remove request.
	C9	Fault FLT_UCLRMV unconditional remove.
	CB	Fault FLT_SOFTSW softswitch request.
	E1	ECD access failure.
	E2	Zero transfer vector encountered.
	E4	ECD audit found errors in UCB data for which EIH is responsible.
	E5	A hard switch was called because of a reconfiguration request from CONFIG.
	E6	A hard switch was called because of a disk limp mode request from CONFIG.
	EA	On-Line Fault during OFLBOOT caused soft-switch.
	EF	Boot out of Off-line Boot request.

Table 4-11. Process/Problem Identification (Contd)

Utility ID	Panic Code	Description
X'85 Processor Audit (PCPAUD)	1	Executing the fault entry too many times. A phase will be called to protect against a kernel stack overflow.
	2	Executing the event entry too many times. A phase will be called to protect against a kernel stack overflow.
X'187 System Integrity Monitor (SIM)	1	Initialization failure.
	4	Audit recovery.
	09	Message buffer overflow.
	0B	Critical message buffer overload.
	C	Unrecognized faults.
	15*	EIH requested a phase during initialization interval.
	16*	Disk Driver (DKDRV) requested a phase during initialization interval.
	17*	IOP driver requested a phase during initialization interval.
	18*	FMGR requested a phase during initialization interval.
	19*	Kernel requested a phase during initialization interval. The kernel consists of the kernel service routines, memory manager, scheduler, kernel display application driver, utility manager, and capability manager.
	1A*	Process manager requested a phase during initialization interval.
	1E*	AIM requested a phase during initialization interval.
	C8	Application sanity failure.
	F5*	EIH requested a phase after initialization interval.
	F6*	DKDRV requested a phase after initialization interval.
	F7*	IOP driver requested a phase after initialization interval.
	F8*	FMGR requested a phase after initialization interval.
	F9*	Kernel requested a phase after initialization interval. The kernel consists of the kernel service routines, memory manager, scheduler, kernel display application driver, utility manager, and capability manager.
	FA*	Process manager requested a phase after initialization interval.
	FE*	AIM requested a phase after initialization interval.
X'26 Kernel Level Disk LLA Process (KLLA)	1	Too many kfault faults.
	2	Too many task faults.
	3	Too many nontask faults.
	4	Stack switch failure.

* There should be a failing PRM whose step number matches the second digit of the panic code with information associated with the system phase.

Table 4-12. Faulty CU Registers

Register	Description
PA	Program Address – This is the program address at the time of the initialization. For software requested initializations, this is the address of the PHASE OST and <i>not</i> the address where the problem was detected.
SAR	Store Address Register – This register is meaningful only if a main store problem is suspected.
SSR	System Status Register – Refer to Table 4-56 for the layout of this register. Information to check includes whether or not cache was enabled, hardware checks were blocked, interrupts were blocked, or sanity timer was inhibited.
PSW	Processor Status Word – Refer to Table 4-53 for the layout of this register. The processor mode, in conjunction with the execution level, can often be used to determine which process was active at the time of the initialization. Processor modes are as follows: 00 – Kernel mode 01 – Kernel process 10 – Supervisor process/user process (Release 1 only) 11 – User process
SCR	Store Control Register – Refer to Table 4-55 for the layout of this register. This register is meaningful only if a main store problem is suspected.
ER	Error Register – Refer to Table 4-32 for the layout of this register.
IB	Instruction Buffer – Contains the first half-word of the current instruction at the time of the MRF.
SIR	Store Instruction Register – Contains the remainder of the current instruction or the next instruction.
PPR	Pulse Point Register – Refer to Table 4-54 for the layout of this register.
HSR	Hardware Status Register – Refer to Table 4-36 for the layout of this register.
SDR	Store Data Register – This register is significant only if a main memory problem is suspected.
IMR	Interrupt Mask Register – This register is active low (a 0 indicates that the function is asserted). Therefore, any bit that is 0 allows the corresponding interrupt in the IS register.
IS	Interrupt Source – Refer to Table 4-40 for the layout of the IS register.
UER	Microinterrupt Error Register – Refer to Table 4-51 for the layout of this register.
UER1	Microinterrupt Error Register 1 – Refer to Table 4-52 for the layout of this register.
CDR	CCIO Bus Data Register – This is the channel data register.
TMR	Timer Register – Bits 6 and 7 indicate on-line (6) and off-line (7) sanity timer time-out. If the source is <i>h</i> and no other errors are present, the timer probably indicates a sanity time-out.

Memory Management Exceptions

There are three messages associated with the memory management exceptions or faults.

- SIND – SEGMENT INDEX TOO LARGE
- SINV – SEGMENT INVALID
- PINV – PAGE INVALID.

Each of these messages is preceded by the message REPT ERSLOT, which is documented in 254-303-111, *PDS Output Messages Manual*.

A memory management exception occurs when a virtual address cannot be translated to a corresponding physical address. Such an exception may or may not be a true software error. For example, it is an error if a process does a wild read or write, or takes a wild transfer. It is not an error if a process intentionally uses the memory management exception, or fault, to add or grow a segment.

When a process, intentionally or not, causes a memory management exception, its execution is suspended and its state saved on the interrupt stack. The EIH is then dispatched. The EIH process uses the Microinterrupt Error Register 1 (UER1) to determine the type of exception. Having determined the exception type, EIH clears the UER1 and the IS bit 2, faults the process incurring the exception with the FLT_SIND, FLT_SINV, or FLT_PINV fault code and releases control to the operating system. No error data is collected or logged by EIH on these exceptions, hence there is no REPT CU *n* ERROR INTERRUPT message associated with these exceptions. The data collection and logging (*ERLOG*) of these exceptions is left up to the discretion of the individual processes using available library functions. The processes which consider the memory management exceptions as errors may choose to collect and log such data.

The following example is a typical memory management exception report. Such a message warrants an investigation. Seek technical assistance. The fields UTILITY ID and PROCESS NUMBER identify the process incurring the memory management exception.

30 REPT ERSLOT
PINV - PAGE INVALID X'11
FAULT UNIT : CU 1
TIME OF ERROR: 01/10/70 00:42:43

SAVE STATE
PA: 00000024 PSW: 4ae5fc90 PSBR: fee8c6ef SSBR: fee8c6ef

GENERAL REGISTERS
0: 00000000 1: 00000000 2: 00000000 3: 00000000
4: 00000000 5: 00240000 6: 80329E48 7: 00000000
8: 07B60000 9: 11A01000 A: 51A01000 B: 03F09F43
C: 0000000F D: 0DF520D0 E: 07C307EE F: 00200000
UTILITY ID: 000000C1 PROCESS NUMBER: 00000008
REAL TIME CLOCK: 2FCA1745
01/01/70 12:30:20 #000231

Interpreting Memory Failures

Table 4-13 classifies memory interrupts according to frequency and location and gives guidelines for corrective action.

Table 4-13. Memory Error Interrupts

Frequency	Location	Action
Infrequent (3 per day)	Random	Check log files to ensure that frequency and random locations are confirmed. With the other processor, run the Control Unit Memory (CUMEM) audit to help isolate the problem. (Refer to Chapter 14, "System Audits.")
Infrequent	Same location	Check appropriate log file for occurrences and frequency. With the other processor, run the CUMEM audit to help isolate the problem. (Refer to Chapter 14, "System Audits.") If the problem is repeated at the same location, replace the board. To isolate and confirm bad boards, move suspected board to another location; record the time, date, failure, old location, and new location.
Frequent (3 or more per day)	Random	Check appropriate log file to ensure random locations. If random, the problem may be in the controller or the cables or backplane connection. - For the 3B20D computer, the memory controller is one of the following: UN39 or UN40 if equipped with TN14; UN59, UN59B, or UN59C if equipped with TN28s; UN59B or UN59C if equipped with TN56s without EMM or VLMM features; UN59C if equipped with TN56s with EMM or VLMM features; or UN618 if equipped with TN2012s. - For the 3B21D, the memory controller is one of the following: KLW32, KLW40, KLW48, KLW64, KLW128, or KLW256. To isolate the problem, check connections and cables and run diagnostics. If the problem is not identified, run the supplementary MASC diagnostic phases. - For the 3B20D computer, the phases are 33, 34, 36, 40, 93, 95, and 96. - For the 3B21D computer, the phases are 45, 46, 48, 50, 51, 52, 53, 95, and 96.
Frequent	Same location	Check the log file to verify the location and frequency; if confirmed, replace faulty board.

Responding to Nondiagnosable Memory Errors

Memory-related problems are often transient and cannot be detected by diagnostics. The problem may be in the memory plane itself, the memory controller, or other memory-related circuit packs. Table 4-14 gives information on resolving nondiagnosable memory errors.

⇒ NOTE:

Since Table 4-14 is used for 3B20D Model 1, 2, and 3 computers, refer to Chapter 1, "Hardware Identification," for hardware configurations.

Table 4-14. TLP for Nondiagnosable Memory Errors (3B20D Computer)

Slot Consistent	Error Type	Access Type System (S) Refresh (R) Mixture (M)	Implicated Slot Equipped (E) Unequipped (U)	Suspected Packs (Ranked)
YES	A ¹	S	–	TN14 ² , UN39, UN40, UN34 TN28 ² or TN56 ² , UN59, UN133 TN2012 ² , UN618
		R,M	–	TN14 ³ , UN39, UN40, TN28 ⁴ or TN56 ² , UN59 TN2012 ³ , UN618
	B	S	E ³	TN14, UN39, UN40, TN28 or TN56, UN59 TN2012, UN618
		S	U ³	SW ⁴ , UN39, UN40, SW ⁴ , UN59, SW ⁴ , UN618,
	C	S	–	TN14, UN40, UN39 TN28 or TN56, UN59 TN2012, UN618
	D	R,M	–	TN14, UN39, UN40, TN28 or TN56, UN59 TN2012, UN618
		S	–	TN14, UN39, UN40, UN34 TN28 or TN56, UN59, UN133 TN2012, UN618

See notes at end of table.

Table 4-14. TLP for Nondiagnosable Memory Errors (3B20D Computer) (Contd)

Slot Consistent	Error Type	Access Type System (S) Refresh (R) Mixture (M)	Implicated Slot Equipped (E) Unequipped (U)	Suspected Packs (Ranked)
NO	A ¹	S	–	UN34, UN39, UN40, TN14 UN133, UN59, TN28 or TN56 UN618, TN2012
		R,M	–	UN39, UN40, TN14 ² UN59, TN28 ² or TN56 ² UN618, TN2012 ²
	B	S	E ³	UN39, UN40, TN14 ² UN59, TN28 ² or TN56 ² , UN618, TN2012 ²
		S	U ³	SW ⁴ , UN39, UN40 SW ⁴ , UN59 SW ⁴ , UN618
	C	S	–	UN40, UN39, TN14 ² , UN34 UN59, TN28 ² or TN56 ² , UN133 UN618, TN2012 ²
	D	S	–	UN40, UN39, TN14 ³ , UN34 UN59, TN28 ² or TN56 ² , UN133 UN618, TN2012 ²
		R,M	–	UN39, UN40, TN14 ² UN59, TN28 ² or TN56 ² UN618, TN2012 ²

See notes at end of table.

**Table 4-14. TLP for Nondiagnosable Memory Errors (3B20D Computer)
(Contd)**

Notes:

1. For additional resolution for ERR A failures, determine the source of the ERR A. Some of the more common ERR A sources are as follows:
 - Generated address parity error (bit 6 in ER 1) multiple column select error (bit 8 in ER 1):
 - UN40
 - TN14 | TN28 | TN56 | TN2012
 - UN34 | UN133.
 - Block store go error (bit 22 in ER 1):
 - UN34 | UN133
 - UN40 | UN59, UN618.
 - Busy f/f cleared (bit 4 in ER 1):
 - UN40 | UN59 | UN618.
2. Replace TN14, TN28, TN56, or TN2012 if some consistency is found in the data.
3. Replace TN14, TN28, TN56, or TN2012 last if the failure is an ERR A and the TAR indicates slot 0.

Other possible causes of store errors are missing bus terminating resistors on the memory backplane or loose cables in the store bus.

 - If the memory plane slots indicated in the TAR are equipped (both physically and in the ECD).
 - If the memory plane slots indicated in the TAR are not equipped (both physically and in the ECD).
4. Software errors often cause ERR Bs. Check for an error in one of the following areas:
 - Incorrect parameter in the SGEN1 or SGEN2 ECD form
 - Incorrect parameter in an IOP or disk file controller ECD unit control block option form
 - Software bug.

Store error types listed in Table 4-14 are as follows:

- ERR A *OR* of internal memory hardware check circuits. (If this error occurs in the active CU, a stop-and-switch occurs.)
- Error Register (ER) 1 bits 0 through 11 or 22 — at least one bit set
 - TAR bits 28 and 29 — both reset.
- ERR B out-of-range address.
- ER 1 bits 0 through 11 and 22 — all reset
 - TAR bits 28 and 29 — both reset.
- ERR C uncorrectable data parity of Hamming check error (system access only).
- TAR bit 29 — set.
- ERR D correctable data parity of Hamming check error or uncorrectable refresh error (system access or refresh access).
- TAR bit 28 — set.

Table 4-15 applies to the 3B21D computer. Refer to Chapter 1, "Hardware Identification," for hardware configurations.

Table 4-15. TLP for Nondiagnosable Memory Errors (3B21D Computer)

Slot Consistent	Error Type	Access Type System (S) Refresh (R) Mixture (M)	Implicated Slot Equipped (E) Unequipped (U)	Suspected Packs (Ranked)
YES	A + Channel Error ¹	S	—	DMA Board (KBN15, KBN415), Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256)
	A,D	S,R,M	—	SIMM, Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256)
	B	S	E ²	SIMM, Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256)
		S	U ³	Software ⁴
	C	S	—	SIMM, Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256)
NO	A,D	S,R,M	—	Memory Board (KLW32, KLW40, KLW48, KLW64, or KLW128), SIMM
	B	S	E ²	Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256), SIMM
		S	U ³	Software ⁴
	C	S	—	Memory Board (KLW32, KLW40, KLW48, KLW64, KLW128, KLW256), SIMM

Notes:

1. Channel error register bit 16 = 0.
2. If the memory plane slots indicated are equipped (both physically and in the ECD).
3. If the memory plane slots indicated are not equipped (both physically and in the ECD).
4. Software errors often cause error Bs.

Responding to an Emergency Dump Partition

To use the EMERDUMP feature, an area in low physical memory is loaded with dump parameters to enable the feature. On the next initialization level specified as part of the setup data, the specified area will be dumped. One usage of this feature is the one described — system resource overflow.

Various applications, depending on the requirement, enable the EMERDUMP feature. Contact the appropriate technical group to get the exact procedure. The *UNIX* RTR operating system automatically enables the feature when a system resource overflow condition occurs within the *UNIX* RTR operating system. An emergency dump partition is written with data from the main store. The following output message is printed:

REPT EMERGENCY DUMP PARTITION FULL

Perform the following procedure when an emergency dump partition is written to the disk. Refer to the Input and Output Message Manuals for formats of all the messages used in conjunction with the emergency dump partition.

Step	Procedure
1	Use the OP:EMERSTAT command to print status information concerning the data written to the emergency partition.
2	Use the COPY:TAPE:EMERDMP command to copy the emergency dump file to tape. (Refer to 254-303-110, <i>PDS Input Messages Manual</i> , or 254-303-112, <i>MML Input Messages Manual</i> , for the complete command.) This copy procedure takes about 10 to 15 minutes.
3	After a successful tape copy, use the CLR:EMERDMP command to clear the status flag and make the partition available for future use.
	 NOTE: DO NOT use the CLR:EMERDMP input command to clear the dump unless the office has repeatedly received the REPT EMERGENCY DUMP message and forwarded a tape to the technical assistance organization.
4	Notify your next level of technical support of the emergency dump status and forward the tape and the printout message from the OP:EMERSTAT input command to the application technical assistance organization.

Processor Recovery Message Analysis

An F-level PRM may be issued before each phase OST call. For each step, the ASW PRM (E-level PRM with code DD) contains the postmortem sequence number. Information on how to interpret PRMs is given in 254-303-103, *Processor Recovery Messages Guide*.

Register Layouts

Tables 4-16 through 4-56 contain register layouts for the registers referred to in the error reports located in this chapter. The register layouts provided are shown on the following list. Refer to 254-303-105, *Hardware Reference Manual*.

Table 4-16 Bit and Byte Numbering (All Registers)
Table 4-17 Cache Error Register A (0x0030000) (3B21D Computer)
Table 4-18 Cache Error Register B (0x0030004) (3B21D Computer)
Table 4-19 Cache Error Register C (0x0030008) (3B21D Computer)
Table 4-20 Cache Error Register D (0x003000C) (3B21D Computer)
Table 4-21 Cache Error Register E (0x0030010) (3B21D Computer)
Table 4-22 Cache Tag Trap Register (0x003001C) (3B21D Computer)
Table 4-23 Cache Error Register G (Hit Flip-Flop)(0x0030014) (3B21D Computer)
Table 4-24 Data Path Command Register
Table 4-25 Data Path Error Data Register
Table 4-26 Data Path Status Register
Table 4-27 DMA Channel Status Register — 3B20D Model 1 Computer
Table 4-28 DMAC Status Register — 3B20D Model 1 Computer
Table 4-29 DMAC Status Register — 3B20D Model 2 and 3 Computers
Table 4-30 DMAC Status Register for 3B21D Computer
Table 4-31 DSCH Channel Status Register
Table 4-32 Error Register
Table 4-33 Error A Indicators (3B20D Computer Only)
Table 4-34 Faulty Circuit Packs Indicated by ER 1
Table 4-35 Error A Indicators (3B21D Computer Only)
Table 4-36 Hardware Status Register
Table 4-37 Hardware Status Register 3-Out-of-6 Codes
Table 4-38 BGB Destination Codes
Table 4-39 Hardware Status Register Maintenance States

Table 4-40 Interrupt Source Register
Table 4-41 Main Store Controller Error Register 1 (3B20D Computer)
Table 4-42 Main Store Controller Error Register 2 (3B20D Computer)
Table 4-43 Main Store Controller Trapped Address Register (3B20D Computer)
Table 4-44 3B21D Computer Main Memory Register
Table 4-45 Memory Control Bank Address Boundary Register
Table 4-46 Memory Control Command Register
Table 4-47 Memory Control Error Register
Table 4-48 Memory Control Error Address Register
Table 4-49 Memory Control Status Register
Table 4-50 Memory Control Strobe Shape Register
Table 4-51 Microinterrupt Error Register
Table 4-52 Microinterrupt Error Register 1
Table 4-53 Processor Status Word Register
Table 4-54 Pulse Point Register
Table 4-55 Store Control Register
Table 4-56 System Status Register

Bit and Byte Numbering

Table 4-16. Bit and Byte Numbering (All Registers)

Bit	31 ... 24	23 ... 16	15 ... 8	7 ... 0
Byte	0	1	2	3

Cache Registers (3B21D Computer)

Table 4-17. Cache Error Register A (0x0030000)

Bit	Description
0	Cache Read SM Active
1	Cache Write SM Active
2	Subroutine Stack Read SM Active
3	Subroutine Stack Write SM Active
4	Interrupt Stack Read SM Active
5	Interrupt Stack Write SM Active
6	Hit Invalidate SM Active
7	DMA Invalidate SM Active
8	NOP SM Active
9	Maintenance Read SM Active
10	Maintenance Write SM Active
11	CA Parity Error
12	SA Parity Error
13	Tag with Respect to CA Error
14	Tag with Respect to SA Error
15	Quad Word Alignment Error
16	Command Bus Error
17	Unused

Table 4-18. Cache Error Register B (0x0030004)

Bit	Description
0	Access Request from CC
1	Access Request from Store Interface
2	Write Cycle
3	Read-Modify-Write Cycle
4	Cache Bypass
5	Subroutine Stack Mode
6	Maintenance Access
7	Interrupt Stack Access
8	Address Valid - Translation Hit
9	Byte Access
10	Half-Word Access
11	Processor to Other Store
12	Unused
13	Unused
14	Unused
15	Unused
16	Unused
17	Unused

Table 4-19. Cache Error Register C (0x0030008)

Bit	Description
0	Cache Address - Byte Select
1	Cache Address - Byte Select
2	Cache Address - Entry Select
3	Cache Address - Entry Select
4	Cache Address - Entry Select
5	Cache Address - Entry Select
6	Cache Address - Entry Select
7	Cache Address - Entry Select
8	Cache Address - Entry Select
9	Cache Address - Entry Select
10	Cache Address - Entry Select
11	Cache Address - Entry Select
12	Cache Address - Entry Select
13	Cache Address - Entry Select
14	Cache Address - Entry Select
15	Cache Address - Entry Select
16	Cache Address Parity 3
17	Cache Address Parity 2

Table 4-20. Cache Error Register D (0x003000C)

Bit	Description
0	Cache Address - Tag Compare
1	Cache Address - Tag Compare
2	Cache Address - Tag Compare
3	Cache Address - Tag Compare
4	Cache Address - Tag Compare
5	Cache Address - Tag Compare
6	Cache Address - Tag Compare
7	Cache Address - Tag Compare
8	Cache Address - Tag Compare
9	Cache Address - Tag Compare
10	Cache Address - Tag Compare
11	Cache Address - Tag Compare
12	Cache Address - Tag Compare
13	Cache Address - Tag Compare
14	Cache Address - Tag Compare
15	Cache Address - Tag Compare
16	Cache Address Parity 1
17	Cache Address Parity 0

Table 4-21. Cache Error Register E (0x0030010)

Bit	Description
0	X-Y matchers in ATB matching logic do not match.
1	A zero indicates a multiple hit in SAT.
2	Parity Error over PSW byte 2 received by the ATB (PSW08:15).
3	Parity Error over one of the three SAR bytes received by the SAT.
4	Translation Buffer A Parity Error over the translated address of ATB A.
5	Translation Buffer B Parity Error over the translated address of ATB B.
6	Unused.
7	Unused.
8	Unused.
9	Unused.
10	Unused.
11	Unused.
12	Unused.
13	Unused.
14	Unused.
15	Unused.
16	Unused.
17	Unused.

Table 4-22. Cache Tag Trap Register (0x003001C)

Bit	Description
0	Match Cache Address
1	Match Cache Address
2	Match Cache Address
3	Match Cache Address
4	Match Cache Address
5	Match Cache Address
6	Match Cache Address
7	Match Cache Address
8	Match Cache Address
9	Match Cache Address
10	Match Cache Address
11	Match Cache Address
12	Negative Valid Bit
13	Positive Valid Bit
14	Cache Tag Address Parity 3
15	Cache Tag Address Parity 2
16	Cache Tag Contents Parity 1
17	Cache Tag Contents Parity 0

Table 4-23. Cache Error Register G (Hit Flip-Flop) (0x0030014)

Bit	Description
0	The last read cycle resulted in a cache hit.
1	Unused.
2	Unused.
3	Unused.
4	Unused.
5	Unused.
6	Unused.
7	Unused.
8	Unused.
9	Unused.
10	Unused.
11	Unused.
12	Unused.
13	Unused.
14	Unused.
15	Unused.
16	Unused.
17	Unused.

Data Path Registers

Table 4-24. Data Path Command Register

Bit	Description	Definition
0-1	Write Data Queue Depth	00 = 1 deep 01 = 2 deep 10 = 3 deep 11 = 4 deep
2-3	Read Data Queue Depth	00 = 1 deep 01 = 2 deep 10 = 3 deep 11 = 4 deep
4	Write Data Parity Selection	0 = Even 1 = Odd
5	Write Data Parity Check	0 = Off 1 = On
6	Force Bad Parity System Bus	0 = Off 1 = On
7	Inhibit Error Checking	0 = Check 1 = Inhibit
8-14	Force Check Bit Field on Memory (Bit 15 = 1)	
15	Force Check Bits	0 = Generate 1 = Force
16	Error Check Type	0 = SEC & DED 1 = Parity
17	Correct Single Bit Error	0 = Correct 1 = No Correct
18-19	Force Bad Parity on System Data Bus (Bit 6 = 1)	00 = SD07:SD00 01 = SD15:SD08 10 = SD23:SD16 11 = SD31:SD24
20	Parity Type (Bit 16 = 1)	0 = Even 1 = Odd
21	March Test Data Pair Selection	0 = 0's / 1's 1 = 5's / A's
22-31	RESERVED	

Table 4-25. Data Path Error Data Register

Bit	Bit Description
0-31	Write Data (Parity Error) Bits/ March Test Data Error Bits (latched at time of error.)

Table 4-26. Data Path Status Register

Bit	Description
0	Write Data Parity Error
1	Single Bit Error
2	Multiple Bit Error
3	March Test Error
4-7	RESERVED
8-14	Syndrom / March Test Data Error Bits
15-23	RESERVED
24-30	Check Bits Last Read Access
31	RESERVED

DMA Channel Status Register

Table 4-27. DMA Channel Status Register — 3B20D Model 1 Computer

Bit	Description				Active State
0 – 5	Last addressed device*				
	3/6 Code	Device No.	3/6 Code	Device No.	
	X'07	0	X'25	8	
	X'0B	1	X'26	9	
	X'0D	2	X'29	10	
	X'0E	3	X'2A	11	
	X'15	4	X'31	12	
	X'16	5	X'32	13	
	X'19	6	X'34	14	
	X'1A	7	X'38	15	
6	Error summary bit (set if any error bits are set)				1
7	Channel access error				1
8,9	Last command issued				1
	Bits 8,9	Description			
	00	Setup data was requested from the device			
	01	DMA write to the device			
	10	PIO operation was sent to the channel			
	11	DMA read from the device			
10	DIO acknowledge error				0
11	DIO ready error				1
12	DIO ASW error				0
13	Channel request error				1
14	DIO data parity error				1
15	Busy indicator				0

* The last addressed device is a 3-out-of-6 code which corresponds to a device number as interpreted in this table.

DMAC Status Register

Table 4-28. DMAC Status Register — 3B20D Model 1 Computer

Bit	Description	Active State
0	My store error A	0
1	My store error B (unequipped memory)	0
2	My store error C (bad data)	0
3	Not used	—
4	Not used	—
5	DIO data parity error	0
6	DMAC data parity error	0
7	DMA RAM parity error	1
8	Read-Only Memory (ROM) sequencer check	0
9	DIO ready error	0
10	DIO ASW error	1
11	DIO acknowledge error	1
12	CCIO data parity error	0
13	CCIO command error	0
14	ROM sequencer parity error	0
15	Force error flip-flop	1
16	Not used	—
17	Not used	—
18	Not used	—
19	Not used	—
20	DMA fatal error summary	0
21	Maintenance mode	0
22	CCIO interface busy	0

Table 4-28. DMAC Status Register — 3B20D Model 1 Computer (Contd)

Bit	Description		Active State
23	Select internal RAM address (1) or Register (REG) address (0) *		1/0
24 – 31	RAM address register bit 0 – 7		1/0
	Bits 24 - 31	Register	
	01	DIO data	
	02	Channel 0 channel status	
	04	Buffer	
	05	Channel mask	
	06	Store data	
	07	Incrementer (incremented by 4)	
	42	Channel 1 status	
	B2	Channel 2 status	
	C2	Channel 3 status	

* If bit 23 is a 0, then bits 24-31 select a register.

If bit 23 is a 1, then:

- Bits 24-31 select a DMA RAM location.
- Bits 24 and 25 designate one of four words in the block of RAM for the device.
- Bits 26 and 29 designate one of 16 devices allocated to each channel.
- Bits 30 and 31 designate one of four DMA channels.

Table 4-29. DMAC Status Register — 3B20D Model 2 and 3 Computers

Bit	Description	Active State	
0	MSERA	My store error A	0
1	MSERB	My store error B	0
2	MSERC	My store error C	0
3	SEQTCPL	Sequencer test complete	0
4	CCIOER	CCIO command error	1
5	ROMPF	ROM parity error	0
6	DMDPF3	DMA data parity error – byte 3	0
7	CCIOPF3	CCIO data parity error – byte 3	0
8	DMABUSY	CCIO busy	1
9	CH0BUSY	Relative channel 0 busy	0
10	CH1BUSY	Relative channel 1 busy	0
11	DIOPF	DIO parity failure	1
12	DMADR_B0	RAM/REG address bit 0 echo	0
13	DMADR_B1	RAM/REG address bit 1 echo	0
14	DMDPF2	DMA data parity error – byte 2	0
15	CCIOPF2	CCIO data parity error – byte 2	0
16	DMADR_B2	RAM/REG address bit 2 echo	0
17	DMADR_B3	RAM/REG address bit 3 echo	0
18	DMADR_B4	RAM/REG address bit 4 echo	0
19	DMADR_B5	RAM/REG address bit 5 echo	0
20	DMADR_B6	RAM/REG address bit 6 echo	0
21	DMADR_B8	Select internal RAM address (1) or REG address (0) (Note 1)	1/0
22	DMDPF1	DMA data parity error – byte 1	0
23	CPIOPF1	CCIO data parity error – byte 1	0
24	CHANOP0	Channel opcode bit 0 (Note 2)	1
25	CHANOP1	Channel opcode bit 1 (Note 2)	1
26	DIOASWF	DIO ASW failure	0
27	DIORDYF	DIO ready failure	0
28	DIORECF	DIO receive error	0
29	DIOSNDF	DIO transmit error	0
30	DMDPF0	DMA data parity error – byte 0	0
31	CCIOPF0	CCIO data parity error – byte 0	0

See notes at end of table.

**Table 4-29. DMAC Status Register — 3B20D Model 2 and 3 Computers
(Contd)**

Notes:

1. If bit 21 is a 1 (RAM), then:

- Bits 12-13, 16-19 select a DMA RAM location.
- Bits 12 and 13 designate one of two words in the block of RAM for the device.
- Bits 16-19 designate one of 16 devices allocated to each channel.
- Bit 20 designates one of two DMA channels.

If bit 21 is a 0 (REG), then:

Bits 12–20	Location
01	DIO data
02	Store address
06	Store data
07	Incrementer (incremented by 4)
0D	Clear mask for channel 0
15	Set mask for channel 0
25	Clear mask for channel 1
45	Set mask for channel 1

2. Bits 24 and 25 have no active state but are decoded as follows:

Bits 24 and 25	Location
0	DMA read from the device
1	PIO to the channel was issued
2	DMA write to the device
3	Setup data was requested from the device

**Table 4-29. DMAC Status Register — 3B20D Model 2 and 3 Computers
(Contd)**

3. If there are no error bits set in the DMAC status register, refer to the channel data register for the error.

Channel Data Register	
Bits 0-2	Error Condition
1	Invalid WCA command
2	Illegal device request
3	End of segment error
4	Page access failure
5	DIO channel failure

If the channel data register has a value of 5 (101), look in the 3B20D Model 2 or 3 computer DMAC status register for the DIO error (bits 26, 27, 28, and 29).

Table 4-30. DMAC Status Register for 3B21D Computer

Bit	Description	Active State
0	My store error A	1
1	My store error B	1
2	My store error C	1
3	ROM sequencer check complete	1
4	CCIO command error	1
5	Pipeline register parity error	1
6	Channel data parity error	1
7	Quad word inhibit	1
8	CCIO parity error byte 3	1
9	CCIO parity error byte 2	1
10	CCIO parity error byte 1	1
11	CCIO parity error byte 0	1
12	DMAC data bus parity error byte 3	1
13	DMAC data bus parity error byte 2	1
14	DMAC data bus parity error byte 1	1
15	DMAC data bus parity error byte 0	1
16	Relative channel 0 busy	1
17	Relative channel 1 busy	1
18	Relative channel 2 busy	1
19	Relative channel 3 busy	1
20	RAM/register address bit 0	0/1
21	RAM/register address bit 1	0/1
22	RAM/register address bit 2	0/1
23	RAM/register address bit 3	0/1
24	RAM/register address bit 4	0/1
25	RAM/register address bit 5	0/1
26	Select internal RAM (1) or REG (0) address (Note 1)	0/1
27	Channel all seems well error (Note 2)	1
28	Channel ready error (Note 2)	1
29	Channel status opcode bit 1 (Note 3)	0/1
30	Channel status opcode bit 1 (Note 3)	0/1
31	CCIO Busy	1

See notes at end of table.

Table 4-30. DMAC Status Register for 3B21D Computer (Contd)

Notes:

1. If bit 26 is 1 (RAM), then:
 - Bit 20-23 designates the four-word block of RAM for the device.
 - Bit 24-25 designates one of four channels of the DMA.

If bit 26 is 0 (REG), bits 20-25 are defined in the following table:

Bits 20-25	Location	
0	DMAC Buffer	
1	DMAC DIO data	
2	DMAC Store Address	
5	DMAC Mask	
	Channel Data Register	
	XXXXXX01	Clear Channel 0 Mask
	XXXXXX10	Set Channel 0 Mask
	XXXX01XX	Clear Channel 1 Mask
	XXXX10XX	Set Channel 1 Mask
	XX01XXXX	Clear Channel 2 Mask
	XX10XXXX	Set Channel 2 Mask
	01XXXXXX	Clear Channel 3 Mask
	10XXXXXX	Set Channel 3 Mask
6	DMAC Store Data	
7	DMAC Store Incrementer	

Channel Data Register	
Bits 0-2	Error Condition
1	Invalid PIO command error
2	Illegal store access request
3	End of segment error
4	Page access error
5	Channel error
6	Channel transmit error
7	Channel receive error

2. If the Channel Data Register has a value of 5 (101), look in the 3B21D computer DMAC status register for the DIO errors (bits 27 and 28).

Table 4-30. DMAC Status Register for 3B21D Computer (Contd)

3. Bits 29 and 30 have no active state but are decoded as follows:

Bits 29 and 30	Location
0	DMA read from the device
1	PIO to the channel was issued
2	DMA write to the device
3	Setup data was requested from the device

DSCH Channel Status Register

In Table 4-31 the register is active high. (If bit = 1, the function is asserted.)

Table 4-31. DSCH Channel Status Register

Bit	Description
0	High bits of low return
1	High bits of low return
2	High bits of low return
3	High bits of high return code
4	High bits of high return code
5	High bits of high return code
6	Channel busy flip-flop
7	IO inhibit override flip-flop
8	Sequencer error
9	Illegal 3-out-of-6 device address
10	Command error
11	Channel error (logical OR on previous three status bits)
12	Maintenance flip-flop

Error Register

In Table 4-32 the register is active low. (If bit = 0, function is asserted.) Bit 22 is unused for 3B20D computer Standard Main Memory (SMM) and EMM hardware configurations.

Table 4-32. Error Register

Bit	Error	Class	Active
0	Source bus parity	Stop-and-switch	0
1	Microcontrol parity	Stop-and-switch	0
2	Clock match error†	Stop-and-switch	0
3	IB parity error	Stop-and-switch	0
4*	ATB parity	Stop-and-switch	0
5	Cache error	Stop-and-switch	0
6	MYSERA	Stop-and-switch	0
7*	My store time-out	Stop-and-switch	0
8*	MYSERC	On-line error interrupt	
9	Data Manipulation Unit (DMU) error	Stop-and-switch	0
10	Store Address Controller (SAC) error	Stop-and-switch	0
11	Invalid Maintenance Channel (MCH) error	Off-line error interrupt	0
12	Other store error A	Off-line error interrupt	0
13	Other store refresh parity	Off-line error interrupt	0
14	Other store data parity	Off-line error interrupt	0
15*	Other store time-out	Off-line error interrupt	0
16	Channel error	On-line error interrupt	0
17	IO response error	On-line error interrupt	0
18	IO addressing error	On-line error interrupt	0
19	Parity divert error	On-line error interrupt	0

Refer to footnotes at end of table.

Table 4-32. Error Register (Contd)

Bit	Error	Class	Active
20	My Store Error D (MYSERD)	On-line error interrupt	0
21*	Protection violation	Software error interrupt	0
22*	Virtual Address Out of Range (VORA)	Software error interrupt	0
23*	Out-of-range address (MYSERB)	Software error interrupt	0
24	Out-of-range reference (other store)	Software error interrupt	0
25*	Privileged instruction	Software error interrupt	0
26	Bad alignment on memory reference	Software error interrupt	0
27	Reserved for Expansion slots (Not used in 3B20D computers)		
28	Source bus parity bits	—	0/1
29	Source bus parity bits	—	0/1
30	Source bus parity bits	—	0/1
31	Source bus parity bits	—	0/1

* These error bits are copied into the UER and cleared in the error register on a microinterrupt.

† The clock match error is set by the fault recovery process.

Error Register Description

An analysis of postmortem dump information requires the ability to interpret several hardware registers. The most important of these registers is the error register. This section describes the error register, one bit at a time. Where necessary, additional tables and register descriptions are included. The intent is to provide insight into what the system does when a particular error bit is set. Hopefully, this insight, combined with the observation of system output, will lead to an accurate analysis of the postmortem dump, error messages, and actions taken by the system recovery software.

To fully appreciate this section, you should have a basic understanding of the EIH messages, log files (*ERLOG*, *MEMLOG*, and *PMLOG*), PRMs, and the postmortem dump. These are discussed in earlier sections of this chapter. The error register layout is also provided in Table 4-32.

Error Bit 0 — Source Bus Parity: When this bit is asserted, there is a source-bus-bit-rotate-parity error. The source bus is an internal bus of the computer. It is one of the primary internal buses used in data manipulation and computation. The presence of this error causes an immediate SAS of the currently processing CU to the mate CU. The error is detected by hardware that induces the SAS sequence (there is no microcode involvement). Because of certain recovery actions taken during a system initialization, this bit should be ignored if other bits of the error register (bits 1 and 3 through 27) are asserted.

Error Bit 1 — Microcontrol Parity: This bit is an indicator of bad parity in the Microinstruction Register (MIR). The MIR controls the sequencing circuitry of the Microprogram Unit (MPU) and also is the buffer register for instructions read from the microstore. The presence of this error results in an immediate SAS of the currently processing CU to the mate CU. The error is detected by hardware that induces the SAS sequence (there is no microcode involvement). Because of certain recovery actions taken during a system initialization, this bit should be ignored if other bits of the error register (bits 1 and 3 through 27) are asserted. The following are suggested hardware areas to troubleshoot: WMS, ROMs, SDC, SAT, MCU, SAC, CSU, SREGs, and DMUs.

Error Bit 2 — Clock: Bit 2 is an indicator of mismatches of the auxiliary and primary result clocks of the MPU. As with bits 0 and 1, this bit monitors hardware internal to the MPU and its assertion results in an immediate SAS with no microcode involvement. Because of certain recovery actions taken during a system initialization, this bit should be ignored if other bits of the error register (bits 1 and 3 through 27) are asserted. The following are suggested hardware areas to troubleshoot: WMS, ROMs, MCU, and DMUs.

Error Bit 3 — Instruction Buffer (IB) Parity: This error occurs when the main instruction buffer register (not microinstruction buffer register) contains bad parity. The presence of this error results in an immediate SAS sequence (there is no microcode involvement). The following are suggested hardware areas to troubleshoot: SDC, SAC, UC, CSU, and MASU.

Error Bit 4 — Address Translation Buffer (ATB) Parity: This bit checks parity over the address leads of the ATB. The ATB is a small memory used by the memory management logic in the translation of a virtual address. The error is detected by the microcode which eventually initiates a CU SAS. The following are suggested hardware areas to troubleshoot: SAT, SAC, and SREGs.

Error Bit 5 — Cache Parity: When this bit is asserted, a parity error in the cache is present. The cache is a high-speed memory used to store frequently accessed main memory data. The presence of this error generates an SAS without microcode involvement. The following are suggested hardware areas to troubleshoot: CAC0, CAC1, CAC, SAT, and CAM.

Error Bit 6 — My store Error A (MYSERA): This bit indicates that a hardware error is present in the memory controller. A MYSERA can occur on either a refresh cycle or a system access. It is a “logical OR” of several hardware check circuits of the memory controller. The MYSERA is an SAS error that does not involve the microcode.

3B20D Computer Error Register Description

In addition to error register bit 6, a MYSERA is further indicated by examination of the TAR. The TAR will indicate the following if a MYSERA exists:

- TAR bits 28 and 29 both equal 0.
- TAR bit 27 equals 1 (TAR contents valid).
- TAR contains the memory address at the time MYSERA occurred.

As a further check, memory ER 1 will also indicate that a MYSERA has occurred. The ER 1 and trapped address register bit layouts are described in Tables 4-32 and 4-43, respectively.

Table 4-33 lists the bits in ER 1 that indicate the reason for the MYSERA. This information is provided to assist those well versed in the memory controller hardware in determining the exact cause of the MYSERA.

Often, the contents of ER 1 may be used to locate and replace the faulty circuit pack. Table 4-34 provides a list of circuit packs to replace (in order) when an error is indicated in the ER 1 register. The table makes reference to the UN39 and UN40 circuit packs that comprise the controller. In some models of the 3B20D computer, these packs are replaced by the UN59 and/or UN618 pack. In these cases, substitute UN59 and/or UN618 where either UN40 or UN39 is indicated.

Table 4-33. Error A Indicators (3B20D Computer Only)

ER 1 Bit	Definition
ALoop (00)	The address looparound flip-flop is stuck.
WDPER (01)	A write data parity error occurred. There was bad parity over input data to be written to main memory.
RACMM (02)	A mismatch in the refresh address counters has occurred. There are two refresh address counters (RACNTIX and RACNT2X). They are matched to see if they are in step.
LNAPER (03)	Latch normal address parity error.
BUSYCL (04)	Indicates that the BUSY flip-flop was not cleared. During a memory operation, the BUSY flip-flop gets set and is cleared out at the end of a store cycle.
PFCOMD (05)	Parity failure over the command leads.
GAPER2 (06)	Generated array parity failure. The parity is checked on the array select, row and column.
SLRCNER (07)	Array row and column error. Either the row or column select leads or both were not returned from the array.
SELCM (08)	Multiple column select response.
SELRM (09)	Multiple row select response.
SELMAER (10)	The maintenance ability exists to check the array (row and column) select error circuitry. An error occurred with CHECK00 and MAINT0 set.
RSELN (11)	No row on a refresh cycle.
BKSGER (22)	Block store go error. BKSG0 flip-flop set at TM01 time if SGOIO is active (0) and should remain set until store complete is received for that cycle. If SGOIO is lost before store complete is received, this error will occur.

Table 4-34. Faulty Circuit Packs Indicated by ER 1

ER 1 Bit	Error	Change Circuit Packs in Order, One at a Time*
00	ALOOP	UN39, UN34, UN43, UN45, UN10, UN35
01	WDPER	UN40, UN39, Memory array per TAR register
02	RACHM	UN39
03	LNAPER	UN39, UN34, UN43, UN45, UN10, UN35
04	BUSYCL	UN40, UN39
05	PFCOMD	UN40, UN34, UN45, UN35
06	GAPER2	UN39, Memory array per TAR register
07	SLRCNER	UN39, Memory array per TAR register
08	SELCM	UN39, Memory array per TAR register
09	SELRM	UN39, Memory array per TAR register
10	SELMAER	UN39, UN40, UN34, UN10
11	RSELN	UN39, Memory array per TAR register
22	BKSGER	UN40, UN39, UN34

* In some models of the 3B20D computer:
 UN10 is replaced by the UN616.
 UN11 is replaced by the UN617.
 UN34 is replaced by the UN133.
 UN35 is replaced by the UN46.
 UN39 and UN40 are replaced by the UN59 and/or UN618.
 UN43 is replaced by the UN611.
 UN45 is replaced by the UN612.

3B21D Computer Error Register Description

In addition to error register bit 6, a MYSERA is further indicated by examination of the following registers:

- Main Memory Register (MMR).
- Memory Control Error Register.
- Data Path (DP) Status Register.

Table 4-35 indicates the bits that are set in these registers on a MYSERA. This information is provided to assist those well versed in the memory controller hardware to determine the exact cause of the MYSERA. The table also gives the definitions of the various errors.

Often, this error indicates a faulty memory control or a faulty memory board (KLW32, KLW40, KLW48, KLW64, or KLW128). Except for “WDPER” errors, replace the main memory circuit pack and the SIMM on the memory board corresponding to the memory bank reported in the error message.

Table 4-35. Error A Indicators (3B21D Computer Only)

Error (Bit)	Definition
ALOOP (MMR 16)	The address looparound flip-flop is stuck.
WDPER (MMR 20) (DP 0)	A write data parity error occurred. There were bad parity over input data to be written to main memory.
LNAPER (MMR 19)	Latch normal address parity error.
PFCOMD (MMR 04)	Parity failure over the command leads.
BKSGER (MMR 13)	Block store go error.
REFAB0 (MMR 21) (Mem Ctl 1)	General error which originates from the memory control. This indicates a refresh abort error.
BANKER (MMR 21) (Mem Ctl 2)	General error which originates from the memory control. This indicates a bad bank address.
STAMER (MMR 21) (Mem Ctl 3)	General error which originates from the memory control. This indicates a state computer failure.
MTSTER (MMR 21) (Mem Ctl 0) (DP 3)	General error which originates from the memory control. This indicates a march test failure.

Error Bit 7 — My Store Time-Out (MS Time-Out): This bit indicates whether or not the memory controller has responded to a memory request within a period of time determined by the microcode. When the microcode detects an MS time-out, it initiates an SAS of the currently running CU. The contents of the error register are also copied to the Microinterrupt Error Register (UER) when this error occurs. The following are suggested hardware areas to troubleshoot: SAC, MASU, MASC, and SREGs.

Error Bit 8 — My Store Error C (MYSERC): When this bit is asserted, a noncorrectable parity error has occurred as the currently running CU attempted to read a location in memory. For a MYSERC, the following conditions are true:

- Bit 8 of the error register is asserted.
- Two or more bits of the addressed location are in error. This implies that the Hamming circuits could not correct the error. It also means that even though the system may remain running, the data that was read could be incorrect (if the system is not in the ACT/STBY state). This may lead to further problems, depending on how critical the data is to the system.
- This error is detected by the microcode which suspends the process encountering the error. It also dispatches the EIH to clean up the error and notify the suspended process of the bad data it has received (if the data could not be corrected).



NOTE:

If the process that was running when the error occurred either is a level 15 kernel process or has interrupts blocked, microcode has no recourse but to initiate a system initialization.

- There are two situations under which a MYSERC can occur:
 - The system is ACT/STBY – Here, even though the error is not correctable, the memory of the standby CU does contain valid data. The memory of the other CU is read to get the good data which in turn is rewritten in the active CU, correcting the bad parity. Once corrected, the suspended process is allowed to continue as though the error never occurred.
 - The system is ACT/OOS – In this case, there is no up-to-date memory with the correct data. The suspended process will be allowed to continue execution in its fault entry. It is expected that the process will take steps to reinitialize whatever operation was in progress when the system continues with no further problems. If this is not the case, the bad data may lead to repeated system initializations until the problem is corrected.

- In all cases (correctable and noncorrectable), system error counters are incremented. If a sufficient number of these errors occur in a specified time interval, the configuration management software will attempt recovery. The recovery may include switches to the out-of-service CU and varied levels of system initialization.
- The TAR indicates that bits 27 and 29 are asserted. The TAR will also contain the address of the location in error that can be used to identify the faulty circuit pack.

For this error, as well as other memory related problems, refer to the “Memory History Log File” section. On memory parity related problems, it is often important that the faulty circuit pack (and address on the pack) be identified from the data presented in the *MEMLOG*. Such information may then be used to locate a malfunctioning device when the error is a soft error (transient). It is possible such transient errors may not be detected by routine diagnostics.

Error Bit 9 — Data Manipulation Unit (DMU): This error results in an SAS. It is detected by low-level hardware circuits and does not involve the microcode in the decision to SAS. The only recourse is to run diagnostics on the faulty CU once the system has recovered from the error. The following are suggested hardware areas to troubleshoot: SDC, DMU0, and DMU1.

Error Bit 10 — My Store Address Controller (SAC): This error results in an SAS. It is detected by low-level hardware circuits and does not involve the microcode in the decision to SAS. The only recourse is to run diagnostics on the faulty CU once the system has recovered from the error. Troubleshoot the SAC circuit pack.

Error Bit 11 — Maintenance Channel Order (MCH Order): This error can occur under two circumstances:

- When the mate CU tries to communicate across the maintenance channel to the active CU. Such communication is not allowed. A CU that has its CC bit set (Table 4-56) will reject most commands across the maintenance channel. The CC bit is used to determine which CU has the right to execute as the active processor. A CU that is not active in most instances will be stopped and will not send MCH orders to the active CU.
- When the mate CU attempts to SAS to the active CU. When the active CU's maintenance channel receives the SAS request, it will not honor it but will instead set bit 11 of the error register. An SAS can be initiated by either hardware circuits or 3B computer microcode. In most instances, however, the hardware errors are inhibited in the mate CU and the mate is stopped.

The thing to notice about both of these situations, is that in order for this error to be detected, the system must be in the ACT/STBY state. Only under this condition will the error be noticed by the system. The MCH order error is handled by the EIH when it is attached to bit 1 of the Interrupt Source (IS)

register (other CC error). The EIH is attached to IS bit 1 only when the mate CU is in the standby state. Look for another source of the problem. This is not an SAS problem.

Error Bit 12 — Other Store Error A (OTHSERA): An OTHSERA is in principle the same error as the my-store-error A, except that an OTHSERA occurs in a standby CU. (Also refer to the description of error bit 6.) For the system to detect this error, the EIH process must be attached to IS bit 1. This is the case only when the system is in the ACT/STBY configuration.

Because the error is in the standby CU, the error handling and recovery actions are not as severe as for the MYSERA. When the error occurs, the currently executing process is suspended and the EIH is dispatched.

⇒ NOTE:

The EIH may not be dispatched immediately if the currently executing process has interrupts blocked or is a level 15 process. In such cases, the EIH will be dispatched as soon as the process level drops below 15 and interrupts are enabled.

The EIH will attempt to clean up the error in the standby CU and keep the memories of both CUs synchronized. A system error counter will be incremented and data about the error collected and eventually spooled to the system printer and *MEMLOG*. Once the error is cleared, control will return to the process suspended when the EIH was dispatched. This error is transparent to all processes in the system except those directly involved in handling the error.

If a series of OTHSERA errors occur in rapid succession and exceed the OTHSERA error threshold, the configuration management software will take steps to remove the standby CU from service. Once the standby is removed, further errors in that CU will be disregarded because the EIH will have detached from other CU interrupts. Upon removal of the standby CU, automatic diagnostics will be requested to determine the faulty hardware.

For further details on the analysis of this error, refer to the description of error bit 6 and the “Memory History Log File” section.

Error Bit 13 — Other Store Error D (OTHSERD): This error is similar to the MYSERD error (error register bit 20). The major difference is that an OTHSERD occurs in the standby CU while a MYSERD occurs in the active CU.

Assuming that interrupts are not blocked and the currently executing process is below level 15, the EIH will be dispatched to clear up the error. If this is not the case, the EIH will be dispatched once interrupts are enabled and the process level drops below 15.

The EIH will attempt to clean up the error in the standby CU by restoring good parity to the faulty location. A system error counter will be incremented and data about the error collected and eventually spooled to the system printer and *MEMLOG*. Once the error is cleared, control will return to the process suspended when the EIH was dispatched. This error is transparent to all processes in the system except those directly involved in handling the error.

If a series of OTHSERD errors occurs in rapid succession and exceeds the OTHSERD error threshold, the configuration management software will take steps to remove the standby CU from service. Once the standby is removed, further errors in that CU will be disregarded because the EIH will have detached from other CU interrupts. Upon removal of the standby CU, automatic diagnostics will be requested to determine the faulty hardware.

For further details on the analysis of this error, refer to the description of error bit 20 and the "Memory History Log File" section.

Error Bit 14 — Other Store Error C (OTHSERC): This error is similar to the MYSERC error (error register bit 8). The major difference is that an OTHSERC occurs in the standby CU while a MYSERC occurs in the active CU. Unlike a MYSERC, an OTHSERC does not present the possibility of a process getting bad data. The error is in the mate CU's memory. The on-line CU's memory is not affected.

Assuming that interrupts are not blocked and the currently executing process is below level 15, the EIH will be dispatched to clear up the error. If this is not the case, the EIH will be dispatched once interrupts are enabled and the process level drops below 15.

The EIH will attempt to clean up the error in the standby CU by restoring good parity to the faulty location. A system error counter will be incremented and data about the error collected and eventually spooled to the system printer and *MEMLOG*. Once the error is cleared, control will return to the process suspended when the EIH was dispatched. This error is transparent to all processes in the system except those directly involved in handling the error.

If a series of OTHSERC errors occur in rapid succession and exceed the OTHSERC error threshold, the configuration management software will take steps to remove the standby CU from service. Once the standby is removed, further errors in that CU will be disregarded because the EIH will have detached from other CU interrupts. Upon removal of the standby CU, automatic diagnostics will be requested to determine the faulty hardware.

For further details on the analysis of this error, refer to the description of error bit 8 and the "Memory History Log File" section.

Error Bit 15 — Other Store (OS) Time-Out: An OS time-out is similar to an MS time-out in that the memory controller does not respond within a specified time interval on a memory access. Because an OS time-out is in the standby CU, the

error is transparent to all system processes except those directly involved in clearing and reporting the error. This assumes that interrupts are enabled and the currently executing process is level 14 or lower.

The OS time-out is detected by the microcode on a memory access. When the system is in the ACT/STBY configuration, data is read and written to both CUs' memories. The microcode checks the completion response from both CUs. Failure to get a successful response from either CU will generate an MS or OS time-out.

On the occurrence of this error, microcode will attempt to transfer control to the EIH so that the necessary error recovery can take place. Transfer of control to the EIH is accomplished when microcode sets interrupt source register bit 1 (OTHCC). By setting this bit, the EIH will be dispatched to its other CU error handling routine. This is possible only if interrupts are not blocked and the currently executing process is level 14 or lower.

If the EIH cannot be dispatched, microcode will reset the system (MRF) resulting in a UNIX RTR operating system level 1 phase. Assuming that the EIH can be dispatched, the standby CU will be removed by decision of the configuration management software. If the EIH could not be dispatched and a system reset takes place, the standby CU will be removed from service by the EIH when it gets control at the start of the initialization.

Because of the nature of the error, the *MEMLOG* and ROP messages for this error are not very informative. The error is usually detectable by automatic diagnostics, requested after the error occurs.

Error Bit 16 — Channel Error: This bit is a logical "OR" of the error leads from all channels. When the channel error bit is asserted in the error register, interrupt source register bit 0 is set as well. Assuming that the currently executing process is level 14 or lower and interrupts are not blocked, the EIH will be dispatched at its on-line hardware error interrupt entry. If this is not the case, the dispatch of the EIH will not occur until the process level drops below 15 and interrupts are enabled.

Unlike an OS time-out error or other microcode detected errors, the system will not phase if the EIH cannot be immediately dispatched. Channel errors do not involve microcode in the dispatch of the EIH.

The EIH does not immediately know the type of channel error that occurred when it gains control. It must poll all the channels (*pioe* instruction) to determine which channel has asserted the channel error bit. In the case of multiple channels asserting the channel error bit, the channel error with the lowest number will be handled first.

Once the channel is identified, the EIH must determine the type from the ECD and its placement and function in the computer. This is necessary because the error cleanup varies depending on the channel type (SCH, DSCH, or other) and function (DMA channel, CCIO channel, or other).

Regardless of the channel type or function, there is a common set of actions that are performed for each channel.

1. Get the status of the channel.
2. Classify the error using status information from the channel hardware.
3. Collect any additional relevant data, such as DMAC internal RAM and registers.
4. Initialize the channel. Clear the error and attempt to put the channel back into an operational state.
5. Report the error to the configuration management software.
6. If the error is because of a software action, the responsible process has had its execution suspended. The suspended process will resume execution at its faulty entry to retry or abort the action it was attempting.
7. If the error is because of a hardware or a device action, the driver process responsible for maintaining the affected hardware will be notified via its fault entry that an error has occurred.

These actions are taken for all channel errors. The channel error recovery as seen by the system fits into one of two major categories:

- Fatal DMA related errors – Errors that are because the DMA hardware itself causes the DMA sequencer to jump to a state which requires extensive initialization of all DMA channels and the DMA itself. When a fatal error is indicated, the EIH will request a system reset. This will provide the level of initialization required to clear the DMA and initialize the system driver processes. Examples of such errors are as follows:
 - Parity errors of the DMA RAM, ROM data, and CCIO bus
 - Main memory noncorrectable parity errors detected by the DMA
 - Out-of-range main memory reference by the DMA
 - MYSERA detected by the DMA
 - A second command sent to the DMA before the first is allowed to complete.
- Nonfatal errors – There are two subclasses of nonfatal errors:
 - Errors that do not involve the DMA
 - Errors that involve a DMA channel but allow the DMA to be restarted without impact to other DMA channels.

Both of the preceding situations allow the error condition to be cleared with minimal impact to the system. All that is required is to clear the error, restart the DMA, and notify the driver process responsible for maintaining the affected hardware. A system initialization is not required to recover from a nonfatal error. Examples of nonfatal channel errors are as follows:

- Acknowledge (ACK) errors
- Ready (RDY) errors
- DSCH data parity
- ASW
- Channel addressing errors.

An analysis of a channel error related problem varies, depending on whether the error is because of bad hardware or a deficiency in software. The error message contents (and log files) provide data specific to the problem. Determining the exact cause of an error often requires more information than that presented by a single-error message.

The “On-Line Hardware Faults” section provides examples of channel related problems. For channel errors, it is best to collect as much ROP and log file information as possible and observe in entirety the logical sequence of system degradation and recovery actions. The tables in this section contain the layouts of some of the registers printed in the error messages. These may provide an explanation for a particular error. Errors of this type need the attention of application support personnel. In any case, the CU where the error occurs should be diagnosed to rule out the possibility of faulty hardware.

Error Bit 17 — IO Response: This error is detected on the leading edge of a pulse point and indicates that a 3-out-of-6 (3/6) acknowledgment of a pulse point is active. At this point, there should be no activity and the response is deemed to be invalid. The only error information collected is the hardware status register. The running process is faulted with a FLT_NDREP faulty code.

If a series of IO response errors occur in rapid succession and exceed the error threshold, configuration management software will take steps to remove the active CU from service. There are two types of removal actions possible:

- Soft switch – If the system is the ACT/STBY configuration, the CU switch will be transparent to most processes. The SIM will be notified of the switch and in turn will notify the Application Integrity Monitor (AIM). This notification allows application software that is CU dependent the chance to initialize. Drivers in general are not affected by the switch because any IO activity at the time of the switch is carried over to the new CU.
- Hard switch – If the mate CU is not in the standby state, a transparent (soft switch) is not possible because the mate CU's memory is not up to date. A hard switch is more severe than a soft switch in that it requires a *UNIX* RTR operating system level 1 initialization. To accomplish this, the

CU hardware is set up to allow the mate CU to gain control and the on-line CU to execute a system reset (MRF). On reset, the mate CU will gain control of the system. To the system (other than the fault recovery processes), a hard switch is identical to an SAS.

Error Bit 18 — IO Address/Pulse Point (IO addr/pp): This error is detected on the trailing edge of the pulse point and indicates either an illegal 3-out-of-6 (3/6) code on the CCIO bus or that multiple pulse points are active. This error is because of a channel holding the CCIO bus, a problem with the CC hardware that accesses the CCIO bus, or software insanity that results in an invalid 3/6 address being used. The only data that is collected on this error is the hardware status register. The running process is faulted with a FLT_NDREP fault code.

If a series of IO address errors occur in rapid succession and exceed the error threshold, configuration management software will take steps to remove the active CU from service. There are two types of removal actions possible:

- Soft switch – If the system is in the ACT/STBY configuration, the CU switch will be transparent to most processes. The SIM will be notified of the switch and in turn will notify the AIM. This notification allows application software that is CU dependent the chance to initialize. Drivers in general are not affected by the switch because any IO activity at the time of the switch is carried over to the new CU.
- Hard switch – If the mate CU is not in the standby state, a transparent (soft switch) is not possible because the mate CU's memory is not up to date. A hard switch is more severe than a soft switch in that it requires a UNIX RTR operating system level 1 initialization. To accomplish this, the CU hardware is set up to allow the mate CU to gain control and the on-line CU to execute a system reset (MRF). On reset, the mate CU will gain control of the system. To the system (other than the fault recovery processes), a hard switch is identical to an SAS.

Error Bit 19 — Parity Divert (pardiv): The normal mode of channel operation assumes the channels respond with good parity. Where bad parity is expected, parity checks can be inhibited. When a channel response returns bad parity and good parity was expected, the result is a parity divert error. The parity divert error usually indicates a device problem since the channel does not regenerate parity over the data it receives. This error can occur only during programmed input/output; hence, the running process is faulted with a FLT_NDREP fault code.

If a series of parity divert errors occur in rapid succession and exceed the error threshold, configuration management software will take steps to remove the active CU from service. There are two types of removal actions possible:

- Soft switch – If the system is in the ACT/STBY configuration, the CU switch will be transparent to most processes. The SIM will be notified of the switch and in turn will notify the AIM. This notification allows application software that is CU dependent the chance to initialize. Drivers

in general are not affected by the switch because any IO activity at the time of the switch is carried over to the new CU.

- Hard switch – If the mate CU is not in the standby state, a transparent (soft switch) is not possible because the mate CU's memory is not up to date. A hard switch is more severe than a soft switch in that it requires a *UNIX* RTR operating system level 1 initialization. To accomplish this, the CU hardware is set up to allow the mate CU to gain control and the on-line CU to execute a system reset (MRF). On reset, the mate CU will gain control of the system. To the system (other than the fault recovery processes), a hard switch is identical to an SAS.

Error Bit 20 — My Store Error D (MYSERD): This bit indicates a correctable (single-bit) error has occurred on a memory read or refresh cycle. When a MYSERD occurs, the following conditions exist:

- Bit 20 of the error register is asserted.
- The TAR has the address of the faulty location.
- A system error counter is incremented.
- The memory has invalid data, but any read of the location will guarantee good data because of the Hamming circuitry.
- No process needs to be faulted because the error is correctable.
- An on-line hardware error interrupt is generated. The MYSERD is detected by hardware and sets IS bit 1. If the currently executing process is level 14 or lower and interrupts are not blocked, the EIH will be dispatched to correct the bad parity. If this is not the case, the EIH will be dispatched when the execution level drops below 15 and interrupts are enabled.

If a sufficient number of MYSERD errors occur and exceed the allowable limit, the configuration management software will take steps to switch to the mate CU. If the mate CU is in the standby state, a soft switch will be attempted and the faulty CU will be removed from service and diagnosed. If the state of the mate CU is OOS, a hard switch will be attempted.

There is a limp mode of operation for MYSERD errors. If a CU switch is not possible or both CUs are experiencing MYSERD errors on a sufficiently frequent basis, the EIH will always yield good data because of the Hamming circuitry. The ability to disable MYSERD detection is a limp mode feature that is used to prevent a situation where a frequently accessed location continuously generates single-bit parity error interrupts.

For this error, as well as other memory related problems, refer to the "Memory History Log File" section. On memory parity related problems, it is often important that the faulty circuit pack (and address on the pack) be identified from the data presented in the *MEMLOG*. Such information may then be used to locate a malfunctioning device when the error is a soft error (transient). It is possible such transient errors may not be detected by routine diagnostics.

Error Bit 21 — Protection Violation: This error indicates that a process attempted to access a segment in which access was not allowed. A good example would be an attempt to write into a text segment. The error is detected by microcode which then dispatches the EIH by asserting IS bit 2 (on-line software error).

⇒ **NOTE:**

If the process that was running when the error occurred is either a level 15 kernel process or has interrupts blocked, microcode has no recourse but to initiate a system initialization.

The error message and *ERLOG* log file will contain the process saved state. The saved state data includes the general registers, Program Address (PA), Segment Base Register (SBR), Processor Status Word (PSW), Utility ID (UTILID), and process number. This is generally sufficient information to identify both the process and the exact place in the process's text where the error occurred.

A system error counter is incremented on each instance of this error. If a sufficient number of errors occur within a specified time interval, the configuration management software will attempt recovery. The recovery may include CU switches and various levels of initialization.

Error Bit 22 — Virtual Address Out of Range (VORA): The VORA error is detected by microcode and it can occur only in the Release 6 (VLMM) generic. The VLMM system can address up to 64 megabytes of memory. Therefore, the valid address values are limited to bits (25 - 0). If any of the remaining high order bits (31 - 26) are set during a memory fetch, the VLMM microcode will generate an interrupt and the Error Interrupt Handler (EIH) will be dispatched. As a result, the message

REPT VIRTUAL ADDRESS OUT OF RANGE

with the interrupt state of the process incurring the error will be printed on the ROP and logged in the *ERLOG* file, unless the CU is inhibited. EIH will then fault the offending process with the FLT_ADDR fault code.

This error can be caused by any *UNIX* RTR operating system or application process and is treated by EIH as a software error. Occasionally, faulty hardware can cause this error and implicate a process that happened to be running at the time.

The error message, or *ERLOG* log file, contains all the information needed to determine the problem. The interrupt save state indicated the address of the instruction which caused the VORA error.

Error Bit 23 — My Store Error B (MYSERB): A MYSERB is an indication that a process has attempted to access a location in memory that is not physically equipped. The most likely reason for an error of this type is mutilation of system segment and page tables. However, incorrect memory equipage or bad hardware can also cause this error. In most cases, incorrect memory equipage problems are detected before the system ever achieves kernel processing capability on initialization.

The preceding assumption is that the error is associated with a virtual memory access. It is also possible, however, for this error to occur on a physical memory access as well. This usually is because a process uses a bad address or pointer. Regardless of whether the error occurs on a virtual or physical access, the offending process is faulted with a `FLT_MYB` fault code.

If the error is because of a mutilated page or segment tables, the error signals the presence of very serious software problems. To rule out the possibility of memory equipage or memory hardware problems, the system should be switched to the mate CU if this does not take place automatically. The CU where the error occurred should be diagnosed and memory equipage checked against what is indicated in the ECD.

If after the CU switch, the problem still exists and the memory equipage is correct, the problem is with the segment and page tables or a process doing a physical write with a bad address. Such a problem should be referred to qualified application support personnel. Error messages ROP and *ERLOG* should be collected for analysis. The MYSERB error message provides the saved state of the process involved in the error. This may be of use if the problem is an errant physical write.

A MYSERB is detected by the microcode which then attempts to dispatch the EIH. If a level 15 process is involved or interrupts are blocked, microcode is unable to dispatch the EIH and initiates a system reset (MRF).

Error Bit 24 — Other Store Error B (OTHSERB): This bit indicates that there is a problem with the off-line memory equipage, the memory hardware, or a process doing off-line physical memory access using a bad address. An OTHSERB can occur only when the system is in the ACT/STBY configuration. The implication of a large number of OTHSERBs is that the configuration management software will eventually request removal of the standby CU, after which diagnostics will be run.

If diagnostics do not locate a problem with a faulty CU, the memory physical equipage would be checked against the equipage shown in the ECD. The equipage of the two CUs does not have to necessarily be the same; however, the equipage given in the ECD should agree with what is physically present. If

the equipage is correct and diagnostics do not locate a problem, the error message contains the saved state of the process. The saved state data includes the PA, PSW, PSBR, UTILID, process number, and the general registers. Saved state data will identify the process and the location in the process's text where the error occurred.

An OTHSERB is a software interrupt that will be detected only if the executing process is level 15 or lower and interrupts are not blocked. If the error is not immediately detected, succeeding memory references (opcode fetches) will clear the error condition. Even though the EIH cannot be dispatched (blocked interrupts), this error will not result in a system reset. The currently executing process will not be faulted since the cause of the error is not clear.

Error Bit 25 — Instruction Privilege Violation: A privilege violation indicates that a process attempted to execute an instruction and did not have appropriate PSW privileges. Good examples of this are the HALT instruction, IO instructions, and write PSW. The error is detected by microcode which then attempts to dispatch the EIH by setting IS bit 2.

⇒ NOTE:

If the process that was running when the error occurred is either a level 15 kernel process or has interrupts blocked, microcode has no recourse but to initiate a system initialization.

On a protection violation the following conditions are true:

- Bit 25 in the error register is set.
- A system error was incremented and may lead to recovery actions.
- The error generates a microinterrupt.
- The guilty process will be faulted with a FLT_PRIV fault code.

The error message and *ERLOG* log file will contain the process saved state. The saved state data includes the general registers, PA, PSBR, PSW, UTILID, and process number. This is generally sufficient information to identify both the process and the exact place in the process's text where the error occurred.

A system error counter is incremented on each instance of this error. If a sufficient number of errors occur within a specified time interval, the configuration management software will attempt recovery. The recovery may include CU switches and various levels of initialization.

Error Bit 26 — Address Violation: This bit indicates that a process attempted to access a location in memory using a byte address. All computer instructions that access memory are restricted to a full-word boundary. When this error occurs, the EIH is dispatched via the software error interrupt (IS bit 2). The process is faulted with a FLT_ADDR fault code.

If interrupts are blocked or the currently executing process is a level 15 process, the EIH will not be dispatched. Because this error does not generate a microinterrupt, the inability to dispatch the EIH will not cause a system reset (MRF). The error in the error register will be cleaned up on the next memory reference (example, opcode fetch).

The error message and *ERLOG* log file will contain the process saved state. The saved state data includes the general registers, PA, PSBR, PSW, UTILID, and process number. This is generally sufficient information to identify both the process and the exact place in the process's text where the error occurred.

A system error counter is incremented on each instance of this error. If a sufficient number of errors occur within a specified time interval, the configuration management software will attempt recovery. The recovery may include CU switches and various levels of initialization.

Hardware Status Registers

Table 4-36. Hardware Status Register

Bit	Description
0	Carry temporary flag
1	Negative temporary flag
2	Overflow temporary flag
3	Zero temporary flag
4	Channel ASW signal (= 1 if IO operation terminated because of an error)
5	Channel maintenance response (=1 if maintenance response is received from device)
6	Error in checksum (=1 if error)
7	Channel ready (=1 if channel is ready)
8	MCH backup - used to disable and MRF the processor in the event that MCH has failed. Can be set by other CC pulse point 14.
9	MCH backup - used to disable and MRF the processor in the event that MCH has failed. Can be set by other CC pulse point 15.
10 *	3-out-of-6 code main channel address for CCIO bus address leads
11 *	3-out-of-6 code main channel address for CCIO bus address leads
12 *	3-out-of-6 code main channel address for CCIO bus address leads
13 *	3-out-of-6 code main channel address for CCIO bus address leads
14 *	3-out-of-6 code main channel address for CCIO bus address leads
15 *	3-out-of-6 code main channel address for CCIO bus address leads
16	Bidirectional Gating Bus (BGB) direction (1 = to the BGR, 0 = from the BGR)
17	Choice made by BGB multiplexer
18	Choice made by BGB multiplexer
19	Choice made by BGB multiplexer
20 †	Selection of BGB destination
21 †	Selection of BGB destination
22 †	Selection of BGB destination
23	BGB enable

Refer to footnotes at end of table.

Table 4-36. Hardware Status Register (Contd)

Bit	Description
24	Not used
25	Not used
26 ‡	Control of maintenance states in CC
27 ‡	Control of maintenance states in CC
28 ‡	Control of maintenance states in CC
29 ‡	Control of maintenance states in CC
30 ‡	Control of maintenance states in CC
31 ‡	Microinterrupt control (0 = normal, 1 = maintenance state)

* Refer to Table 4-37.

† Refer to Table 4-38.

‡ Refer to Table 4-39.

Table 4-37. Hardware Status Register 3-Out-of-6 Codes

3/6 Code	Channel
07	0
0B	1
0D	2
0E	3
13	4
15	5
16	6
19	7
1A	8
1C	9
23	10
25	11
26	12
29	13
2A	14
2C	15
31	16
32	17
34	18
38	19

Table 4-38. BGB Destination Codes

Bits 20, 21, and 22	Destination
000	WMS – Writable Microstore
001	MLTS – Micro Level Test Set
010	MCH – Maintenance Channel
011	UC – Utility Circuit
100	SID – Self Identification

Hardware Status Register Maintenance States

Table 4-39. Hardware Status Register Maintenance States

30	HSR Bits*				Function
	29	28	27	26	
0	X	X	X	X	No maintenance state active
1	0	0	0	X	RTC timing chain (increment PPR25, reset PPR24)
1	0	0	1	X	3-out-of-6 CCIO response check
1	0	1	0	X	Force bad parity: store control signals
1	0	1	1	X	Force bad parity DST32-35
1	1	0	0	X	Block parity: ALU clock
1	1	0	1	X	Force ATB check circuitry mismatch
1	1	1	0	X	Update circuit clock disable
1	1	1	1	X	EAI maintenance state
X	X	0	0	1	Not used
X	X	0	1	1	ATB memory byte 0 read (Release 6 or later)
X	X	1	0	1	Clear bit rotate error latch
X	X	1	1	1	Clear SRC parity error latch

* X = do not care.

Interrupt Source Register

In Table 4-40 the register is active low. (If bit = 0, the function is asserted.)

Table 4-40. Interrupt Source Register

Bit	Meaning
0	On-line hardware error interrupt
1	Other CC error interrupt
2	Software error interrupt
3	Not connected for the 3B20D computer (reserved for Expansion slot in the 3B21D computer)
4	5-millisecond timer
5	10-millisecond timer for the 3B20D computer (reserved for Expansion slot in the 3B21D computer)
6	1A processor simulation for the 3B20D computer (reserved for Expansion slot in the 3B21D computer)
7	Flash interrupt (Release 6.4 and later, not connected) (reserved for Expansion slot in the 3B21D computer)
8	Utility circuit interrupt
9	Stop the world interrupt
10	DMA channel interrupts
11	DMA channel interrupts
12	Reserved for future use (not connected for the 3B20D computer)
13	Reserved for future use (not connected for the 3B20D computer)
14	Reserved for future use (not connected for the 3B20D computer)
15	Reserved for future use (not connected for the 3B20D computer)
16	EAI Interrupt for the 3B21D computer (not connected for the 3B20D computer)
17	Programmed Interrupt Request (PIR) 15
18	PIR 14
19	PIR 13
20	PIR 12
21	PIR 11
22	PIR 10
23	PIR 9
24	PIR 8
25	PIR 7
26	PIR 6
27	PIR 5
28	PIR 4
29	PIR 3
30	PIR 2
31	PIR 1

Main Store Controller Error Register 1 (3B20D Computer)

Table 4-41. Main Store Controller Error Register 1 (3B20D Computer)

Bit	Name	Definition	Active
0	ALoop	Address looparound (err A)	1
1	WDPER	Write data parity error (err A)	1
2	RACMM	Refresh address counter mismatch (err A)	1
3	LNAPER	Latched address parity error (err A)	1
4	BSYCLR	Busy F/F cleared (err A)	1
5	PFCOMD	Parity failed over commands (err A)	1
6	GAPER21	Generated address parity error (err A)	1
7	SLRCNER	Select row column error (err A)	1
8	SELCM	Multiple column select response (err A)	1
9	SELRM	Multiple row select response (err A)	1
10	SELMAER	Select response on maintenance operation (err A)	1
11	RSELN	No row response on refresh cycle (err A)	1
12	NOSHF	Maintenance order. Disable clock to sequencer	1
13	NOREF	Maintenance state. Stop refresh	1
14	BLKERD	Maintenance state. Block error D	1
15	CHECK	Maintenance state. Check array select responses	1
16	REQREF	Maintenance order. Request refresh	1
17	STRAC	Maintenance order. Step refresh counter	1
18	STSF	Maintenance order. Single step sequencer	1
19	STREFC	Maintenance order. Step refresh timer	1
20	SETBSY	Maintenance order. Set busy F/F	1
21	EALoop	Maintenance order. End address looparound	1
22	BKSGER	Block store go error (err A)	1
23	CLERA	Maintenance order. Clear error A	1
24	CLERD	Maintenance order. Clear error D	1
25	IREF	Maintenance order. Zero refresh address counters	1
26	RCNTLD	Maintenance order. Load refresh rate	1
27	LATCH	Maintenance order. Latch state of data bus	1
28	BYTE3	Decoded byte select	0
29	BYTE2	Decoded byte select	0
30	BYTE1	Decoded byte select	0
31	BYTE0	Decoded byte select	0

Main Store Controller Error Register 2 (3B20D Computer)

Table 4-42. Main Store Controller Error Register 2 (3B20D Computer)

Bit	Name	Definition	Active
0	PFB3	Parity failure byte 3	1
1	PFB2	Parity failure byte 2	1
2	PFB1	Parity failure byte 1	1
3	PFB0	Parity failure byte 0	1
4	PFPZ	Parity failure hamming bit Z	1
5	PFPY	Parity failure hamming bit Y	1
6	PFPX	Parity failure hamming bit X	1
7	GMREFAD	Maintenance order. Gate main refresh address into address trap register	1
8	BLKERC	Maintenance state. Block error C	1
9	BLKRFD	Maintenance state. Block refresh data parity check	1
10	INHWH	Maintenance state. Inhibit hamming bit write	1
11	INHANC	Maintenance state. Inhibit hamming correction	1
12	DRA02	Duplicate refresh address bit 2	0
13	DRA03	Duplicate refresh address bit 3	0
14	DRA04	Duplicate refresh address bit 4	0
15	DRA05	Duplicate refresh address bit 5	0
16	DRA06	Duplicate refresh address bit 6	0
17	DRA07	Duplicate refresh address bit 7	0
18	DRA08	Duplicate refresh address bit 8	0
19	DRA09	Duplicate refresh address bit 9	0
20	BLKSCM	Maintenance order. Block store complete	1
21*	TAP0	Trapped address parity byte 0	1
22*		Set to 1 for EMM	1
23†	TA25	Trapped address bit 25 (VLMM)	1
24	TIMS02	Maintenance state. Refresh rate	1
25	TIMS04	Maintenance state. Refresh rate	1
26	TIMS08	Maintenance state. Refresh rate	1
27	TIMS32	Maintenance state. Refresh rate	1
28‡	PFFPA	Parity failure PA	1
29‡		TN56 arrays (1)	1
30	NREF	Need refresh signal	1
31	FREF	Force refresh signal	1

* Valid for UN59C or UN618 MASC in EMM or VLMM; unused otherwise.

† Valid for UN618 MASC in VLMM; unused otherwise.

‡ Valid for UN59 and UN618 MASC; unused otherwise.

Main Store Controller Trapped Address Register (3B20D Computer)

Table 4-43. Main Store Controller Trapped Address Register (3B20D Computer)

Bit	Name	Description	Active
0	CPF	Correctable parity fail indicator	1
1	DPF	Double parity fail indicator	1
2	TA02	Trapped address bit 2	1
3	TA03	Trapped address bit 3	1
4	TA04	Trapped address bit 4	1
5	TA05	Trapped address bit 5	1
6	TA06	Trapped address bit 6	1
7	TA07	Trapped address bit 7	1
8	TA08	Trapped address bit 8	1
9	TA09	Trapped address bit 9	1
10	TA10	Trapped address bit 10	1
11	TA11	Trapped address bit 11	1
12	TA12	Trapped address bit 12	1
13	TA13	Trapped address bit 13	1
14	TA14	Trapped address bit 14	1
15	TA15	Trapped address bit 15	1
16	TA16	Trapped address bit 16	1
17	TA17	Trapped address bit 17	1
18	TA18	Trapped address bit 18	1
19	TA19	Trapped address bit 19	1
20	TA20	Trapped address bit 20	1
21	TA21	Trapped address bit 21	1
22	TA22	Trapped address bit 22	1
23	REF/SYS	Refresh/System address trapped	0/1
24	TAP3	Trapped address parity byte 3	1
25	TAP2	Trapped address parity byte 2	1
26	TAP1	Trapped address parity byte 1	1
27	TRAP	Trapped address indicator	1
28	ERRD	Error D indication	1
29	ERRC	Error C indication	1
30	TA23	Trapped address bit 23*	1
31	TA24	Trapped Address bit 24†	1

* Valid for UN59 or UN618 MASC; unused otherwise.

† Valid for UN59 or UN618 MASC with EMM or VLMM; software generated module selected otherwise.

Memory Registers

Table 4-44. 3B21D Computer Main Memory Register

Bit	Definition
0	Trapped Address Parity Bit 3 (Address Bits 7:0) (EVEN)
1	Trapped Address Parity Bit 2 (Address Bits 15:8) (EVEN)
2	Trapped Address Parity Bit 1 (Address Bits 23:16) (EVEN)
3	Trapped Address Parity Bit 0 (Address Bits 27:24) (EVEN)
4	Command Bit Parity Error (Error A)
5	Byte Select 3
6	Byte Select 2
7	Byte Select 1
8	Byte Select 0
9	Block Error C (Decoded Maint Op)
10	Block Error D (Decoded Maint Op)
11	Latch Maintenance Operation Off Data (Decoded Maint Op)
12	Block Store Complete (Decoded Maint Op)
13	Block Store Go Error (Error A)
14	Clear Error A (Decoded Maint Op)
15	Clear Error D (Decoded Maint Op)
16	Set Address Loop-Around (Error A) (Decoded Maint Op)
17	End Address Loop-Around (Decoded Maint Op)
18	Memory Control Soft Reset (RESET10)
19	Address Parity Error (Error A)
20	Data Parity Error (Error A)
21	MC General Error (Error B)
22	March Test Complete
23	Inhibit Error A (Latched Maint Op D1)
24	March Test Stop (Latched Maint Op D2)
25	Memory Control Hard Reset (RESET20)

Table 4-45. Memory Control Bank Address Boundary Register

Bit	Description
0-11	Bank Lower Address Boundary Value
12-23	Bank Upper Address Boundary Value
24-31	RESERVED

Table 4-46. Memory Control Command Register

Bit	Description	Definition
0	Run March Test	0 = OFF 1 = ON
1-8	Banks [7:0] March Test Enable	0 = OFF 1 = ON
9	Qualify March Test	0 1 = Stop on Error
10	Page Mode Memory Column Address Complement/Increment	0 = Complement 1 = Increment
11	RESERVED	
12-13	Page Mode Operation Number of Accesses [# of Column Address Strokes (CASs)]	00 = 2 01 = 4 10 = 8 11 = 16
14-15	Address Queue Depth	00 = 1 01 = 2 10 = 3 11 = 4
16-17	Refresh Interval Selection	00 = 376 Clocks 01 = 504 Clocks 1X = RESERVED
18	Memory Error Scrubbing Mode Normal Read Operation	0 = ON 1 = OFF
19-26	Bank [7:0] Status	0 = In Service 1 = Out of Service
27	RESERVED	
28	Automatic Internal Refresh Generation Mode	0 = ON 1 = OFF
29	Odd/Even Address Parity	0 = Even 1 = Odd
30	Address Parity Checking	0 = Off 1 = On
31	RESERVED	

Table 4-47. Memory Control Error Register

Bit	Description
0	March Test Error
1	Refresh Abort Error
2	Bank Error
3	State Machine Error
4-31	RESERVED

Table 4-48. Memory Control Error Address Register

Bit	Description
0-1	RESERVED
2-31	Latched Error Address

Table 4-49. Memory Control Status Register

Bit	Description	Definition
0-3	Type Field	
4-11	Bank [7:0] Status	0 = Present 1 = Absent
12-13	Bank 0 Size	00 = 256K X 1 or 256K X 4 DRAM 01 = 1M X 1 or 1M X 4 DRAM 10 = 4M X 1 or 4M X4 DRAM 11 = 16M X 1 or 16M X 4 DRAM
14-15	Bank 1 Size	
16-17	Bank 2 Size	
18-19	Bank 3 Size	
20-21	Bank 4 Size	
22-23	Bank 5 Size	
24-25	Bank 6 Size	
26-27	Bank 7 Size	
28	Chip Version	
29-31	March Test Fault	000 = Bank 0 001 = Bank 1 010 = Bank 2 011 = Bank 3 100 = Bank 4 101 = Bank 5 110 = Bank 6 111 = Bank 7

Table 4-50. Memory Control Strobe Shape Register

Bit	Description
0-2	Row Address Strobe (RAS) Precharge Time
3	Row Address Setup Time
4	Row Address Hold Time
5-7	RAS Pulse Width
8-9	Write Data Setup Time
10-12	CAS Pulse Width First Read Access Page Mode Operation
13-15	CAS Pulse Width Additional Accesses Page Mode Operation
16-17	CAS Precharge Time
18-19	RAS Hold Time
20	Data Setup Time Write Back Partial and Scrubbing Operations
21-23	CAS Pulse Width First Write Access Page Mode Operation
24-31	RESERVED

Microinterrupt Error Register

In Table 4-51 the register is active low. (If bit = 0, the function is asserted.) This register is also known as Firm Register B.

Table 4-51. Microinterrupt Error Register

Bit	Error	Class
0	Unused	—
1	Unused	—
2	Unused	—
3	Unused	—
4	ATB error	Stop-and-switch
5	Unused	—
6	Unused	—
7	My store time-out	Stop-and-switch
8	MYSERC	On-line error interrupt
9	Unused	—
10	Unused	—
11	Unused	—
12	Unused	—
13	Unused	—
14	Unused	—
15	Other store time-out	Off-line error interrupt
16	Unused	—
17	Unused	—
18	Unused	—
19	Unused	—
20	Unused	—
21	Protection violation	Software error interrupt
22	VORA	Software error interrupt
23	MYSERB	Software error interrupt
24	Unused	—
25	Privileged instruction violation	Software error interrupt
26	Unused	—
27	Unused	—
28	Unused	—
29	Unused	—
30	Unused	—
31	Unused	—

Microinterrupt Error Register 1

Microinterrupt Error Register 1 (UER1), defined in Table 4-52, is also known as Firm Register C. Values not listed in the table are valid and should be ignored.

Table 4-52. Microinterrupt Error Register 1

Value	Error	Class
00000001	Segment index too large	Software error interrupt
00000008	Page invalid	Software error interrupt
FFFFFFFF	IO parity divert error if found on hardware interrupt with ER bit 19 active	Hardware error interrupt
7FFFFFFF	Illegal switch from private to kernel stack	Software error interrupt
3FFFFFFF	Illegal instruction	Software error interrupt
1FFFFFFF	Illegal instruction or operand subdecode	Software error interrupt
0FFFFFFF	No interrupt source during external interrupt entry	Software error interrupt
07FFFFFF	Unused microstore location was executed	Software error interrupt
03FFFFFF	No error showing during error microinterrupt entry	Software error interrupt
01FFFFFF	Unable to flush contents of ATB in purge ATB instruction	Software error interrupt
0001FFFF	Virtual out-of-range error. Tried to read memory out of virtual address space - beyond 64 megabytes.	Software error interrupt
12345678	MRF caused by the microcode or software	Software error interrupt
12345679	Processor going through level 2 (SAS) MRF	Software error interrupt
1234567A	Stop-and-switch because of an error microinterrupt that proved fatal	Software error interrupt

Processor Status Word Register

In Table 4-53 the register is active high. (If bit = 1, the function is asserted.)

Table 4-53. Processor Status Word Register

Bit	Description
0	Carry flag
1	Negative flag
2	Overflow flag
3	Zero flag
4	Kernel stack on (yes = 1)
5	Unused
6	Interrupt stack on (yes = 1)
7	Memory management on (yes = 1)
8	Source: SSBR mode = 1, PSBR mode = 0
9	Destination: SBR mode = 1, PSBR mode = 0
10	Primary segmentation base register index
11	Primary segmentation base register index
12	Primary segmentation base register index
13	Secondary segmentation base register index
14	Secondary segmentation base register index
15	Secondary segmentation base register index
16	Emulation control: opcode decoding (Spare for the 3B21D computer)
17	Emulation control: opcode decoding (Spare for the 3B21D computer)
18	Emulation control: program counter increment: half-word = 0, full-word = 1 (Spare for the 3B21D computer)
19	Emulation control: interrupt control — control interrupt recognition when instruction halfword is available (Spare for the 3B21D computer)
20	Set execution level privilege (yes = 1)
21	Normal input/output privilege (yes = 1) (3B20D computer)
	Maintenance privilege (yes = 1) (3B21D computer)
22	System IO privilege (yes = 1)
23	Write PSW privilege (yes = 1)
24	Execution level (0-15)
25	Execution level
26	Execution level
27	Execution level
28	Spare
29	Spare
30	Processor mode → 00 kernel mode
31	Processor mode → 01 kernel process
	10 supervisor process

There are eight ATBs in the system. They are currently assigned by the *UNIX* RTR operating system as follows:

Bit	Description
10, 11, 12	RTR Process Assignments
0	Kernel
1	Kernel Process
2	Supervisor/User Process
3	File Manager
4	Disk Driver
7	IO Driver

Pulse Point Register

In Table 4-54 the register is active high. (If bit = 1, the function is asserted.)

Table 4-54. Pulse Point Register

Bit	Meaning
0	Read the channel data buffer (RD)
1	Read the channel status register (RST)
2	Read channel interrupt state (RINT)
3	Read channel service request (RSR)
4	Input/Output interrupt acknowledge (IAK)
5	Channel error acknowledge (EACK)
6	Service request acknowledge (SRACK) (not used in the 3B21D computer)
7	Idle channel sequencer (IDLE)
8	Clear channel errors (CLRER)
9	Write the channel data buffer (WD)
10	Write channel control/address register (WCA)
11	EAI (not used in the 3B20D computer)
12	Unused
13	Unused
14	Backup maintenance channel
15	Backup maintenance channel
16	BGB pulse points
17	BGB pulse points
18	BGB pulse points
19	BGB pulse points
20	BGB pulse points
21	Test sync for the 3B20D computer BGB pulse points for the 3B21D computer
22	Interrupt EAI
23	Interrupt EAI
24	Clear timer
25	Increment timer
26	Test sync (not used in the 3B20D computer)
27	IO read clock
28	IO response clock
29	Error register clear
30	Enable a
31	Enable b

Store Control Register

In Table 4-55 the register is active low. (If bit = 0, the function is asserted.)

Table 4-55. Store Control Register

Bit	Description
0	ATB invalidation counter
1	ATB invalidation counter
2	ATB invalidation counter
3	ATB invalidation counter
4	ATB invalidation counter
5	ATB invalidation counter
6	ATB invalidation counter
7	ATB invalidation counter
8	ATB invalidation counter (3B20D computer) ATB block select (3B21D computer)
9	ATB invalidation counter (3B20D computer) ATB block select (3B21D computer)
10	Enable counter SMM and EMM, ATB invalidation counter in VLMM (3B20D computer) ATB block select (3B21D computer)
11	Half-Word available
12	Program counter (PA) shadow
13	Program counter (PA) shadow
14	Program counter (PA) shadow
15	Program counter (PA) shadow
16	Access
17	Fetch
18	Write
19	Clear (3B20D computer) Read-Modify-Write (3B21D computer)

Table 4-55. Store Control Register (Contd)

Bit	Description
20	Bypass ATB
21	Byte
22	Half
23	Stack
24	SAR auto
25	Memory arbiter reset (not used in the 3B20D computer)
26	Not used in SMM or EMM. Enable counter in VLMM (3B20D computer) ATB flush counter enable (3B21D computer)
27	ATB select
28	Counter select
29	Invalidate ATB
30	Other store
31	Maintenance

System Status Register

In Table 4-56 the register is active low unless otherwise indicated. (If bit = 0, the function is asserted.)

Table 4-56. System Status Register

Bit	Description
0	CC identification (CC0 = 1, CC1 = 0)
1*	Boot time hardware configuration indicator
2	Simplex = 1, duplex = 0 (not used in the 3B21D computer)
3	Request out-of-service key
4	Initialization sequence control
5	Initialization sequence control
6	Force boot device primary
7	Force boot device secondary
8	Panel interrupt
9	Force off-line
10	Force on-line
11	Inhibit sanity timer
12	Enable panel interrupt. (not used in the 3B20D computer)
13	Cache bypass
14	Emergency action interface MRF
15	Power clear

*** 3B20D Computer:**

Bit 1 = 0 for Release 1 indicates UN616/UN617 enlarged cache

Bit 1 = 1 for Release 1 indicates using standard cache

For Releases 6 and 21

Bit 1 = 0 indicates VLMM (MASC can accommodate 64 Mbytes physical memory)

Bit 1 = 1 indicates the MASC is not using VLMM

3B21D Computer:

Bit 1 = 0 indicates LDFT from IOP tape drive

Bit 1 = 1 indicates LDFT from SCSI DAT unit drive

Table 4-56. System Status Register (Contd)

Bit	Description
16	CC on-line
17	Halt
18	Block interrupts
19	Enable update writes
20	Isolate DMA from my store
21	Isolate update from my store
22	Isolate Expansion slots from my store (not used in the 3B20D computer)
23	Block hardware checks
24	EAI bus
25	EAI bus
26	EAI bus
27	EAI bus
28	Stop
29	Block timer circuit
30	IO disable
31	Power key

Contents

Introduction	5-1
Purpose of REX	5-1
REX Operation	5-1
REX Scheduling	5-2
REX Scheduling Recommendations	5-2
REX Manual Commands	5-3
■ Execute	5-4
■ Stop	5-4
■ Inhibit	5-4
■ Allow	5-5
■ Display Inhibits	5-5
REX Inhibits	5-7
REX Inhibit Recommendations	5-8
REX Inhibit Implementations	5-9
■ REX Inhibits Prior to Enhanced AM REX Inhibit and Display Feature	5-9
■ REX Inhibits with the Enhanced AM REX Inhibit and Display Feature	5-10
REX Summary	5-12

Introduction

This section provides a description of the *UNIX*® Real-Time Reliable (RTR) operating system Routine Exercise (REX) process and procedures for the 3B20D and 3B21D computers. Included are detailed descriptions and examples of the REX craft interface and recommendations for scheduling REX and inhibiting REX.

Purpose of REX

The purpose of REX in the *UNIX* RTR operating system is to routinely schedule hardware tests on 3B20D/3B21D computers in order to detect latent faults in in-service units. REX is an important component of the overall reliability and fault tolerance of the 3B20D/3B21D computers. Simply stated, *UNIX* RTR operating system REX is a scheduled process that initiates maintenance requests on in-service hardware communities. A hardware community is a major device controller (top unit) and its associated subdevices.

REX Operation

The REX processor searches the *ucb* records of the database for hardware communities (top units) that can be removed from service. When REX locates a hardware community that can be removed from service, REX requests a remove, diagnose, and restore of the hardware community. When the exercise of the hardware community is completed, REX resumes searching the database for the next hardware community that can be removed from service and exercised.

REX exercises the Control Unit (CU) hardware communities first and in a slightly different manner than that previously described. If the non-active CU is in the standby state, REX does remove, diagnose, and restore requests on the standby CU. If the remove, diagnose, and restore requests are successful, REX also requests a soft-switch of the CUs. This makes the REX exercised standby CU the active CU and makes the formerly active CU the standby CU. REX does NOT exercise the formerly active now standby CU so that it remains standby. In case the newly active CU encounters difficulties after the soft switch. Therefore, each time REX is run, only one CU may be completely exercised and switched to active.

REX Scheduling

A single *UNIX* RTR operating system REX process exercises all the 3B20D/3B21D computers hardware communities and application hardware communities. The REX process is started by the cron process, which dispatches scheduled tasks according to the contents of the *crontab* file. The default *UNIX* RTR operating system *crontab* file, which is installed on the 3B20D/3B21D computers file system as */usr/lib/crontab*, has REX scheduled to run daily at midnight. Hardware communities cannot be individually scheduled or unscheduled for REX. However, the user can inhibit the REX process from exercising one, some, or all hardware communities. Refer to the “REX Inhibits” and “REX Inhibit Recommendations” sections for further information on using REX inhibits.

REX Scheduling Recommendations

REX is an important component of the overall reliability and fault tolerance of the 3B20D/3B21D computers. The REX process should be scheduled to run daily. When REX is scheduled daily and there is no other maintenance activity, most hardware communities are exercised once every 24 hours, except the CUs since each CU is completely exercised once every 48 hours.

REX is scheduled to begin at a time when the software load on the computer is likely to be minimal for several hours. REX is started at midnight because REX takes several hours to complete even under light system load because of the following reasons:

- Additional long running diagnostic phases may be added to the regular diagnostic phases when the requesting process is REX.
- REX exercises the hardware communities one at a time rather than concurrently for the following overall reliability reasons:
 - Minimize the number of hardware communities that are Out of Service (OOS) at any one time.

- Minimize the time each hardware community is OOS by having a minimum number of active maintenance requests.
- Minimize the chance that manually initiated maintenance requests will be blocked by active maintenance requests from REX.

REX can be rescheduled by changing the */usr/lib/crontab* file.



CAUTION:

Failure to execute the REX process daily may degrade the ability of the 3B20D/3B21D computers to recover from hardware or software faults.

Several factors that follow should be considered before changing the schedule for REX:

- REX can be unscheduled by errors in the */usr/lib/crontab* file or by removing REX from the */usr/lib/crontab* file.
- REX removes hardware communities from service.
- REX maintenance activities require system resources.
- REX maintenance activities may block or slow other maintenance activities.
- REX takes several hours to complete even under light system load.

Information on how to modify the entries in the */usr/lib/crontab* file can be found in the “Using CRON” section of Chapter 10, “Maintaining Software Integrity.”

REX Manual Commands

Refer to the following information products (IP) for further explanation of the input messages and output messages described in this section:

- 254-303-110, *PDS Input Messages Manual*
- 254-303-111, *PDS Output Messages Manual*
- 254-303-112, *MML Input Messages Manual*
- 254-303-113, *MML Output Messages Manual*.

Execute

The *UNIX* RTR operating system REX process removes in-service hardware communities and uses system resources for long periods of time. The REX process is intended to be initiated only by the cron process using the schedule in the */usr/lib/crontab* file. The REX process is not intended to be initiated by the user. Therefore, there is no input message to start or execute the REX process.

Stop

The *UNIX* RTR operating system REX process is an integral part of the fault tolerance and high availability of the 3B20D/3B21D computers. The REX process is not intended to be stopped by the user. Stopping REX would leave the hardware community currently under test in the OOS state. Manual action would be required to restore the hardware community to service. Since REX cannot be inadvertently started by a user command, and since it is desirable for REX to run to completion, there is no input message to stop the REX process.

The user can stop an active or waiting REX process maintenance request in the Deferred Maintenance Queue (DMQ) of the Maintenance Input Request Administrator (MIRA). **STOP:DMQ** may leave the hardware community in the OOS state if the REX maintenance request was active, and manual action may be required to restore the hardware community to service. **STOP:DMQ** does not stop the REX process. The format of the **STOP:DMQ** input message is as follows:

In PDS, {**STOP** | **STP**}:DMQ:{**ACTIVE** | **WAITING** | unit #[,subunit #]}!
In MML, {**STOP** | **STP**}:DMQ:{**ACTIVE** | **WAITING** | unit=#[,subunit=#]};

Inhibit

Inhibiting REX is usually desirable during certain procedures such as software update, growth, or retrofit. The user can temporarily inhibit the REX process from exercising one, some, or all hardware communities, and/or the user can temporarily inhibit all REX process maintenance requests at the MIRA DMQ. Explanations of the **ALW:DMQ**, **INH:DMQ**, and **OP:DMQ** input messages are included in this section. Refer to the "REX Inhibits" section for information on how REX inhibits are implemented.



CAUTION:

Temporary REX inhibits in any form may degrade the ability of the 3B20D/3B21D computers to recover from hardware or software faults.

The **INH:REX** input message is used to temporarily inhibit REX on one, some, or all hardware communities. The format of the **INH:REX** input message is as follows:

In PDS, **INH:REX:{ALL | unit [#]}!**
In MML, **INH:REX:{ALL | unit[=#]};**

The **INH:DMQ** input message can be used to temporarily inhibit maintenance requests from the REX process in MIRA's DMQ. Either source "ALL" or "REX" being active in MIRA's DMQ inhibited sources list inhibits REX process maintenance requests. The format of the **INH:DMQ** input message is as follows:

In PDS, **INH:DMQ:SRC {ALL | REX | ADP}!**
In MML, **INH:DMQ:SRC={ALL | REX | ADP};**

Allow

The **ALW:REX** input message is used to remove a temporary REX inhibit on one, some, or all hardware communities. The format of the **ALW:REX** input message is as follows:

In PDS, **ALW:REX:{ALL | unit [#]}!**
In MML, **ALW:REX:{ALL | unit[=#]};**

The **ALW:DMQ** input message is used to remove the temporary inhibit on maintenance requests from the REX process in MIRA's DMQ. Both sources "ALL" and "REX" must not be active in MIRA's DMQ inhibited sources list for REX maintenance requests to be allowed in MIRA's DMQ. The format of the **ALW:DMQ** input message is as follows:

In PDS, **ALW:DMQ:SRC {ALL | REX | ADP}!**
In MML, **ALW:DMQ:SRC={ALL | REX | ADP};**

Display Inhibits

The **OP:REXINH** input message is used to request a report of the temporary REX inhibits in the hardware communities. The format of the **OP:REXINH** input message is as follows:

In PDS, **OP:REXINH!**
In MML, **OP:REXINH;**

The **OP REXINH** output message should resemble the following:

```
OP REXINH {COMPLETED | IN PROGRESS}  
[NO] REX INHIBITED UNITS [:]  
[UNIT #]  
" "  
" "  
" "
```

With the Enhanced Administrative Module (AM) REX Inhibit and Display Feature, the report includes MIRA's DMQ REX inhibit, if it is active. The format of the new **OP REXINH** output message is as follows:

```
OP REXINH {COMPLETED | IN PROGRESS}  
[NO] REX INHIBITS  
[UNIT #]  
" "  
" "  
" "  
[[REX INHIBITED IN DMQ] |  
[REX DMQ STATUS NOT AVAILABLE]]
```

With the Enhanced AM REX Inhibit and Display Feature, the PERM option to the **OP:REXINH** input message can be used to request a report of the permanent REX inhibits in the hardware communities. The format of the new **OP:REXINH** input message is as follows:

In PDS, **OP:REXINH[:PERM]!**
In MML, **OP:REXINH[:PERM];**

The format of the new **OP REXINH** output message when reporting permanent REX inhibits is as follows:

```
OP REXINH {COMPLETED | IN PROGRESS}  
[NO] PERMANENT REX INHIBITS  
[UNIT #]  
" "  
" "  
" "
```

The **OP:DMQ** input message is used to request a report of the temporary inhibits in MIRA's DMQ inhibited sources list. Either source "ALL" or "REX" being active in MIRA's DMQ inhibited sources list inhibits REX process maintenance requests.

The format of the **OP:DMQ** input message is as follows:

In PDS, **OP:DMQ!**

In MML, **OP:DMQ;**

The format of the **OP DMQ** output message is as follows:

OP DMQ

REQUEST ACTIVE

.
.
.

REQUEST WAITING

.
.
.

INHIBIT SOURCES

{NONE | [[ALL] | [REX] | [ADP]]}

OP DMQ COMPLETED

REX Inhibits

The REX process is permanently inhibited from exercising certain hardware communities where REX is not desired for reliability reasons. The user does not have control over permanent REX inhibits on hardware communities in early *UNIX* RTR operating system releases. The user has control over permanent REX inhibits on hardware communities in later *UNIX* RTR operating system releases.

The REX process can be temporarily inhibited from exercising one, some, or all hardware communities. The temporary REX inhibit on a hardware community is controlled by administering a bit in the top unit *ucb* record of the incore database. The user has direct control over temporary REX inhibits on hardware communities.

REX processes-maintenance requests for RMV, DGN, and RST can also be temporarily inhibited at the MIRA DMQ. Either source "ALL" or "REX" being active in MIRA's DMQ inhibited sources list inhibits REX process maintenance requests. When REX is inhibited in MIRA's DMQ, REX process maintenance requests are denied execution and canceled. The user has direct control over MIRA's DMQ inhibited sources list.

REX Inhibit Recommendations

Since REX is an important component of the overall reliability and fault tolerance of the 3B20D/3B21D computers, REX inhibits should be used with care.



CAUTION:

Temporary REX inhibits in any form may degrade the ability of the 3B20D/3B21D computers to recover from hardware or software faults.

In general, use the minimum REX inhibit required to achieve the desired results, and remove the REX inhibit as soon as possible. It may not be necessary to inhibit REX at all if REX is not scheduled for the next few hours. (Refer to the “REX Scheduling Recommendations” section.) The following are some specific recommendations for using REX inhibits:

- Use the **INH:REX** input message to temporarily inhibit REX on specific hardware communities.
- Avoid using the **INH:REX** input message to keep a constant temporary REX inhibit on specific hardware communities.
- Avoid using the **INH:REX:ALL** input message, especially in command scripts. It inhibits all hardware communities from REX. The inhibit need to be removed from those hardware communities where temporary REX inhibits are NOT desired.
- Avoid using the **ALW:REX:ALL** input message, especially in command scripts. It removes the temporary REX inhibits from all hardware communities, including hardware communities where temporary REX inhibits are desired. The desired temporary REX inhibits have to be reentered manually.
- Use the **INH:DMQ:SRC REX** input message to temporarily block all REX requests.
- Use the **ALW:DMQ:SRC REX** input message to reverse the affects of the **INH:DMQ:SRC REX** input message.
- Avoid using the **INH:DMQ:SRC ALL** input message as it blocks requests from other automatic processes which may need to run.

REX Inhibit Implementations

UNIX RTR Operating System Release 6.7 and Release 21.1 have the Enhanced AM REX Inhibit and Display Feature. This feature changes the way permanent and temporary REX inhibits are implemented on hardware communities. This feature also notifies the Application Integrity Monitor (AIM) process when there are changes to the temporary REX inhibits. The following sections explain how REX inhibits are implemented on hardware communities before and after the feature.

REX Inhibits Prior to Enhanced AM REX Inhibit and Display Feature

UNIX RTR Operating System Releases 1 through 6.6 and Release 6.8 implement REX inhibits on hardware communities in the following manner:

- The **INH:REX** and **ALW:REX** input messages provide temporary REX inhibits by changing the top unit *ucb* record *rexinh* field to “y” and “n”, respectively. The *rexinh* field appears in the Recent Change and Verify (RC/V) *ucb* form as “19. rexinh.”
- The **ALW:REX** and **INH:REX** input messages do not alter the *rexinh* field of a *ucb* record whose *device_type* field identifies certain device types. The *device_type* field appears in the RC/V *ucb* form as “12. device_type.”
- When an **ALW:REX** or **INH:REX** input message cannot alter the *rexinh* field(s) of the hardware community(ies) requested because of permanent REX inhibits, the input message responds with ?E DATA FIELD ERROR or a ? D response if the input message catalog is bypassed. Exceptions are **ALW:REX:ALL** and **INH:REX:ALL** that respond with OK or PF.
- The contents of top unit *ucb* record *device_type* field is used to permanently inhibit REX on a hardware community type, such as DFC. The *device_type* field is used to identify the device type of the unit described by the *ucb* record. The *device_type* field cannot be arbitrarily changed to permanently inhibit or allow REX as other processes use the contents of the *device_type* field.
- The *device_type* and *rexinh* fields can be changed by RC/V. However, an RC/V form check requires the *rexinh* field to be “y” when the *device_type* field identifies certain hardware community types, such as “DEV_DFC” and “DEV_MSGS.”
- An RC/V form check requires all Unit Control Blocks (UCBs) under the top unit UCB and the top unit UCB to be in the “UNEQUIP” state before RC/V will make changes to the *device_type* or *rexinh* fields in the incore database.

Top unit UCB *device_type* and *rexinh* fields control the REX process and **OP:REXINH** output as shown in Table 5-1.

Table 5-1. Early REX Inhibits

ucb (static) 12. device_type	ucb (static) 19. rexinh	REX Mode	OP:REXINH
Note 1	n	run - REX allowed	(void)
Note 1	y	skip - REX temporarily inhibited	unit inhibited
Note 2	y (required)	skip - REX permanently inhibited	(void)

Notes:

1. Releases 1 – 6.6 and Release 6.8, all device types other than those mentioned in Note 2.
2. Releases 1 – 6.6 and Release 6.8, device type DEV_DFC.
Releases 6.3 – 6.6 and Release 6.8, include device type DEV_MSGS.
Release 6.8, includes device types DEV_SOFT and DEV_RUN.

REX Inhibits with the Enhanced AM REX Inhibit and Display Feature

UNIX RTR Operating System Release 6.7 and Release 21.1 have the Enhanced AM REX Inhibit and Display Feature. The Enhanced AM REX Inhibit and Display Feature implements REX inhibits on hardware communities in the following manner:

- A new field in the *ucb* record called *t_rexinh* is created for temporary REX inhibits. New field appears in the RC/V *ucbdyn* form as “23. t_rexinh” with legal values of “y” and “n”. The **INH:REX** and **ALW:REX** input messages provide temporary REX inhibits by changing the top unit *ucb* record *t_rexinh* field to “y” and “n”, respectively. The *t_rexinh* field can only be changed by the **ALW:REX** and **INH:REX** input messages and not by RC/V.
- The **ALW:REX** and **INH:REX** input messages do not alter the *t_rexinh* field of a *ucb* record whose *rexinh* field is “y”.
- When an **ALW:REX** or **INH:REX** input message cannot alter the *t_rexinh* field(s) of the hardware community(ies) requested because the *rexinh* field(s) are “y”, the input message responds with the following:

```
{ALW | INH} REX NOT STARTED
CONFLICT WITH CURRENT SYSTEM STATUS
```

or “NG” if the input message catalog is bypassed.

Release 6.7 exceptions are **ALW:REX:ALL** and **INH:REX:ALL** input messages that do not have the NG response. They respond with OK or PF and can appear to have succeeded while actually they may not be able to change any *t_rexinh* fields.

- The *ucb* record *rexinh* field now indicates the permanent REX inhibited status. It can only be changed by RC/V and is not affected by the **ALW:REX** and **INH:REX** input messages. The applications or sites can permanently inhibit REX on selected hardware communities by setting *rexinh* to “y”.

The *device_type* field is no longer used by REX to determine if a unit type should be permanently inhibited from REX. However, the RC/V form check requiring the *rexinh* field to be “y” for device types “DEV_DFC” and “DEV_MSGS” is retained for reliability reasons.

- Add an RC/V form check that requires the *t_rexinh* field to be “n” before it accepts changes to the *rexinh* field in the incore database. If it is not, RC/V informs the user that *rexinh* changes are not allowed when *t_rexinh* is “y”. This form check prevents the temporary REX inhibit status sent to the AIM process from silently becoming incorrect because of RC/V activity to the *rexinh* field.
- Change the RC/V form check that requires all UCBs under the top unit UCB and the top unit UCB to be in the “UNEQUIP” state before RC/V can change the *rexinh* field on a *ucb* record in the incore database. The RC/V form check now allows changes to the *rexinh* field when the top unit UCB is in the OOS state in the incore database. This makes it fairly easy for the user to permanently inhibit or allow REX on a hardware community.

Top unit UCB *rexinh* and *t_rexinh* fields control the REX process and **OP:REXINH** output for the hardware community as shown in Table 5-2.

Table 5-2. New REX Inhibits

ucb (static) 19. rexinh	ucbdyn (dynamic) 23. t_rexinh	REX Mode	OP:REXINH
n	n	run - REX allowed	(void)
n	y	skip - REX temporarily INH	unit inhibited
y	n (required)	skip - REX permanently INH	(void)

The Enhanced AM REX Inhibit and Display Feature also improves the reporting of REX inhibits in the following manner:

- Improve the **OP REXINH** output message to include the REX temporary inhibit state in MIRA's DMQ.
- The "PERM" option is added to the OP:REXINH input message. When this option is specified, the **OP REXINH** output message reports only the permanent REX inhibits.
- When temporary REX inhibits are changed in either *t_rexinh* field of a top unit UCB or in MIRA's DMQ, send a message to the AIM process updating AIM temporary REX inhibit status.
- To give the reasons for the off-normal indication, 5EUS activates the AM REX INH box on 5EUS MCC Page 110 and maps "9" poke from this box to **OP:REXINH:AM** input message. The "9" poke supports the "PERM" option. It maps the poke "9,PERM" to an **OP:REXINH:AM,PERM** input message.

REX Summary

When the *UNIX* RTR operating system REX process has finished exercising all the hardware communities, it prints an output message that summarizes the REX results. The REX summary output message does not include hardware communities that are permanently inhibited from REX.

It is important to note that the contents of the REPT REX SUMMARY output message was collected over the time period that REX was running, which is usually several hours. The status of units in this output message can be out of date with current unit status. The following is the format of the REPT REX SUMMARY output message:

```
REPT REX SUMMARY {IN PROGRESS | COMPLETED}
TIME      EQUIPMENT ID  RESULT  RET    STATE
a         b c [d e]    f      [g]    h
```

Where:

a = Local time (in hours:minutes:seconds) when REX of the unit finished.

b = Unit name.

c = Unit number.

d = Subunit name (optional).

e = Subunit number (optional).

f = Result of REX on the specified unit [and subunit]:

ABT - Diagnostics were aborted.

ATP - Diagnostics were All Tests Pass.

CATP - Diagnostics were Conditional ATP.

ERR - Error return code from MIRA.

FAULT - Unit was excessively faulty.

INH - REX was temporarily inhibited for this unit.
MAN - Unit was removed manually.
NODGN - No diagnostic phases exist for this unit.
NRPY - No reply received from MIRA.
NTR - No diagnostic tests were run.
OOS - Unit was OOS and could not be exercised.
PORTSW - Could not port switch.
STF - Diagnostic Some Tests Failed.
TIMEOUT - Time-out occurred.

g = Return code from MIRA. Possible return codes are:

- 2 – REX requests inhibited in MIRA DMQ.
- 3 – MIRA DMQ full.
- 4 – MIRA cannot spawn Diagnostic Monitor (DIAMON) process.
- 5 – Duplicate request already in MIRA DMQ.
- 6 – Request stopped by STOP:DMQ or STP:DMQ input message.
- 7 – System problem or MIRA problem; request aborted.
- 8 – Remove attempt failed.
- 9 – Restore attempt failed.

h = State of the unit after REX exercised it:

- ACT – Active.
- GROW – Grow.
- INIT – Initializing.
- OFL – Off-line.
- OOS – Out of service.
- STBY – Standby
- UNAV – Unavailable.
- UNEQUIP – Unequipped.

Contents

Introduction	6-1
Maintenance Input Request Administrator	6-3
■ MIRA Initialization	6-3
■ Diagnostic Commands	6-4
■ Concurrent Maintenance Requests	6-4
■ Scheduling Requests	6-5
■ Priority Scheme	6-6
■ Canceled and Denied Requests	6-6
Automatic Diagnostics	6-7
Performing Diagnostics	6-8
■ How to Request Diagnostics	6-8
Using the DGN Command	6-8
Requesting Supplementary Phases	6-10
Suggested Routine Test Procedures	6-11
Requesting Interactive Diagnostics	6-12
■ How to Respond to a Denied Request	6-16
■ How to Inhibit and Allow Automatic Maintenance Requests	6-19
■ How to Obtain Status of Requests	6-20
■ How to Abort or Cancel a Request	6-21
■ How to Remove and Restore Units	6-22
■ How to Inform Diagnostic Program of Hardware Changes	6-23
■ How to Interpret Diagnostic Output and Clear Trouble	6-25
Diagnostic Output Message	6-25
Output Message Format	6-26

Contents

■ Main Store Failure Data	6-30
How to Interpret Output from TLP Option	6-30
How to Interpret TLP Equipment Codes	6-35
How to Clear Trouble Using the TLP Circuit Pack List	6-36
How to Clear Trouble Using the Diagnostic Listing	6-38
How to Respond to a Diagnostic Audit Message	6-41
Synchronous Data Link Restore Tool	6-54
Locating Pins	6-57
Processor Recovery Messages	6-59

Introduction

This chapter covers the diagnostic tests for the *UNIX*® Real-Time Reliable (RTR) Operating System Releases 1 and 6.

The diagnostic system for the 3B20D and 3B21D computers is used primarily to detect hardware faults and to verify that a repair is successful.

Diagnostics are requested through automatic and manual sources. The automatic sources are from the fault recovery and the peripheral unit driver subsystems and include the following:

- REX Routine Exercise—requests a conditional restore (which includes diagnostics) at a time of day specified in file */usr/lib/crontab*.
- ADP Automatic Diagnostic Process—requests a conditional restore of units not brought up at boot time and on units removed through fault recovery actions.
- SDLRT Synchronous Data Link Restore Tool—consists of two processes executed by the User-Level Automatic Restart Process (ULARP). These processes periodically check the status of synchronous data link units and controllers associated with the Communications Protocol Handler (CPH) and the Network Protocol Handler (NPH). They also request conditional restores of those units/controllers that are Out of Service (OOS).

Manual sources include the following:

Terminal Input	Maintenance requests can be entered for any unit from any user terminal.
Power Switch	Requests for removal and conditional restore of a unit can be made from the power switch monitor process.

A diagnostic for a unit is divided into individual programs called "phases" which are executed in sequence. A phase usually tests one particular function of a unit. For example, the "Control Unit Diagnostic Test Sequence" section in Appendix A, "3B20D Computer References," and Appendix B, "3B21D Computer References," indicate the sequence for the control unit hardware diagnostic tests.

Normally only certain phases, designated as "regular," are executed when a diagnostic is run. However, optional phases ("supplementary" phases) can be requested by specifying certain parameters in the diagnostic input message. These tests are optional because the phase takes a long time to run or requires manual intervention. They should not be run on a regular basis. However, some of the long-running supplementary phases are run when the REX process originates the request. Information about supplementary phases is included in the "How to Request Diagnostics" section.

Maintenance Input Request Administrator

The Maintenance Input Request Administrator (MIRA) is the message-driven front-end process of the Diagnostic Control Structure (DCS), responsible for scheduling and dispatching remove, diagnose, and restore maintenance requests. MIRA maintains a waiting queue and an active queue to manage and track each maintenance request. If the active queue is not full and the needed resources are available, an incoming request is serviced immediately. Otherwise, the request is inserted into the waiting queue. The requests in the waiting queue are ordered according to priority.

The importance of a maintenance request depends on the type of request and its source. MIRA schedules a request according to its importance relative to other requests. It permits an inhibit of all automatic requests or specific classes of automatic requests to prevent interference with manual requests. When MIRA places a maintenance request in the active queue, it spawns a Diagnostic Monitor (DIAMON) process to service the request. DIAMON processes the request and reports to MIRA when the request is completed and issues output messages when error conditions occur.

MIRA Initialization

MIRA is spawned by the ULARP. It accepts the following arguments:

```
mira      conc=<n1,n2> tty=<string1>  
           dir=<string2> diam=<string3>  
           prty=<p> mtim=<t> rtag=<port>
```

Where:

- <n1> specifies the length of the active queue (the degree of concurrency). <n1> must be greater than <n2>. The default value is 1.
- <n2> specifies the portion of the active queue that is protected from automatic requests. If <n2> is greater than or equal to <n1>, then <n2> is set to <n1> minus one. If <n2> is missing, the default value is 0.
- <string1> specifies the device name of a TTY for output of MIRA's error messages, for example, */dev/ttya*. The default value on the 3B20D and 3B21D computers is */dev/ttya*.
- <string2> specifies the current working directory of MIRA. The default value is the current working directory of MIRA's spawning process.

- `<string3>` specifies the filename of the DIAMON object file. The default value on the 3B20D and 3B21D computers is *ppdiamon*.
- `<p>` specifies a priority increment to be applied to MIRA. A negative number results in a higher Central Processing Unit (CPU) priority. All children of MIRA will inherit the modified priority.
- `<t>` specifies the time-out value for the diagnostic timer process. The default value is 0, which indicates that the timer process will not be used.
- `<port>` specifies the system port number where diagnostic failure messages are to be sent.

D diagnostic Commands

The DCS accepts diagnose (**DGN**), interactively diagnose (**EX**), restore (**RST**), and remove (**RMV**) commands. Complete descriptions of these commands are provided in this chapter. (Refer to the "Performing Diagnostics" section.)

Concurrent Maintenance Requests

A MIRA invocation interface to the DCS supports concurrent service of maintenance requests. The interface's degree of concurrency (length of the active queue) is passed to MIRA when MIRA is initialized.

⇒ NOTE:

The length of the active queue limits the number of concurrent requests because it contains the maximum number of requests that can be active.

When the degree of concurrency is greater than one (1), resource contention can occur when two active requests compete for the same resource. For example, both the Central Control (CC) and Direct Memory Access (DMA) diagnostic require the Control Unit (CU) to execute their diagnostics. Requests in resource contention with active requests must wait for the completion of the conflicting active requests. There are several resource contention detection levels a request must pass before the request is activated.

A first level of resource contention detection assures that an automatic request is not assigned to the portion of the active queue reserved for manual requests only.

A second level of resource contention detection accesses the Equipment Configuration Database (ECD) to determine the equipment name of the top Unit Control Block (UCB). A request will be activated if no requests in the active queue have the same top UCB equipment name.

A third level of resource contention detection involves checking the status of the reserve bit in the ECD for the requested unit(s). This resource contention detection is performed by DIAMON. If DIAMON can reserve the unit, the request is serviced.

- If the requested unit(s) cannot be reserved, it is likely the units are reserved by another active request in MIRA, or by another nondiagnostic process. The request is returned to the waiting queue. The request is not attempted again until a request in the active queue has been completed.
- If the active list is empty, the request is canceled.
- If the canceled request is manual, the following message is printed on the maintenance terminal:

REPT MIRA ERROR "request" RETRY FAILED

or

REPT MIRA CANNOT RESERVE UNITS FOR "request"

Scheduling Requests

The following criteria must be met before MIRA can assign a request to the active queue:

- An active slot is available.
- The request does not have concurrent maintenance request resource contention.

Once a request goes into the active queue, it cannot be canceled to make room for another request. The following action occurs each time an active request completes:

- The units associated with the completed request are unreserved.
- A slot is opened in the active queue.
- The requests in the waiting queue are examined to find those that can be moved to the active queue (no resource contention).
- If more than one is eligible, the one with the highest priority is chosen.

The MIRA formats a diagnostic request as a message and spawns a DIAMON process to service the request.

Priority Scheme

The request header's priority field determines the priority of a request. A request's priority is a number assigned by MIRA.

An active request cannot be preempted by an incoming maintenance request. Only a stop request can terminate an active request before it is completed. Therefore, a maintenance request that cannot be serviced immediately is put into a waiting queue. The waiting queue is divided into two sublists: manual requests [Manual (MAN) or Power Switch Monitor (PSM)] and automatic requests. The combined length of the sublists cannot exceed the length of the waiting queue. Requests in the manual sublist are always serviced before those in the automatic sublist. Requests are ordered within each sublist according to their priority assignments. A priority can be any value between 0 and 32767, with 0 as the lowest priority. The request with the highest priority is always placed at the top of the sublist. Requests with the same priority are ordered according to their arrival time – the first request received is placed closest to the top of the sublist. The value of a priority varies according to the following three factors:

- Request type – diagnose, remove, or restore
- Unit type – central processor or peripheral device
- Request source – PSM, MAN, Automatic Diagnostic Process (ADP), or Routine Exercises (REX).

Canceled and Denied Requests

A request via an input message or a power switch is given the source type MANUAL, while all other requests are given the source type AUTOMATIC. The following rules are applied to decide whether an incoming request will be denied or a request in the waiting queue will be canceled:

- Requests in the active queue are never canceled.
- MANUAL requests in the waiting queue are never canceled.
- A request is canceled when the unit cannot be reserved and there are no other requests in the active queue.
- An incoming AUTOMATIC request is denied if there is a similar MANUAL or AUTOMATIC request in the active queue.
- An incoming AUTOMATIC request is denied if there is a similar MANUAL request in the waiting queue.

- The priority of an incoming AUTOMATIC request is compared with the priority of similar AUTOMATIC request already in the waiting queue. The lower priority request will be canceled or denied.
- If the waiting queue is full, an incoming MANUAL request cancels and replaces the AUTOMATIC request with the lowest priority.

When an AUTOMATIC request is canceled or denied, the calling process is informed.

Automatic Diagnostics

The following information will aid in using the automatic diagnostic feature of the *UNIX* RTR operating system:

- Verify that an ADP requested conditional restore is scheduled after a unit has been removed from service. (Refer to the **OP:DMQ** input command in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)
If a restore is not scheduled, conditionally or unconditionally restore the unit. (Refer to the **RST:** input command in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)
- Check the output messages to determine if the diagnostic was All Tests Pass (ATP), Conditional All Tests Pass (CATP), No Tests Run (NTR), or Some Tests Fail (STF).
- Sources of automatic maintenance requests can be inhibited. (Refer to the **INH:DMQ** input command in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.)

Performing Diagnostics

How to Request Diagnostics

This section describes how to request the various types of diagnostics.

Using the DGN Command

The basic form of the diagnostic input message [in Program Documentation Standard (PDS) and Man-Machine Language (MML)] is as follows:

In PDS, **DGN:unit a[,subunit b]!**
In MML, **DGN:unit=a[,subunit=b];**

The subunit designation may be used to specify subunits that make up the CU when a specific subunit diagnostic is desired. When the subunit is not specified, the named unit and all subunits under it in the diagnostic hierarchy will be tested. Table 6-1 is an index to the diagnostic hierarchy as a function of the 3B20D/3B21D computer and hardware.

Table 6-1. Diagnostic Hierarchy Appendix Index

Hardware	Appendix	Section
3B20D Computer CU	A	CU Diagnostic Hierarchy
3B21D Computer CU	B	CU Diagnostic Hierarchy
IOP	A	IOP Diagnostic Hierarchy
IOP 2	B	IOP Diagnostic Hierarchy
DFC	A	DFC Diagnostic Hierarchy
DFC 2	B	DFC Diagnostic Hierarchy
DFC 3	B	DFC Diagnostic Hierarchy

Because of limitations in the Input/Output Processor (IOP) and Disk File Controller (DFC) drivers, only the controller diagnostic is executed for the IOP and the DFC.

In addition to the basic form of the diagnostic input message, optional parameters may be specified. The full command format is as follows:

In PDS,

DGN:*unit a* [,*subunit b*] [:**RPT** *c*] [,**RAW**] [,**UCL**] [,**REX** | **DEX**] \\
[:**PH** *d* [-*e*]] [,**CONT**] [,**TLP**] [,*helperunit f*] !

In MML,

DGN:*unit=a* [,*subunit=b*] [:**RPT**=*c*] [,**RAW**] [,**UCL**] [,**REX** | **DEX**] \\
[:**DATA**,**PH**=*d* [&*e*]] [,**CONT**] [,**TLP**] [,*helperunit=f*];

The following is a description of the optional fields:

RPT Indicates that the diagnostic should be repeated *n* times. The maximum is 256. Default is 1.

 NOTE:

Repeat does not work with CU diagnostic unless a subunit is specified.

RAW Indicates that the diagnostic results of every phase should be printed. If any failures occur, data for up to 100 failing tests is printed for each failing phase. Normally, only the first five failures per failing phase are printed.

 NOTE:

The **RAW** option should not be used indiscriminately since this option can generate a lengthy output.

UCL Executes diagnostics with no early termination. The basic diagnostic contains many early termination points. If one of these points is reached and previous tests have failed, the diagnostic does not continue. This may occur for the following two reasons:

- Data obtained from the previous tests is sufficient to uncover the problem.
- If previous tests have failed, going further may be dangerous.

The **UCL** option directs the diagnostic to ignore these early termination points.

 CAUTION:

*The **UCL** option should not be specified unless absolutely necessary since continued diagnostic testing may cause undesirable results.*

REX	Requests that regular and Routine Exercise (REX) phases within the specified range be run. If no range is specified, all regular and routine exercise phases are run. [Refer to the “Requesting Supplementary Phases” section, Appendix A (for the 3B20D computer), and Appendix B (for the 3B21D computer)].
DEX	Requests that regular, routine exercise, and demand exercise phases within the specified range be run. If no range is specified, all regular, routine, and demand exercise phases are run. [Refer to Appendix A (for the 3B20D computer) and Appendix B (for the 3B21D computer)].
PH	Requests either a particular phase or a range of phases. This may be a subset of the regular phases and/or a supplementary phase or phases. Using the PH option is the only way that supplementary phases can be run. Refer to Appendix A (for the 3B20D computer) and Appendix B (for the 3B21D computer) for a list of all diagnostic phases. Also, refer to the “Requesting Supplementary Phases” section in Appendix A or B.
CONT	Specifies that subunits under the requested unit (and subunit) are not to be diagnosed. CONT option is the default for DFC and IOP Diagnostic (DGN) requests because of limitations in the DFC and IOP drivers. CONT option is also available for DMA subunit of the CU.
TLP	Generates a list of suspected faulty equipment if failures are detected by the diagnostic. (Refer to the “How to Interpret Output from TLP Option” section.)
<i>helperunit</i>	Name (and member number <i>f</i>) of the unit to be used as a helper for some diagnostic phases.

Refer to the “Requesting Supplementary Phases” and “Requesting Interactive Diagnostics” sections for explanation of variations on the basic input messages. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for details on the possible diagnostic input messages.

Requesting Supplementary Phases

In addition to the regular diagnostic phases, there are supplementary diagnostic phases. Three levels of supplementary phases are available as follows:

- Routine Exercise (REX) phases are those executed in addition to the regular phases when the request originates from the REX process or the REX option is specified on the input request. REX phases test or exercise the unit under test more completely but have a long run time which makes them undersirable as regular phases.
- Demand Exercise (DEX) phases are those executed in addition to the regular and REX phases when the DEX option is specified on the input request. (REX phases are executed by the DEX option.) DEX phases

supplement the regular and REX phases by varying conditions and/or test patterns.

- Demand (only) phases are those executed only when their phase number is specified or included in a range of phase numbers in the input message and the DEX or REX options are NOT specified. (REX and DEX options exclude “demand only” phases.) Demand phases are not suitable for inclusion as regular, REX, or DEX phases because they have one or more of the following characteristics:
 - Very long run time
 - Require manual observation
 - Require manual interpretation of results
 - Require special procedures
 - Require an OOS helper unit.

To exercise the system and identify possible problem areas before they become serious, run the supplementary phases as indicated in Appendix A, Table A-1 (for the 3B20D computer), and Appendix B, Table B-1 (for the 3B21D computer) in addition to the scheduled REX diagnostics. Use the **RAW** and **TLP** options on each diagnostic request.

Suggested Routine Test Procedures

Before starting a routine test procedure, make note of the type of hardware being tested and refer to Table 6-2 for an index to the related procedure found in Appendix A or Appendix B.

Table 6-2. Suggested Routine Test Procedures

Hardware Type	Appendix	Section
3B20D Computer CU	A	CU Suggested Routine Test Procedure
3B21D Computer CU	B	CU Suggested Routine Test Procedure
IOP	A	IOP Suggested Routine Test Procedure
IOP 2	B	IOP 2 Suggested Routine Test Procedure
DFC	A	DFC Suggested Routine Test Procedure
DFC 2	B	DFC 2 Suggested Routine Test Procedure
DFC 3	B	DFC 3 Suggested Routine Test Procedure
DCI	B	DCI Suggested Routine Test Procedure

Requesting Interactive Diagnostics

Interactive diagnostics can be used to:

- Run a diagnostic phase up to a specified point in the data table and then stop.
- Perform a specified group of tests repeatedly.
- Exercise main store.
- Step through diagnostic data table commands.

Use the **EX** command to initiate an interactive diagnostic. The command format is as follows:

In PDS,

EX:unit a [,subunit b] [:[RAW][,UCL]]:PH c [,TLP] [,helperunit d]!

In MML,

EX:unit=a [,subunit=b] [:[RAW][,UCL]]:DATA,PH=c [,TLP] [,helperunit=d];

All of the options have the same meaning as in the **DGN** command except that one phase must be specified in the initial **EX** command. Also, when the unit is a CU, the subunit must be specified. The diagnostic listing for the phase is necessary when using the interactive mode. Do not confuse statement numbers with test numbers. Refer to the "How to Clear Trouble Using the Diagnostic Listing" section.

An interactive diagnostic should be requested as described in the following procedure:

Step	Procedure
1	Enter the EX command. This command returns a slot number.
2	Wait for the display of the following: <pre>EX unit a [subunit b] STARTED AT STATEMENT 1</pre> which indicates that the interactive mode has started.
3	Execute the diagnostic by entering one of the commands listed in Table 6-3.

Table 6-3. EX Commands

Action	Command
To run to a specified point and then stop	In PDS, EX:PAUSE;unit a[,subunit b]:ST f! In MML, EX:PAUSE:unit=a[,subunit=b]:DATA,ST=f; The variable <i>f</i> is the statement number where the diagnostic is to pause prior to executing the statement. If the statement number is not reached, does not exist, or is less than or equal to the current statement, the diagnostic runs to the end of the phase.
To perform a specified group of tasks repeatedly	In PDS, EX:LOOP;unit a[,subunit b]:ST g-h! In MML, EX:LOOP:unit=a[,subunit=b]:DATA,ST=g&&h; The variable <i>g</i> is the starting statement (the first command in the loop) which must be greater than the current statement number. The variable <i>h</i> is the stopping statement (the first command <i>after</i> the loop). Note: Choose a loop where the initialization and deinitialization of variables and state machines are done correctly. Otherwise, false failures may result.
To stop the loop	In PDS, EX:STOP;unit a[,subunit b]! In MML, EX:STOP:unit=a[,subunit=b];

Table 6-3. EX Commands (Contd)

Action	Command
For example, to exercise main store PH95 and PH96 for the 3B20D computer	In PDS, EX:CU a,MASC b:PH 95! or EX:CU a,MASC b:PH 96!
	In MML, EX:CU=a,MASC=b:DATA,PH=95; or EX:CU=a,MASC=b:DATA,PH=96;
	In PDS, EX:LDPARM;CU a,MASC b:\ [SA i][,EA j][,PAT k][,REF m]\ [,FNUM n][,WPA p][,TIME q]]!
	In MML, EX:LDPARM:CU=a,MASC=b:\ [:DATA,[SA=i][EA=j][PAT=k]\ [,REF=m][,FNUM=n][,WPA=p][,TIME=q]];

Where:

- SA i** Start address. [Default = lowest address on Main Store Controller (MASC).]
- EA j** End address. (Default = highest address on MASC.)
- PAT k** Data pattern (Phase 95, 3B20D computer only). (Default = 0xFFFFFFFF.)
- REF m** Refresh rate. Normal rate is 2 ms, Extended rate is 4 ms. (Default = 4 ms.)
- FNUM n** Number of failures for which data is to be collected. (Default = 100.)
- WPA p** Number of words to be copied per memory array (Phase 96, 3B20D computer only). (Default = all words.) Defaulted on the 3B21D computer. Attempts to change on the 3B21D computer will cause errors.
- TIME q** Time (in seconds) that refreshes data parity checks will be active after memory is updated (Phase 96, 3B20D computer only). Default times assume the extended refresh rate. Refer to Table A-19 for default times for the 3B20D computer. Phase 96 does not use this option for the 3B21D computer. Defaulted on the 3B21D computer. Attempts to change on the 3B21D computer will cause errors.

Any omitted parameters will be replaced by default values.

Table 6-3. EX Commands (Contd)

Action	Command
To step through diagnostic commands	In PDS, EX:STEP;unit a[,subunit b]:ST f! In MML, EX:STEP:unit=a[,subunit=b]:DATA,ST=f; Same function as PAUSE but also displays data passed between diagnostics and the hardware unit driver program between the current statement and the statement number given <i>f</i>. This display occurs in CU diagnostics only.
To complete phase execution	In PDS, EX:PAUSE;unit a[,subunit b]:ST 1! In MML, EX:PAUSE:unit=a[,subunit=b]:DATA,ST=1;
To exit from the interactive mode	In PDS, [STOP STP]:DMQ:[unit a,subunit b][,ACTIVE WAITING]! In MML, [STOP STP]:DMQ:[unit=a,subunit=b][,ACTIVE WAITING];

How to Respond to a Denied Request

When a request is denied, the following message is output:

```
REPT MIRA QUEUE FULL - RETRY LATER
```

Reenter the request at a later time.

When an automatic request is denied, the calling process receives a fail return message; no action is required.

It is possible for a request to be continually denied. If this occurs, the following message is output:

```
REPT MIRA ERROR req unit a [subunit b] RETRY FAILED
```

or

```
REPT MIRA CANNOT RESERVE UNITS FOR req unit a [subunit b]
```

Where *req* is the type of request (RST, DGN, EX, and so forth).

Take the following actions:

Step	Procedure
1	Wait until the UCB is no longer reserved. The unit specified in the diagnostic request may be in use by another process. Both the INIT:MHD [initialize Moving Head Disk (MHD)] command and the UPD:GEN (generic update) command reserve units. Verify that these processes are not active. Retry the maintenance request. Continue to Step 2 if Step 1 fails.
2	Enter: In PDS, RMV:unit a ! In MML, RMV:unit=a ; Where <i>unit</i> is the unit name and <i>a</i> is the unit number from the original maintenance request. Retry the maintenance request. Continue to Step 3 if Step 2 fails.
3	To run the ECD audit, enter: In PDS, AUD:ECDOWN 1! In MML, AUD:ECDOWN=1; Retry the maintenance request. Continue to Step 4 if Step 3 fails.

Step	Procedure
4	Enter: In PDS, OP:STATUS:PROCESS,ALL! In MML, OP:STATUS:DATA,PROCESS,ALL; Record the identifications for all processes named <i>ppdiamon</i>, <i>cudiagc</i>, <i>iodiag</i>, <i>dfdiag</i>, and <i>inhtimer</i>. For each process identification, enter: In PDS, STOP:EXC;USER,PID a,SIG 9! In MML, STOP:EXC:DATA,USER,PID=a,SIG=9; Where a represents a process identification. To run the ECD audit, enter: In PDS, AUD:ECDOWN 1! In MML, AUD:ECDOWN=1; Retry the maintenance request. Continue to Step 5 if Step 4 fails.
5	Repeat Step 4 using the following commands. For each process identification from Step 4, enter: In PDS, STOP:EXC:ANY,PID a! In MML, STOP:EXC:DATA,ANY,PID=a; Where a represents a process identification. To run the ECD audit, enter: In PDS, AUD:ECDOWN 1! In MML, AUD:ECDOWN=1; Retry the maintenance request. Continue to Step 6 if Step 5 fails.

Step	Procedure
6	Enter: In PDS, OP:STATUS:PROCESS,ALL! In MML, OP:STATUS:DATA,PROCESS,ALL; Record the identification for the process <i>mira</i>. In PDS, STOP:EXC:ANY,PID a! In MML, STOP:EXC:DATA,ANY,PID=a; Where a represents a process identification. To run the ECD audit, enter: In PDS, AUD:ECDOWN 1! In MML, AUD:ECDOWN=1; Retry the maintenance request. Continue to Step 7 if Step 6 fails.
7	If problems persist, seek technical assistance. An active process has the UCB reserved.

How to Inhibit and Allow Automatic Maintenance Requests

Automatic maintenance requests must be inhibited while a field update is applied to the diagnostic files. Stop entering manual requests and perform the following procedure:

Step	Procedure
1	To inhibit automatic maintenance requests, enter the following: In PDS, INH:DMQ;SRC a[,TINH b][,AINH c]! In MML, INH:DMQ:SRC=a[,TINH=b][,AINH=c]; The variables have the following meanings: - a The source of the automatic maintenance requests to be denied. This may be either ADP, REX, the name of an application process, or ALL. ALL inhibits all automatic maintenance requests by changing the inhibited sources list to ALL. ALL will not inhibit SDLRT. - b The number of minutes the inhibit is in effect. The default is infinity. - c The number of minutes between warning messages informing the user that an inhibit is in effect. The default is 5 minutes.
2	Observe the following warning message: REPT DMQ INHIBIT a ACTIVE, REMAINING TIME n MIN or, for an infinite inhibit, REPT DMQ INHIBIT a ACTIVE.
3	The inhibit may be deactivated manually by entering the following commands: In PDS, ALW:DMQ;SRC a! In MML, ALW:DMQ:SRC=a; An entry of "ALL" allows all automatic maintenance requests by clearing the inhibited sources list.
4	Observe the following output message indicating that the inhibit is deactivated: ALW DMQ ENABLED a

Only automatic maintenance request sources may be inhibited. A request to inhibit a manual source is denied, and the following message is output:

```
INH DMQ CAN'T INHIBIT A MANUAL SOURCE - source
```

Also, only a limited number of inhibits may be active at one time. If this limit is reached, further inhibit requests result in the following message:

```
INH DMQ TOO MANY ACTIVE INHIBITS
```

In this case, another source must be allowed or the "ALL" source must be used in place of all currently active inhibits.

How to Obtain Status of Requests

To obtain information on the status of maintenance requests in the Diagnostic Maintenance Queue (DMQ), enter the following:

```
In PDS,  OP:DMQ!  
In MML,  OP:DMQ;
```

The output from this command indicates the task number (the number on the far left in Figure 6-1) and the queue assigned to a particular request.

OP DMQ

REQUEST ACTIVE

```
0  RST  DFC 1  BPH=1  EPH=100  
SOURCE = MAN          PRIORITY = 0  
STATUS = ACTIVE
```

REQUEST WAITING

```
1  RMV  CU 0  BPH=1  EPH=100  
SOURCE = MAN          PRIORITY = 0  
STATUS = WAITING
```

INHIBIT SOURCES

```
NONE
```

```
OP DMQ COMPLETED
```

Figure 6-1. Sample Output from OP:DMQ

How to Abort or Cancel a Request

It may be necessary to cancel a request in the waiting queue or to abort a request in the active queue if:

- The request was entered by mistake.
- A request of high importance is in the waiting queue and an active queue must be cleared to make room.
- An interactive diagnostic is to be executed.
- The active and waiting queues of all requests must be cleared for the field update of diagnostic files.

Step	Procedure
1	Enter: In PDS, OP:DMQ! In MML, OP:DMQ; The output from this command indicates the task number and queue assigned to a particular request. (Refer to Figure 6-1 for an example of this output.) The <i>SOURCE</i> of the maintenance request may be (but is not limited to) one of the following: ADP – Automatic Diagnostic Process. MAN – Manual requests input by the user. PSM – Power Switch Monitor. This request is either a remove request on a unit when the Request Out of Service (ROS) key on the unit power switch is activated or a restore request when the ROS key is released. PSM requests are classified as manual. REX – Routine Exercise.
2	To abort a request in the active queue or to cancel it from the waiting queue, enter: In PDS, [STOP STP]:DMQ:[unit a,subunit b][,ACTIVE WAITING]! In MML, [STOP STP]:DMQ:[unit=a,subunit=b][,ACTIVE WAITING]; The default action is to cancel both active and waiting jobs from the queue.

How to Remove and Restore Units

It is necessary to remove and restore units when applying a hardware change or when you wish to prevent automatic restoration of a unit you suspect may be faulty.

To remove a unit, enter:

In PDS, **RMV:unit a!**
In MML, **RMV:unit=a;**

To restore a unit, enter:

In PDS,
RST:unit a [[[:UCL][:CONT]] | [[[:RAW][:TLP][:CONT]]]]!
In MML,
RST:unit=a [[[:UCL][:DATA,CONT]] | [[[:RAW][:DATA,[TLP][:CONT]]]]];

When a unit is removed from service or restored to service, all units under it in the hierarchy are affected. If a diagnostic is available for a unit, it is executed before the unit is restored unless the **UCL** option is specified. The restore is stopped if the diagnostic fails.

The **TLP** option is available if there are diagnostic failures.

The **RAW** option is available for complete diagnostic results.

If the **CONT** option is specified, only the requested unit is restored. Units under the requested unit are not diagnosed or restored. (An entire CU is restored and removed at the same time; its subunits may not be handled separately.)

Subunits cannot be restored unless units above them in the hierarchy are in service.

How to Inform Diagnostic Program of Hardware Changes

Occasionally hardware Change Notices (CNs) that require a change in the diagnostic program are issued to the field. These CNs specify changes to the UCB that must be entered into the ECD. These changes should be implemented with the following procedure:

Step	Procedure
1	To remove the affected unit from service, enter: In PDS, RMV:unit a! In MML, RMV:unit=a;
2	If the affected unit is a CU, activate key 13 on the Emergency Action Page display. This deactivates any CU force that is in effect. (Refer to Chapter 12, "Craft Interface.")
3	Install any updates to the diagnostic program using standard field update procedures. (Refer to Chapter 13, "Field Update.")
4	To diagnose the affected unit and verify that it is ATP, enter: In PDS, DGN:unit a:TLP! In MML, DGN:unit=a:DATA,TLP;
5	Install the new hardware.
6	Update the UCB for the affected unit using the RC/V system. (Refer to 254-303-104, <i>Recent Change and Verify Manual</i> .) Four separate transactions are required. Each transaction requires entries to the <i>trbegin</i> , <i>ucb</i> , and <i>trend</i> pages. The transactions are the following: (a) Change the unit status from OOS or Unavailable (UNAV) to Unequipped (UNEQ). (b) Change the unit UCB fields to the values as the CN directs. (c) Change the unit status from UNEQ to GROW. (d) Change the unit status from GROW to OOS.
7	Diagnose the unit and verify that the change was installed correctly.

Step	Procedure
8	Activate the recent change to the UCB using the <i>activate</i> RC/V page.
9	To restore the unit to service, enter: In PDS, RST:unit a;UCL:TLP[,CONT]! In MML, RST:unit=a:DATA,TLP,CONT;
	 NOTE: If the change is applied to more than one unit, all steps should be completed for each unit before beginning work on the next unit.

How to Interpret Diagnostic Output and Clear Trouble

Diagnostic Output Message

If an automatic or manual diagnostic encounters trouble, a message is output. There are three basic scenarios for this procedure.

- When a unit is automatically removed from service, a major alarm is sounded and a message is output to the Maintenance Teletypewriter (MTTY) and Receive-Only Printer (ROP). A diagnostic is automatically requested by the ADP process with the **TLP** option specified. You may verify that a diagnostic is in progress by entering:

In PDS, **OP:DMQ!**

In MML, **OP:DMQ;**

If no diagnostic is in progress, begin one by entering:

In PDS, **RST:unit a:TLP!**

In MML, **RST:unit=a:DATA,TLP;**

A Trouble Locating Procedure (TLP) list will be produced if there are diagnostic failures.

- A diagnostic is automatically requested by the REX process with the **TLP** option specified. A TLP list is produced if there are diagnostic failures. At the completion of a REX, a summary table is printed. Any units that have failed a REX diagnostic request are marked STF.
- If the **TLP** option was specified in a manual diagnostic request, as is recommended, a TLP list is produced if there are diagnostic failures.

The failed diagnostic test data in the output messages and the TLP circuit pack list are your tools for locating and correcting trouble.

Output Message Format

The diagnostic output message consists of a header line followed by the failed test data (if failures occurred). (Refer to Figure 6-2.) The header line identifies the unit (and subunit for the CU), the phase number, and the diagnostic results as described in Table 6-4. If there are test failures and/or if all available tests are not run, the number of test failures precedes the test skipped reasons in parentheses on the header line. The test skipped reasons are 64 bits numbered from 0 to 63 starting from the right. When a bit is set, it indicates why some or all of the tests were not run. Refer to Table 6-5 for explanation of test skipped reasons.

If there are test failures, the header line is followed by a five-column failing test display. The TEST column identifies the failing test numbers in the phase. Unless the **RAW** option was specified on the diagnostic request, only the first five failing tests will be printed. The MASK column indicates which bits are actually included in the test. A one in the MASK column means that bit is part of the test. The ACTUAL column displays the results read from the hardware under test. The EXPECTED column displays the expected results for the read. The MISMATCH column indicates which bits failed the test.

The data in the MISMATCH column is derived from the ACTUAL, MASK, and EXPECTED data in the following manner. First, the ACTUAL and EXPECTED data are EXCLUSIVE OR'd, which yields a one in the bits that are not in agreement. Second, the results of the EXCLUSIVE OR are logically AND'd with the MASK to filter out disagreeing bits not included in the test. The ACTUAL, MASK, and EXPECTED data are not available for some 3B20D/3B21D computer peripheral unit diagnostics. When this is the case, "N/A" (Not Applicable) is printed for the ACTUAL, MASK, and EXPECTED data.

```

DGN  CU  1  MASC 0  PH 35  STF  ( 2 X'00000000 X'00000000 )

TEST      MISMATCH      ACTUAL      MASK      EXPECTED
   7      X'00100000      X'00100000      X'FFFFFFFF      X'00000000
   8      X'01000000      X'54555555      X'FFFFFFFF      X'55555555

DGN  CU 1  MASC 0  TERMINATED AT PH 35 STMNT 32 AFTER TEST 8

DGN  CU 1  MASC 0  COMPLETED  STF  ( 2 X'00000000 X'00000000 )

```

Figure 6-2. Sample Diagnostic Failure

Table 6-4. Diagnostic Results

Results	Explanation	Action to be Taken
ATP	All Tests Passed.	None.
CATP	Conditional All Tests Passed. Not all available tests were run, but those that were run passed. The test skipped reason bits are two 8-digit hexadecimal numbers that should be interpreted as a 64-bit binary number beginning with bit number 0 on the right. Bit numbers that are a 1 indicate a reason why tests were not run. The currently assigned test skipped reason bits are listed in Table 6-5.	Determine why tests were not run, and rerun the diagnostic.
NTR	No Tests Run. The test skipped reason bits are two 8-digit hexadecimal numbers that should be interpreted as a 64-bit binary number beginning with bit number 0 on the right. Bit numbers that are a 1 indicate a reason why tests were not run. The currently assigned test skipped reason bits are listed in Table 6-5.	Determine why tests were not run, and rerun the diagnostic. If the test skipped reason bits are all 0s, all available tests were run and no further action is necessary.
STF	Some Tests Failed and possibly not all available tests were run. The number of failing tests appears first, followed by test skipped information. The test skipped reason bits are two 8-digit hexadecimal numbers that should be interpreted as a 64-bit binary number beginning with bit number 0 on the right. Bit numbers that are a 1 indicate a reason why tests were not run. The currently assigned test skipped reason bits are listed in Table 6-5.	Determine why tests failed, and rerun the diagnostic. Refer to the "How to Interpret Diagnostic Output and Clear Trouble" section for the procedures. Determine why tests were not run, and rerun the diagnostic. If the test skipped reason bits are all 0s, all available tests were run.

Table 6-5. Explanation of Test Skipped Reasons

Unit	Bit Number	Reason
CU	0	Not used.
	1	Unit under test is forced active.
	2	Sanity timers are disabled.
	3	EAI forces on CU, disk, or sanity timer are active.
	4	Peripheral unit controller is OOS.
	5	Helper unit is not specified or invalid.
	6	Helper unit is unavailable.
	7	Unit under test is forced OOS.
	8	EAI initialization parameter buffer data pending, EAI must be cleared.
	9	Helper unit is not connected to subunit under test.
DFC, IOP, HSDC, MTC, MT	0	Helper unit is not specified or invalid.
	1	Helper unit cannot be reserved.
	2	Helper unit is not OOS.
	3	Helper unit is not ATP.
	4	Helper unit is not connected to controller being diagnosed.
	10	Data link equipage bits not properly set in ECD.
	11	The UN582 Multipurpose Peripheral Controller (MPC) diagnostic pumpcode does not currently support this phase.
	17	Helper IOP Magnetic Tape (MT) transport is not mounted and/or transport door open.
	18	Helper IOP MT transport is not in "on-line" mode.
	19	Helper IOP MT reel has no Write-Enable Ring (WER).
	20	Helper IOP MT is too long for diagnostic test tape.
	21	Helper IOP MT has no header block at Beginning of Tape (BOT) or header block indicates not diagnostic test tape.
	23	Phase cannot diagnose pack in growth unit.
	24	Helper unit is invalid (use the other CU as the helper unit).

Table 6-5. Explanation of Test Skipped Reasons (Contd)

Unit	Bit Number	Reason
DFC [Small Computer System Interface (SCSI) only], DFC 2, and DFC 3	32	First MHD on first Small Computer System Interface Bus (SBUS) of DFC did not respond correctly.
	33	Second MHD on first SBUS of DFC did not respond correctly.
	34	Third MHD on first SBUS of DFC did not respond correctly.
	35	Fourth MHD on first SBUS of DFC did not respond correctly.
	36	Reserved for fifth MHD on first SBUS of DFC.
	37	Reserved for sixth MHD on first SBUS of DFC.
	38	Reserved for seventh MHD on first SBUS of DFC.
	40	First MHD on second SBUS of DFC did not respond correctly.
	41	Second MHD on second SBUS of DFC did not respond correctly.
	42	Third MHD on second SBUS of DFC did not respond correctly.
	43	Fourth MHD on second SBUS of DFC did not respond correctly.
	44	Reserved for fifth MHD on second SBUS of DFC.
	45	Reserved for sixth MHD on second SBUS of DFC.
	46	Reserved for seventh MHD on second SBUS of DFC.

⇒ NOTE:

(For SCSI DFCs only): To interpret test skipped reason bits 32 through 46, the structure of the system must be understood. All device references are relative to the DFC under test and may not correspond to "real" unit numbers. The down pointer of the DFC will point to the first SBUS. The side pointer of the first SBUS points to the second SBUS. Similarly, the down pointer of an SBUS points to the first MHD beneath it. The side pointer of an MHD points to the next MHD in the chain. Currently, only four MHDs per SBUS are supported.

Main Store Failure Data

Failure data for faults detected in the main store may be output in the form of a histogram. Refer to the “Main Store Failure Data” section in Appendix A (for the 3B20D computer) or Appendix B (for the 3B21D computer) for a discussion and an example output of a main store failure.

How to Interpret Output from TLP Option

The **TLP** option produces a list of suspected faulty equipment when a diagnostic fails. An equipment list is output after the diagnostic has completed. Figure 6-3 is a sample TLP equipment list. The equipment most likely to be faulty is listed at the top, followed by other suspected equipment in descending order. For each entry, the following information may be given:

- Code Equipment code
- EQL Equipment location
- SD Schematic drawing number within the schematic drawing
- FS Functional schematic within the schematic drawing
- SYM Symbol number within the schematic drawing
- UNIT Unit in which the equipment resides (if not the unit under diagnosis)
- WT Weight, on a scale from one to ten, with the equipment most likely to be faulty having the highest number.
- Note If the note is nonzero, refer to Table 6-6 for further information.

Code	EQL	SD	FS	SYM	UNIT	WT	Note
TN61	12-128	4C052	1	6	—	10	0
TN70	12-160	4C052	1	2	—	10	0
TN62	12-144	4C052	1	5	—	10	0
TN62	12-136	4C052	1	5	—	10	0

Figure 6-3. Sample TLP Circuit Pack List

Table 6-6. Meaning of TLP Notes

Unit	Note	Explanation
CC	01	Do not remove power or equipment from an active or in-service unit.
	02	Cable to the maintenance channel of the CU.
	03	CU could be held stopped by the Micro Level Test Set (MLTS) or Bus Display Unit (BDU).
	04	If a "forced switch" of the CU is required, special precautions are required to ensure the appropriate initialization level of the out-of-service (OOS) CU being forced to the active state. An RTR level 4 boot is always required if the side to be forced active has been previously powered down. Seek technical assistance (3B Customer Technical Support) for diagnostic failure analysis and appropriate procedures for the force and boot of an out-of-service CU.
	05	MLTS, TN16, or UN16, if equipped.
	06	CU could be powered down.
	07	Power switch board or one of the power converters may be bad. If ALARM is lighted on the power switch board, power cycle the CU. If light remains on in one of the power converters, that converter may be bad. If lights go out on all converters and the CU fails to power up, then the power switch board may be bad. (3B20D Model 1 computer CU power switch circuitry includes EAI board, TN11.)
	10	Board is in IOP 0.
	11	Board is in IOP 1.
	50	Channel board in the channel 0 position.
	51	Channel board in the channel 1 position.
	52	Channel board in the channel 2 position.
	53	Channel board in the channel 3 position.
	54	Channel board in the channel 4 position.
	55	Channel board in the channel 5 position.
	56	Channel board in the channel 6 position.
	57	Board in the DMA 0 position (3B20D Model 1 computer – UN36).
	58	Board in the DMA 1 position (3B20D Model 1 computer – UN36).

Table 6-6. Meaning of TLP Notes (Contd)

Unit	Note	Explanation
CC	59	A channel address lead is being held active by CU, Channel (CH), or DMA.
	60	A CH or DMA may be responding and/or acknowledging falsely.
	61	A CH or DMA is holding channel error (ER bit 16) active low.
CH	01	Suspect any CH or DMA.
	02	Unterminated or defective cable on port under test.
	03	Suspect Helper Unit circuit packs. IOP and DFC circuit packs may be at fault. For the SCSI DFC, these circuit packs include the Host Adapter (HA) and the DFC.
	04	The CH under test is connected, via the DSCH cable, to another circuit pack. Suspect the DDSBS logic in that circuit pack or the DSCH cable.
CSU	01	Verify hardware for the kernel stack in Interrupt Stack Cache Memory feature corresponds with the UCB Hardware Version (HV) field of the Cache Storage Unit (CSU). A "1" in bit 0 of this field indicates the feature is present. The hardware for this feature is a UN45C Store Address Translator (SAT) with pins 149 and 047 wired together on the SAT backplane to enable the feature. Discrepancy between HV value and hardware should be corrected.
	04	If a "forced switch" of the CU is required, special precautions are required to ensure the appropriate initialization level of the out-of-service (OOS) CU being forced to the active state. An RTR level 4 boot is always required if the side to be forced active has been previously powered down. Seek technical assistance (3B Customer Technical Support) for diagnostic failure analysis and appropriate procedures for the force and boot of an out-of-service CU.
DCI	01	The DCI could be powered down. This board is located in the attached processor.
	02	The attached processor or its software may be the cause of this phase failing.

Table 6-6. Meaning of TLP Notes (Contd)

Unit	Note	Explanation
DFC	01	The DFC could be powered down.
	02	Do not remove power or equipment from an active or in-service unit.
DMA	01	Suspect any channel or DMA.
	04	If a “forced switch” of the CU is required, special precautions are required to ensure the appropriate initialization level of the out-of-service (OOS) CU being forced to the active state. An RTR level 4 boot is always required if the side to be forced active has been previously powered down. Seek technical assistance (3B Customer Technical Support) for diagnostic failure analysis and appropriate procedures for the force and boot of an out-of-service CU.
DUIC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
	02	Replacement equipment must have same microcode number.
IOP	01	The IOP could be powered down.
	02	Do not remove power or equipment from an active or in-service unit.
MASC	01	Do not remove power or equipment from an active or in-service unit.
	04	If a “forced switch” of the CU is required, special precautions are required to ensure the appropriate initialization level of the out-of-service (OOS) CU being forced to the active state. An RTR level 4 boot is always required if the side to be forced active has been previously powered down. Seek technical assistance (3B Customer Technical Support) for diagnostic failure analysis and appropriate procedures for the force and boot of an out-of-service CU.
	09	Array identified by DGN CU_ MASC_ PH_ MEMORY FAILURE RESULTS output message.
	10	Additional suspected units on the store bus are CC, other MASC in CU under test (in CUs with two MASCs), DMA(s), CSU (if equipped), and SAT.

Table 6-6. Meaning of TLP Notes (Contd)

Unit	Note	Explanation
MASC	12	Any array in MASC.
	13	Suspect other MASC (in CUs with two UN39-UN40 MASCs) or any one of its arrays.
MHD	01	Part of the DFC must be removed from service and powered down before this equipment can be replaced.
	02	MHD_2GB means that the under-test disk drive must be replaced with another drive that has a capacity of greater than or equal to 2 GB (gigabytes).
	09	Refer to the <i>Maintenance Instruction Manual</i> for KS-22707, L2 memory.
	10	These circuit cards are located in the logic card cage of the KS-22707, L2 memory unit. Also refer to MHD Note 09.
	11	The power amplifier, VP58, is located behind the logic card cage. Also refer to MHD Note 09.
	12	Codes for all circuit cards in the KS-22707, L1 memory are a four-character string of alphanumeric or alpha characters. The first character in the string determines the revision level, and the last three characters determine the circuit card code. Identification of the circuit card, therefore, is based on the last three characters.
	13	Refer to MHD Note 12, then proceed. These circuit cards are located in the logic chassis. Refer to the <i>Maintenance Instruction Manual</i> for KS-22707, L1 memory.
	14	Refer to MHD Note 12, then proceed. These circuit cards are located on the deck plate of the memory unit. Refer to the <i>Maintenance Instruction Manual</i> for KS-22707, L1 memory.
	16	The MHD could be spun down.
	17	The MHD could be powered down. Check power inverter for alarm or no power. Check MHD power cord, power switch, and power indicator.
	18	An incorrect ID setting or incorrect equipage value in the MHD UCB could be the cause of this failure.

Table 6-6. Meaning of TLP Notes (Contd)

Unit	Note	Explanation
MTC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
	02	The indicated faulty equipment is in the transport.
MTTYC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
SCSDC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
SDLC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
TTYC	01	Part of the IOP controller. The IOP must be removed from service and powered down before this equipment can be replaced.
UC	01	Field Test Set (FTS) may have control of the Utility Circuit (UC).



NOTE:

MHD Note 02 refers to the 2-Gigabyte Volume Table of Contents (2GB VTOC) feature that provides the capability to generate and use a 2GB VTOC. Users may increase the number of allocated blocks on their MHDs to have a capacity up to 2 GB.

How to Interpret TLP Equipment Codes

The TLP prints the minimum equipment codes/circuit pack codes required for the features that are in use. These features are determined by elements of the database. The TLP does not print the latest available letter suffix circuit pack codes, even though they may be available and in use. Refer to the "Suffix Coding of Circuit Packs" section in Chapter 1.

Circuit packs with later letter suffix codes may be used as replacements for circuit pack codes identified by the TLP. Circuit packs with earlier letter suffix codes cannot be used as replacements for circuit pack codes identified by the TLP. For example, if the TLP identifies a UN133C, a UN133C or UN133D may be used to replace it. However, a UN133 or a UN133B cannot be used to replace the required UN133C. (The "A" suffix, such as in UN133A, is never used.)

How to Clear Trouble Using the TLP Circuit Pack List

Using the TLP circuit pack list as a reference, perform the following procedure:

Step	Procedure
1	Go to the supply of spare circuit packs and obtain one of each pack code mentioned on the list.
2	Go to the failing unit and set the ROS/RST key to the ROS position. The Request In Progress (RQIP) lamp will light. Wait for it to extinguish and for the OOS lamp to light (if it was not already lighted).
3	Press the OFF key to remove the power.
4	Remove the first circuit pack listed on the TLP list and insert the spare pack in its place. Determine its location from the Equipment Location (EQL) specified.
5	Press the ON key to restore the power.
6	Set the ROS/RST key to the Restore (RST) position. The RQIP lamp will light to indicate that a diagnostic of the unit is in progress. The RQIP lamp will stay lighted until the diagnostic completes or, if the diagnostic passes, a restore is completed.
7	If the OOS lamp goes out when the RQIP lamp does, the trouble has been cleared and the unit has been restored. You should suspect that the last pack you removed is bad. If the OOS lamp stays on, repeat Steps 2 through 6 for each subsequent pack on the list.

Occasionally, a problem may be cleared simply by reseating a circuit pack. To check this, use the following procedure to reseat a pack that you have determined to be bad. This procedure may be performed sometime after replacing the circuit pack.

Step	Procedure
1	Go to the failing unit and set the ROS/RST key to the ROS position. The RQIP lamp will light. Wait for it to extinguish and for the OOS LED to light (if it was not already lighted).
2	Press the OFF key to remove the power.
3	Remove the new circuit pack and reinsert the suspected bad pack into the EQL from which it came.
4	Press the ON key to restore power.
5	Set the ROS/RST key to the RST position. The RQIP lamp will light to indicate that a diagnostic on the unit is in progress. The RQIP lamp will stay lighted until the diagnostic completes or, if the diagnostic passes, a restore is completed.
6	If the ROS lamp goes out when the RQIP lamp does, the unit has been restored. The suspected bad pack was not bad, and the trouble was cleared by reseating the pack. Make note of the EQL where the trouble occurred.



CAUTION:

If the trouble recurs at this EQL, the pack may be intermittently faulty; replace it even though the retest passes.

If the ROS lamp stays on, the trouble has returned with the reseating. This indicates that the pack was defective and should be replaced. Repeat Steps 1 through 5 to remove the bad pack, and reinsert the good pack.

How to Clear Trouble Using the Diagnostic Listing

If the problem still exists after using the TLP list, you must analyze the diagnostic test results. This is done using the diagnostic output message and the diagnostic listings provided on either paper or microfiche with your equipment. Generally, the first failing phase and the first few failing tests within that phase are very useful for analysis. If more data is required, rerun the diagnostic using the **RAW** option to print all the test failures.

A diagnostic listing consists of a prologue followed by one or more program units. Each program unit has a prologue that gives information about what is tested, how the testing is done, and the hardware involved. Lines that begin with "C" (capital C) are the comment lines. The remainder of the program unit is diagnostic commands. Each diagnostic command has a statement number. Some commands generate tests and appear as shown in Figure 6-4. The test numbers in the listing correspond to the test numbers in the diagnostic output messages.

```
C
C test that cache tag mod B bit 14 (address parity check
C bit) and bit 15 (data parity check bit) were
C properly generated by the parity generated
C
    csread adr=TAGAB,0x400 exp=0x80000000 mask=0xc0000000
CC Clear the SCR
25    _csread_wrtmiis data=0xffffffff
        dst=10 clevel=_csread
        e    adr 00c0 data 08 0e fd4affff ff ff 55
CC Clear the SDR.
26    _csread_wrtmiis data=0
        dst=12 clevel=_csread
        *    adr 00c9 data 08 0e fd4c0000 00 00 55
CC Load the address into the SAR.
27    _csread_wrtmiis data=0x0000f000
        dst=8 clevel=_csread
        *    adr 00d2 data 08 0e fd4800fd f4 00 55
CC Set access enable.
28    _csread_wrtmiis data=0x7fffffff
        dst=10 clevel=_csread
        *    adr 00db data 08 0e fd4a7fee ff ff 55
CC Read the SDR (contains the cache reply).
29    _csread_ccread sp=12
        exp=0x80000000 mask=0xc0000000
        *****
        *test 002 adr 00e4 data 0b 0c 05 0c 80000000 c0000000
        *****
30    endblk
        *    adr 00f0 data 01 0a
31    bgnblk
        *    adr 00f2 data 01 09
```

Figure 6-4. Sample Diagnostic Listing

Perform the following procedure to clear trouble:

Step	Procedure
1	Examine the diagnostic output messages to determine which phases failed.
2	Obtain the diagnostic listing and read the prologues for the phase and program unit in which the failing tests occur. Pay particular attention to the sections labeled troubleshooting hints.
3	Find the diagnostic commands associated with the failed tests by checking the test numbers.
4	Read the comments (lines beginning with a "C") on the page that precedes the command list to gain insight as to where the problem lies.
5	If you are unable to determine how to proceed to clear the trouble, contact next level of technical support.

⇒ NOTE:

Certain diagnostic phases may generate a pulse on the backplane of the CU under test when failures occur. At present, this feature is available only for tests that use the CU main store mode of execution.

How to Respond to a Diagnostic Audit Message

At various points within the diagnostic control structure, audits are performed to verify that all is well. These include verification of the following:

- Called functions do not give bad return codes
- Needed system resources are available
- Necessary files can be opened and read or executed
- Hardware errors have not occurred
- Illegal operations are not attempted.

If an audit fails, a report is printed on the maintenance terminals. Respond to an audit report as detailed in the following procedure:

Step	Procedure
1	If a test fails prior to an audit failure, clear the problem indicated by the test failure. This may also clear the audit failure.
2	Refer to 254-303-111, <i>PDS Output Messages Manual</i> , or 254-303-113, <i>MML Output Messages Manual</i> , to determine the reason for the audit failure. It may be because of a situation you can control. For example, an attempt to execute a nonexistent diagnostic phase would cause an audit failure.
3	Save the printout pertaining to the diagnostic request.
4	Refer to 254-303-111, <i>PDS Output Messages Manual</i> , or 254-303-113, <i>MML Output Messages Manual</i> , to see if any additional data should be collected and forwarded to your technical assistance organization.

The output messages and error codes for diagnostic audits are described in the following paragraphs.

Audit Failure Within the DIAMON: An audit failure within the DIAMON process appears as follows:

```
REPT DIAMON  ERROR = a  ERRNO = b
```

The number following the "ERROR =" is one of the error codes given in Table 6-7. The meaning of "ERROR =" is defined in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The number following the "ERRNO =" is a system error number. The meaning of "ERRNO =" is defined in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

A sample output from a DIAMON audit will appear as follows:

```
REPT  DIAMON  ERROR =X'74  ERRNO =43
```

Table 6-7. DIAMON Error Codes

Error Code (Note)	Meaning
X'41	Manual request to abort the DIAMON process.
X'42	Cannot send event to Diagnostic Control Process (DIAGC).
X'43	Cannot find the TLP process pathname in the utility file.
X'44	Cannot spawn the TLP process.
X'45	Cannot receive secondary message.
X'46	Cannot make segment.
X'47	Unable to enable message receive function.
X'48	Message receive function failed.
X'49	Teletypewriter (TTY) name passed from MIRA invalid.
X'4A	Invalid spooler tag; check file.fmat.
X'4B	Kill of DIAGC failed.
X'4C	Invalid request received from MIRA.
X'4D	Unable to find the remove process in the utility file.
X'4E	Unable to find the restore process in the utility file.
X'4F	Attempted to spawn the remove process but failed.
X'50	Attempted to spawn the restore process but failed.
X'51	Invalid return tag from the remove process.
X'52	Invalid return tag from the restore process.
X'53	Cannot send remove process a message.
X'54	Cannot send restore process a message.
X'55	Invalid request received from user.
X'56	An attempt to restore a unit that is not a restorable unit.
X'57	Cannot open utility file.
X'58	Cannot release UCB from ECD manager.
X'59	Cannot send the result of the diagnostic to MIRA.
X'5A	Cannot find a match on the tag in the utility file.

Note: X indicates PDS format, and H indicates MML format.

Table 6-7. DIAMON Error Codes (Contd)

Error Code (Note)	Meaning
X'62	Improper use of repeat counter and RPT option.
X'63	Unrecognized child process.
X'64	DIAGC was running on death of child signal.
X'65	No process running on death of child signal.
X'66	Illegal format on tag in the utility file.
X'67	Could not remove specified unit/subunit.
X'68	Could not restore specified unit/subunit.
X'69	No TLP for the unit requested.
X'6A	No diagnostic for the unit requested.
X'6B	ECD manager failed to perform the UWUTIL function.
X'6C	ECD failed to release special device file.
X'6D	DIAMON unable to open message format file.
X'6E	DIAMON unable to open diagnostic phase table.
X'6F	DIAMON unable to open the ECD manager.
X'70	Invalid equipment identification on input request.
X'71	Unable to get next UCB from ECD manager.
X'72	Unable to read the diagnostic phase table.
X'73	DIAMON unable to spawn DIAGC (lcall failed).
X'74	Illegal phase specified on input request.
X'75	UCB is reserved by another maintenance client.
X'76	ECD had trouble with the special device file.
X'77	Invalid secondary message received from MIRA or DIAGC.
X'78	Invalid primary message received from MIRA.
X'79	Return code from DIAGC invalid (dcb->d_prslt).
X'7A	Return code from a unit diagnostic invalid.
X'7B	ECDMAN failed to perform the UWWHYRM function.
X'7C	User attempt to run diagnostics on an IOP tape unit.
X'7D	User attempt to diagnose an invalid 3B21D computer CU subunit.

Note: X indicates PDS format, and H indicates MML format.

Audit Failure Within a CU Diagnostic: An example of an audit failure within a CU diagnostic is shown in Figure 6-5. Explanations of the error codes are given in Table 6-8. The meanings of the variables are as follows:

X = The position within the Maintenance Channel (MCH) buffer of the MCH command being executed.

ERRNO = A system error code reported by the MCH driver (PCPMD) from Table 6-9.

IL = The left half of the current MCH command.

IR = The right half of the current MCH command.

ONL = Active CU MCH master status register. (Refer to Table 6-10 for the meaning of each bit set.)

OFL = Under test CU MCH slave status register. (Refer to Table 6-10 for the meaning of each bit set.)

T = The number of the last diagnostic test that was executed.

A = The current offset into the data table file.

S = The number of the currently executing diagnostic statement.

I = The index (opcode) of the diagnostic command currently executing.

The off-line critical registers may or may not be displayed according to the type of audit failure.

```
REPT DGN CU          ERROR X'45
DRIVER DATA
X = 0  ERRNO = 0  IL = X'FF40FFFF  IR = X'DFFF550D
MCH STATUS
ONL = X'00000000  OFL = X'00000000
PHASE DATA
T = 2  A = X'007F  S = 16  I = X'17
```

Figure 6-5. Sample Output from CU DGN Audit

Table 6-8. DGN CU Error Codes

Error Code	Meaning
X'41:	Test count specified in phase does not match tests executed.
X'42:	Microcode execution timed out in CU under test.
X'43:	Main store execution setarg error in CU under test.
X'44:	Main store execution timed out in CU under test.
X'45:	Microcode pump failed in CU under test.
X'46:	Illegal object file for main store execution in CU under test.
X'47:	Main store pump failed in CU under test.
X'48:	Cannot set register for main store execution in CU under test.
X'49:	Cannot set up stack for main store execution in CU under test.
X'4A:	Main store execution problem in CU under test.
X'4B:	Execute on-line (exconl) failure.
X'4C:	Cannot open ECD manager devecd or devdgnecd.
X'4D:	Illegal or unspecified helper unit requested.
X'4E:	UCB data invalid.
X'4F:	Psetup failed to pass topucb ptr to pcpmd driver.
X'50:	Incorrect input parameters in interactive mode.
X'51:	Cannot get UCB.
X'52:	Reserve UCB failed.
X'53:	Release UCB failed.
X'54:	Cannot switch to helper unit UCB.

Table 6-8. DGN CU Error Codes (Contd)

Error Code	Meaning
X'56:	Signal from operating system – illegal instruction (SIGILL).
X'57:	Signal from operating system – bus error (SIGBUS).
X'58:	Signal from operating system – segment violation (SEGVOL).
X'59:	Signal from operating system – error on system call (SIGSYS).
X'5A:	ECD manager cannot get special device file pathname.
X'5B:	Failed to open special device file, read mode.
X'5C:	Failed to release special device file.
X'5D:	Failed to close special device file, read mode.
X'61:	Open PCPMD failed.
X'62:	Write to PCPMD failed.
X'63:	Normal completion of a phase.
X'64:	Cannot open data table file.
X'65:	Cannot share the DCB from DIAMON (connseg failed).
X'66:	Error in data table.
X'67:	Invalid number of parameters from DIAMON.
X'68:	Open function failed open TTY device.
X'69:	Cannot open dgn format file.
X'6A:	Invalid command received while in interactive mode.
X'6B:	Cannot read register in CU under test.
X'6C:	Cannot open TLP file—TLP option is denied.
X'6D:	Maintenance restart function occurred in CU under test.
X'6E:	Hardware platform/instruction mismatch.

Table 6-9. Selected System Error Codes

Error Code	Meaning
0:	Not a system error.
60:	Invalid emt value for this driver.
61:	IO map error.
62:	Hardware is not OOS or reserved.
63:	Not pseudo unit.
64:	Pointer is not valid.
65:	No mate unit.
66:	Not a legitimate start UCB.
67:	Bad equipment identification entry.
68:	Bad search level entry.
69:	Could not find eqid unit.
70:	Unit is not OOS.
71:	Unit is already reserved.
72:	No idle minor device chain table entry.
73:	Backup is not available.
74:	UCB type incorrect.
75:	Not a CU UCB.
76:	Invalid MCH order.
77:	PCPMD driver busy—try again later.
78:	Micro-assist code in off-line/slave failed.
79:	Pointer is not pointing to a UCB.
80:	Invalid PCPMD driver command.
81:	Cannot fault the driver.
82:	Configuration control fail return.
83:	Processor is not duplexed.
84:	Mate processor is not active.
85:	Invalid argument.
86:	MCH hardware error.

Table 6-9. Selected System Error Codes (Contd)

Error Code	Meaning
87:	Execmch function did not process expected number of bytes.
88:	Ptimer kernel function returned fail.
89:	Time-out on reading off-line.
90:	Parity in on-line processor.
91:	Parity failure in off-line processor.
92:	Data is not same in both processors.
93:	The on-line master processor was specified.
94:	Stop DMA routine failed.
95:	Start DMA routine failed.
96:	Unequipped UCB specified or accessed.
97:	Not a top UCB.
98:	No match found.
99:	Bad tag value.
100:	Off-line CU MRFed.
120:	lla_nsqid error.
121:	lla_rdget error.
122:	lla_uprid error.
123:	lla_opnsq error.
124:	lla_attrid error.
125:	lla_delete error.
126:	lla_sinfo error.
127:	lla_get error.
128:	lla_gtrid error.
129:	Update bit is not asserted in the SSR.
130:	The on-line CU RID in error.
131:	No DMA ucl(s) under the CU.
132:	My store parity error detected during memory update.
133:	Other store time-out error during memory update.

Table 6-10. Maintenance Channel Status

MCH Master Status Register (Active CU)	
Bit	Meaning
0	MCH is idle.
1	Slave received bad start code.
2	Slave received invalid command.
3	Slave error.
4	Master received bad return code.
5	Master given invalid command.
6	Master error.
7	Panel interrupt.

MCH Slave Status Register (CU Under Test)	
Bit	Meaning
0	Received bad start code.
1	Received invalid command.
2	Received bad command parity.
3	CC is on-line and is not disabled.
4	CC is not stopped.
5	Extra start code [did receive start code after LOAD Microstore Data (MSD) bus].
6	Slave error.
7	Received MCHL interrupt (stop and switch).

Audit Failure Within a Non-CU Diagnostic: An audit failure within an IOP, DFC, or DCI diagnostic appears as shown in Figure 6-6. The variables have the same meaning as in a CU audit (Figure 6-5) except for the following:

R = An error code from Table 6-11.

D = If DRIVER DATA, an error code returned from the IOP, DFC, or DCI driver, or if SYSTEM DATA, a system error code.

REPT DGN AUDIT IOP	—	<i>Release 6.3 and later</i>
DGN AUDIT IOP	—	<i>Generic 2 through Release 6.2</i>
R = X'41		
DRIVER DATA		
D = X'2		
PHASE DATA		
T = 0 A = X'FFFFFFFF S = 0 I = X'00		
REPT DGN AUDIT DFC	—	<i>Release 6.3 and later</i>
DGN AUDIT DFC	—	<i>Generic 2 through Release 6.2</i>
R = X'41		
SYSTEM DATA		
D = X'2		
PHASE DATA		
T = 0 A = X'FFFFFFFF S = 0 I = X'00		

Figure 6-6. Sample Output from Non-DGN CU Audit

Table 6-11. DGN Audit IOP, DFC, and DCI Error Codes

Error Code	Meaning
X'41:	Open special device file failed.
X'42:	SETIO failed.
X'43:	Make seg failed.
X'44:	Msgport failed, d = reason.
X'45:	Read peripheral controller pump file failed.
X'46:	Driver operation failed.
X'47:	Open peripheral controller pump file failed.
X'48:	Magenab failure.
X'49:	Msgdisab failure.
X'4A:	Open PCPMD failed.
X'4B:	Write to PCPMD failed.
X'4C:	Excofl timed out.
X'4D:	Open ect manager failed.
X'4E:	Function ugucbn failed.
X'4F:	Peripheral interface controller microstore size not in UCB.
X'50:	Receive wait failed.
X'51:	Lexec function failed.
X'52:	E_host of UCB not initialized.
X'53:	Lseek function failure.
X'54:	MHD drive size is not 5 or 19 tracks.
X'55:	Makeseg failed for disk write/read buffer.
X'56:	Disk write/read segment is too small.
X'57:	Function ugetucb failed.
X'58:	DFC is not active.
X'59:	IOP is not active.
X'61:	SCSI bus is not active.
X'62:	Incorrect flash ram file count.
X'63:	Feature not supported.

Table 6-11. DGN Audit IOP, DFC, and DCI Error Codes (Contd)

Error Code	Meaning
X'64:	Cannot open data table file.
X'65:	Cannot share DCB from DIAMON.
X'66:	Error in data table.
X'67:	Invalid number of parameters from DIAMON.
X'68:	Open spooler failed.
X'69:	Invalid equipage field value.
X'6A:	Invalid interactive mode command.
X'6C:	Open tlpfile failed.
X'6D:	Signal: illegal instruction.
X'6E:	Signal: bus error.
X'6F:	Signal: segment violation.
X'70:	Signal: software termination.
X'71:	Function ugnxtucb failed.
X'72:	Illegal device type for DFC.
X'73:	Function ucbrsrv failed.
X'74:	Function ugsdf failed.
X'75:	Function diskinfo failed.
X'77:	Specified file is not accessible.
X'78:	Could not create output file.
X'79:	Flash RAM data file wrong size
X'7A:	Could not write output file.

Synchronous Data Link Restore Tool

The Synchronous Data Link Restore Tool (SDLRT) is a *UNIX* RTR operating system release tool designed to continue the restoration attempts after the ADP has reached its restore threshold on the synchronous data link units and controllers. It consists of two *UNIX* system level processes with two executable files, */etc/sdlrtc* and */etc/sdlrtn*, and two ULARP records, keyed SDLRTC and SDLRTN. If for some reason these two *UNIX* system processes die, ULARP will restart them.

The purpose of this tool is to eliminate the need for maintenance personnel to restore the synchronous data link units and controllers manually when they are removed from service. This condition is caused if dial-up links are disconnected or modems (remote systems, and so forth) are powered down.

The SDLRTC process, executed by ULARP, periodically checks the status of Synchronous Data Link (SDL) or High-Speed Data Link (HSD) units and their Synchronous Data Link Controller (SDLC) or High-Speed Data Link Controller (HSDC) (under certain criteria) associated with the Communications Protocol Handler (CPH). The process restores those units/controllers that are OOS and have been fault removed by ADP. Inhibits do not block SDLRT.

The SDLRTN process, executed by ULARP, periodically checks the status of SDLs/HSDs and SDLCs/HSDCs (under certain criteria) associated with the Network Protocol Handler (NPH) and restores those units/controllers that are OOS and have been fault removed by ADP.

In *UNIX* RTR Operating System Releases 1 through 6.5, the SDLRTC and SDLRTN processes check only those synchronous data link units that reside on TN75/UN582 (SDLC) boards that have passed diagnostics (that is, ATP) and whose *carrier* field in the CPBLX3/CPPHNONE/NPHOPT record in the ECD is set to s for switched. However, in Release 6.6 and later, SDLRTC and SDLRTN processes also check those synchronous data link units that reside on TN82/UN582 (HSDC) boards that have passed diagnostics. The Switching Control Center (SCC) data link units that reside on TN[9]83/UN583 (MTTYC) boards are not affected by these two processes.

ULARP initially starts the SDLRTC and SDLRTN processes with three process arguments in Releases 1 through 6.5 and five process arguments in Release 6.6 and later. If no arguments are supplied, the default values will be used.

The first argument specifies the interval (in seconds) between two successive checks. The default value is 300 (that is, every 5 minutes).

The second argument specifies the inhibit/allow flag for printing to the ROP the following maintenance class restore request messages (OMDB key is 586):

REPT SDLRTC RESTORE MESSAGE FOR *b c* SENT TO MIRA

REPT SDLRTN RESTORE MESSAGE FOR *b c* SENT TO MIRA

where *b* is the name of the synchronous data link unit or controller and *c* is its number.

The valid values are 1 for inhibit and 0 for allow. The default value for this argument is 1, that is, this message will not be printed to the ROP.

The third argument specifies the inhibit/allow flag for printing to the ROP the following maintenance class complete restore messages (OMDB key is 586):

REPT SDLRTC COMPLETED RESTORE OF SDLS

REPT SDLRTN COMPLETED RESTORE OF SDLS

The valid values are 1 for inhibit and 0 for allow. The default value for this argument is 1, that is, this message will not be printed to the ROP.

The fourth argument (Release 6.6 and later) specifies the tool's coverage regarding ECD settings. The default value, A, and other valid values are described as follows:

Option	Meaning
A or a	(All) No restriction as far as ECD setting is concerned. The <i>carrier</i> field in the CPBLX3/CPPHNONE/NPHOPT record and the <i>h_type</i> field in the Unit Control Block (UCB) record are ignored and the unit/controller is restored if it has been fault removed.
C or c	(Continuous) Restore only the data link units whose <i>carrier</i> field in the CPBLX3/CPPHNONE/NPHOPT record is set to <i>c</i> .
S or s	(Switched) Restore only the data link units whose <i>carrier</i> field in the CPBLX3/CPPHNONE/NPHOPT record is set to <i>s</i> .
U or u	(UCB specified) Restore only the data link units/controllers whose <i>h_type</i> field in the UCB record has the Automatic Delayed Restoration (AUTODRST) bit set, that is, bit 2 is 1 for the AUTODRST feature supported by CPH. Since CPH uses this field, SDLRTC does not look at it. This argument value should only be used for NPH links.

The fifth argument (Release 6.6 and later) specifies the tool's coverage regarding the type of boards where the data link controllers reside. The default value, B, and other valid values are as follows:

Option	Meaning
B or b	(Both Low- and High-speed) TN75, TN82, or UN582 boards.
L or l	(Low-speed) TN75/UN582 boards only.
H or h	(High-speed) TN82/UN582 boards only.

The setting of the *proc_args* field in the ULARP records for SDLRTC and SDLRTN provided by *UNIX* RTR operating system is *300 1 1* in Releases 1 through 6.5 and *300 1 1 S L* in Release 6.6 and later.

This setting is updatable by the applications. For example, applications using Release 6.6 and later that want to restore all units/controllers that have been fault removed must update their setting from *300 1 1 S L* to *300 1 1 A B*. After the changes are made to the ULARP record(s), the SDLRTC and/or SDLRTN process(s) must be restarted for the new setting to become effective.

To do this, applications must enter the following input message(s) for the particular executable file to cause ULARP to restart the particular processes.

In PDS, **STOP:EXC:ANY, FN "/etc/sdlrtc", UCL!**
STOP:EXC:ANY, FN "/etc/sdlrtn", UCL!

In MML, **STOP:EXC:DATA, ANY, FN="/etc/sdlrtc", UCL;**
STOP:EXC:DATA, ANY, FN="/etc/sdlrtn", UCL;

To disable SDLRTC, SDLRTN, or both processes, applications must remove the particular ULARP record(s) and enter the appropriate input message(s) previously described.

Locating Pins

Pins are grouped according to sets. A set consists of 6 columns and 50 rows for a UN-type circuit pack, 4 columns and 50 rows for a TN-type circuit pack, 6 columns and 100 rows for a KLV-type circuit pack, and 8 columns and 50 rows for a KBN-type circuit pack. The columns are labeled 0 through 5 from **right to left**; the rows are labeled 0 through 24, then 32 through 56, from **bottom to top**.

A pin number for a pin in a given set is YZZ, where:

Y = The column number in which the pin is located

ZZ = The row number in which the pin is located.

The location number for a set is WWW-XXX, where:

WWW = The set location in number of inches from the floor

XXX = The number of the column that is in the 0 position for that set.

Refer to Figure 6-7 to see how a set looks. In Figure 6-7, pin number 315 is darkened.

0	0	0	0	0	0	56
0	0	0	0	0	0	55
0	0	0	0	0	0	54
0	0	0	0	0	0	53
0	0	0	0	0	0	52
0	0	0	0	0	0	51
0	0	0	0	0	0	50
0	0	0	0	0	0	49
0	0	0	0	0	0	48
0	0	0	0	0	0	47
0	0	0	0	0	0	46
0	0	0	0	0	0	45
0	0	0	0	0	0	44
0	0	0	0	0	0	43
0	0	0	0	0	0	42
0	0	0	0	0	0	41
0	0	0	0	0	0	40
0	0	0	0	0	0	39
0	0	0	0	0	0	38
0	0	0	0	0	0	37
0	0	0	0	0	0	36
0	0	0	0	0	0	35
0	0	0	0	0	0	34
0	0	0	0	0	0	33
0	0	0	0	0	0	32
0	0	0	0	0	0	24
0	0	0	0	0	0	23
0	0	0	0	0	0	22
0	0	0	0	0	0	21
0	0	0	0	0	0	20
0	0	0	0	0	0	19
0	0	0	0	0	0	18
0	0	0	0	0	0	17
0	0	0	0	0	0	16
0	0	•	0	0	0	15
0	0	0	0	0	0	14
0	0	0	0	0	0	13
0	0	0	0	0	0	12
0	0	0	0	0	0	11
0	0	0	0	0	0	10
0	0	0	0	0	0	09
0	0	0	0	0	0	08
0	0	0	0	0	0	07
0	0	0	0	0	0	06
0	0	0	0	0	0	05
0	0	0	0	0	0	04
0	0	0	0	0	0	03
0	0	0	0	0	0	02
0	0	0	0	0	0	01
0	0	0	0	0	0	00
5	4	3	2	1	0	

Figure 6-7. Backplane Pin Set

Processor Recovery Messages

This section covers Processor Recovery Messages (PRMs) from CU Central Control (CC) EAI diagnostics. The CU under test produces PRMs while executing the diagnostics associated with the EAI hardware. The PRMs output are different for the 3B20D computer and the 3B21D computer.

- For the 3B20D computer, refer to the “Processor Recovery Messages” section of Appendix A.
- For the 3B21D computer, refer to the “Processor Recovery Messages” section of Appendix B.

Contents

Introduction	7-1
■ System Configuration	7-1
■ Side Switch Operation	7-2
■ End Operation	7-2
Using OFLBOOT	7-3
Performing OFLBOOT	7-4
■ How to Start OFLBOOT	7-5
Using the EXC Command	7-5
Stepping Through the EXC Command	7-10
■ How to Stop OFLBOOT	7-12
Using the STOP Command	7-12
Stepping Through the STOP Command	7-13
■ How to Side Switch During OFLBOOT	7-13
Using the SW Command	7-15
Stepping Through the SW Command	7-15
Checking OFLBOOT Results	7-17
■ Available Information During OFLBOOT	7-17
■ Available Information After OFLBOOT	7-17
Examples of Performing OFLBOOT	7-20
■ Performing OFLBOOT Without IOP Selected	7-20
■ Performing OFLBOOT with One IOP Selected	7-23

Introduction

This chapter covers the Off-Line Boot (OFLBOOT) procedure for the *UNIX*® Real-Time Reliable (RTR) Operating System Release 21.1 and later. OFLBOOT is used primarily to verify the capability to boot essential files without interrupting the system operation. OFLBOOT can be used by applications as a prelude to a software release transition. If the feature is supported by a specific application, refer to the application documents for instructions.



CAUTION:

During this procedure, the system is split into two simplex configurations, thus making it vulnerable to errors. Therefore, OFLBOOT should be used during low traffic hours and with constant monitoring.

System Configuration

During this procedure, the system is split into two simplex configurations: one side is on-line, the other side is off-line. Both sides are active, but only the on-line side handles call processing related activities.

The off-line side is put through a boot procedure with parameters provided by a command line from the on-line side. All the files used during a boot procedure are exercised. This boot procedure is monitored closely by the on-line side.

Side Switch Operation

When the boot procedure is complete and the off-line side is up and running, a side switch operation can be performed. The side switch operation makes the off-line side the new on-line side, and the original on-line side the new off-line side. Call processing activities will be switched to the new on-line side.

End Operation

The boot procedure can be stopped at any time from the current on-line side. The **STOP** command causes the system to be reconfigured in duplex and optionally restores the devices.

Unless override options are selected, the boot procedure will be aborted through software if the system detects a failure during its execution or a fault that endangers the on-line side operation.

Using OFLBOOT

OFLBOOT is a powerful tool and should be used cautiously. It is recommended in the following circumstances:

- Before system backup to ensure the integrity of the files being backed up.
- During the software release transition procedure to minimize the system service interruption.

OFLBOOT is **not** recommended in the following circumstances:

- When the system is experiencing hardware problems.
- During high traffic hours.

When OFLBOOT is in progress, because of internal queue overflow, the following message may be seen:

```
REPT ERRPORT # MESSAGES LOST
```

Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The following message may also be seen:

```
REPT SOP # MESSAGES LOST
```

The second message can be minimized by changing field 7 of ROP0's *ucb* form. Refer to 254-303-104, *Recent Change and Verify Manual*, for instructions.

Performing OFLBOOT



CAUTION:

Before starting OFLBOOT, the system should be free of any known hardware problems. This will increase the chance of OFLBOOT executing successfully and decrease the chance of endangering the system operation of the on-line side.

If the system is not in duplex configuration, contact the technical support personnel before using override options.

Perform three checks from the Emergency Action Interface (EAI) page before starting OFLBOOT:

- The disks must be forced either primary or secondary from the EAI. This will prevent the system from booting from the wrong software release if a problem occurs and the system phases.
- The Control Unit (CU) must not be forced because it will block the Input/Output (IO) of the CU that has been forced off-line.
- The sanity timer must not be inhibited because it will disable the system recovery mechanism.

One additional check must be done to minimize the chance of OFLBOOT failure. The Switching Control Centers (SCCs), if equipped, must be restorable to Active (ACT) state; otherwise, if a port switch is invoked as part of OFLBOOT, SCC may not be ACT afterward.

In order to access information from the off-line side after OFLBOOT, verify that the log files are located in the `/etc/log` directory. If the log files are located somewhere else, check with your technical support personnel about their location. This information may be needed after stopping the OFLBOOT procedure.

The boot signal generated in the off-line side will result in the yellow CRITICAL alarm indicator being lit. This is the expected behavior of the indicator in all bootstraps (including the off-line boot). The indicator will be cleared automatically within 30 seconds.

How to Start OFLBOOT

The OFLBOOT procedure can be started in the following ways:

- Through the use of an input command
- As part of the application's software release transition procedure defined by the application.

Only the input command procedure is described in this section.

Using the EXC Command

The form of the **EXC:OFLBOOT** input message is as follows:

In PDS,

```
EXC:OFLBOOT: [ TTY a] [, APARM b] [, INITLVL c] [, IOPx [MAX]] \  
[, HDWCHK] [, SFTCHK] [, ERRINT] [, BROOT] [, MINCONFIG] \  
[, UCL d] [, MONITOR] [, RETROFIT] [, OOS] [, TRACE]!
```

In MML,

```
EXC:OFLBOOT:DATA [ TTY=a] [, APARM=b] [, INITLVL=c] \  
[, IOPx [=MAX]] [, HDWCHK] [, SFTCHK] [, ERRINT] [, BROOT] [, MINCONFIG] \  
[, UCL=d] [, MONITOR] [, RETROFIT] [, OOS] [, TRACE];
```

The options and their defaults are described in Table 7-1.

Table 7-1. Options for the EXC:OFLBOOT Input Message

Option	Parameter	Function
TTY	<i>a</i>	CAUTION: <i>By selecting the Teletypewriter (TTY) option, the Input/Output Processor (IOP) associated with it is implicitly assigned to the off-line side.</i> The TTY <i>a</i> option is used as a pseudo Receive-Only Printer (ROP) on the off-line side. All of the off-line side output directed to the ROP is sent to this TTY. The valid range is 0 to 254. If the selected TTY is a <i>UNIX</i> system TTY, the output sent to it may be unformatted until someone logs in. The default is no TTY; the off-line ROP output will not be sent to any output device.
APARM	<i>b</i>	The APARM <i>b</i> option is used as an application parameter for the off-line side. The range is a single alphanumeric character. This is equivalent to the function of Poke 42 on the EAI page during normal boot. The default is no APARM option set for off-line side.
INITLVL	<i>c</i>	CAUTION: <i>These levels apply to RTR initialization. Different applications may or may not support all of them.</i> The INITLVL <i>c</i> option is used as the <i>UNIX</i> RTR operating system bootstrap initialization level for the off-line side. 2 — Boot with existing Equipment Configuration Database (ECD). The initial CU system configuration must be Active/Standby (ACT/STBY). The Out of Service (OOS) option will not override this requirement. 3 — Boot and load new ECD. 4 — (default) Boot and load new ECD and clear Protected Application Segment (PAS).
OOS		Units selected for the off-line side may be initially OOS. Refer to Tables 7-2 and 7-3. The default requires those units to be ACT or STBY.

Table 7-1. Options for the EXC:OFLBOOT Input Message (Contd)

Option	Parameter	Function
IOPx	MAX	The IOP will be removed from service in the on-line side and remain on the original off-line side until OFLBOOT is stopped. Side switching does not affect this assignment. The range (x) can be 0 through 7. If MAX is specified, all of the devices under this IOP will be equipped except the ROP, Maintenance Teletypewriter (MTTY), and SCC and made available to the off-line side. The ROP, MTTY, and SCC will be unequipped on the off-line side. CAUTION: <i>Even with the SCC unequipped, it may perform a link switch. In this case, all EAI capabilities are functional.</i> If MAX is not specified, only peripheral controllers under this IOP will be equipped. Peripheral units under this IOP will not be equipped. If the default option and TTY a option are selected, only TTY a will be equipped under this IOP in addition to the peripheral controllers. If OFLBOOT is a prelude to the software release transition procedure, x has to be either 0 or 1. The default is IOPx without MAX.
HDWCHK		Hardware checks will be inhibited for the off-line side. The default is allowed.
SFTCHK		Software checks will be inhibited for the off-line side. The default is allowed.
ERRINT		Error interrupt will be inhibited for the off-line side. The default is allowed.
BROOT		Backup partitions will be used for the off-line side. The default is the primary partitions. CAUTION: <i>If a side switch is done and then the OFLBOOT procedure stopped, the system will be running on the backup partitions. A boot will be needed afterward to return to the primary (ROOT) partition.</i>

Table 7-1. Options for the EXC:OFLBOOT Input Message (Contd)

Option	Parameter	Function
MINCONFIG		Minimum configuration database will be used for the off-line side. Any subsequent side switch will not be allowed. The default is the full configuration database.
UCL	<i>d</i>	By selecting the UCL option, some application prechecks will be overridden. Refer to Tables 7-2 and 7-3. The default enforces all prechecks. CAUTION: <i>Application initiated aborts will be ignored. Because the UCL option inhibits system recovery actions, it should be used only in a software release transition procedure under technical guidance. Misuse can cause call processing downtime.</i> NOTE: Using the UCL option requires a password (<i>d</i>); to obtain it seek technical assistance. The value <i>d</i> will change every time the UCL option is entered; thus, using the UCL option requires contacting technical assistance.
MONITOR		All off-line Processor Recovery Messages (PRMs) will be directed to the on-line ROP. The default case will have only off-line PRMs up to Step 4 (KBOOT) directed to the on-line ROP.
RETROFIT		Recovery strategy will be modified to accommodate the software release transition. Precheck results will be overridden. Refer to Table 7-2. The default is normal recovery strategy. CAUTION: <i>The RETROFIT option should not be used outside of the software release transition procedure. Misuse can cause call processing downtime.</i>
TRACE		The TRACE option is for troubleshooting purposes. It generates a significant amount of undocumented trace messages and should be used by authorized technical personnel only. The default is no trace message generated.

Table 7-2. Precheck Override Conditions

Option	Application Precheck Result				RTR Precheck ¶
	hwother*	hwmy†	swmay‡	swnever§	
OOS	X				X
UCL	X	X	X		X
RETROFIT**	††	X	X	X	

* Faulty hardware on other side.

† Faulty hardware on my side.

‡ Some faulty software may be overridden.

§ Faulty software may never be overridden.

¶ This column describes the effect of the options on OOS devices belonging to RTR like IOPs, Disk File Controllers (DFCs), and CU. The options will not work if INITLVL 2 is specified and the CU to be removed is not STBY.

** RETROFIT overrides only during side switch.

†† Do not care.

Table 7-3. Secondary DFC Override Condition

On-Line DFC 2 or 3 State	Off-Line DFC 3 or 2 State	Override Option
ACT	ACT	none
ACT	OOS	OOS or UCL
OOS	OOS	UCL
OOS	ACT	UCL*

* UCL will override the OOS DFC state only if units marked as always essential in ECD do not exist under the ACT DFC.

Stepping Through the EXC Command

The following is a sequence of events that occurs when executing the **EXC:OFLBOOT** command. It is not a step-by-step procedure.

Do not proceed with the OFLBOOT procedure if the system is experiencing any hardware problem. Seek technical assistance.

1. Display the EAI page and check that the CU is not forced. If forced, remove the force and restore the CUs to ACT/STBY.
2. Check that the disk is forced either primary or secondary. By forcing the disk, the selection of the DFC is done implicitly. When an office is equipped with four DFCs, and they are ACT or OOS, the selection will be two DFCs, either even-numbered pair (0/2) or odd-numbered pair (1/3) for the on-line side and the remaining pair of DFCs for the off-line side.
3. Check if the nonactive CU is OOS. If it is, restore the CU or use the OOS option. The INITLVL option cannot be 2.
4. The RETROFIT option should be used only during a manual software release transition scenario.



CAUTION:

The RETROFIT option has a strong impact during the side switch. The option allows the side switch to occur even if the switch precheck fails. This can result in a level 3 boot depending on the nature of switch precheck failure. Misuse of this option can cause call processing downtime.

5. Check if the DFC or Moving Head Disk (MHD) that is designated for the off-line side is OOS. If it is, restore the device or use the OOS or UCL option. Refer to Table 7-3 for secondary DFC override condition.
6. If the IOP or TTY option is selected, and any device under this selected IOP is OOS, either restore the device or use the OOS option.



CAUTION:

Even with the SCC unequipped, the SCC may perform a link switch, in which case, all EAI capabilities are functional.

7. If the application precheck needs to be overridden, the UCL option is needed. Refer to Tables 7-2 and 7-3.



CAUTION:

Application initiated aborts will be ignored. Because it inhibits system recovery actions, it should be used only in a software release transition procedure under technical guidance. Misuse of this option can cause call processing downtime.

8. To monitor the off-line ROP output, use the TTY option. The TTY option will implicitly move the associated IOP to the off-line side without the MAX parameter. The off-line ROP output is saved on the off-line side in a file accessible after the OFLBOOT procedure is stopped (see the "Checking OFLBOOT Results" section).
9. To access equipment under a particular IOP from the off-line side, the IOPx MAX option or TTY option is needed.

When the IOP option is used, all the devices under this IOP will be removed from service and then unequipped on the on-line side. The on-line side will no longer have access to these devices.
10. Enter the command with the appropriate options, and the following message will be printed:

EXC OFLBOOT STARTED
11. Every 2 minutes until the off-line side starts the boot, the following message will be printed:

EXC OFLBOOT IN PROGRESS
12. From the start of the off-line side boot procedure until the boot finishes, the following message will be printed:

REPT OFLBOOT INFO
BOOT IN PROGRESS
13. At this time, the Common Processor Display Page (CPDP) shows the status of the equipment. The on-line side equipment designated for the off-line side will be removed from service (OOS). It will then be changed again to Unequipped (UNEQ). There will be many messages printed on the ROP which report the status changes of the equipment. At this moment, the drivers are degrowing the devices.
14. The CPDP will display the following message:

OFLBOOT IP - ONLINE.
15. If an IOP is selected for the off-line side without the MAX parameter, the terminals under this IOP will no longer be available throughout the OFLBOOT procedure until they are restored. The terminals will be in a frozen state (the screen display will no longer change, and no input will be accepted). This phenomenon can occur implicitly when a user selects the TTY option without selecting the IOP with MAX option.
16. The off-line CU boot procedure may be monitored by displaying the EAI page on the MTTY and observing the PRM field.
17. By default, the PRMs from Identifier 0 to Identifier 4 (KBOOT) can be seen on the ROP. The rest of the PRMs will be directed to the *OLBLOG* log files on the off-line side. To monitor all of the off-line PRMs from the on-line ROP, use the MONITOR option on the command line. The failing PRM from the off-line side can be seen on the ROP.

18. At the end of the boot procedure, one of the following messages indicating that the boot is complete will be seen:

EXC OFLBOOT COMPLETED
OFFLINE SIDE BOOT COMPLETED
or
EXC OFLBOOT COMPLETED
OFFLINE SIDE BOOT COMPLETED WITH ERRORS

Because of excessive messages in the system, this completion message may sometimes be lost. The boot completion can be verified by the next step.

19. After the boot procedure, the on-line CU will start to monitor the off-line CU. During this monitoring period, one of the following messages will be output every 5 minutes to the on-line ROP:

REPT OFFLINE BOOT IN PROGRESS - OFFLINE SIDE BOOT COMPLETED
or

REPT OFFLINE BOOT IN PROGRESS - OFFLINE SIDE BOOT COMPLETED WITH ERRORS

Once a side-switch is executed, the following message is output to the ROP:

REPT OFFLINE BOOT IN PROGRESS

20. If a craft terminal is assigned to the off-line side, the following message can be seen on the CPDP:

OFLBOOT IP - OFFLINE

21. If the TTY option is given, the user can monitor the off-line output from this TTY and can also log on to the off-line side.

If the selected TTY is a *UNIX* system terminal, the ROP output redirected to this terminal will not be formatted until the user logs in. If the selected TTY is a craft terminal, the output will be formatted.

How to Stop OFLBOOT

OFLBOOT can be stopped at any time in the following ways:

- Through an input message
- Through an internal software request to abort.

The abort can be requested when a problem has been detected. This request can come from the *UNIX* RTR operating system or other application processes. Only the input command procedure is described in this section.

Using the STOP Command

The form of the **STOP:OFLBOOT** input message is as follows:

In PDS, **STOP:OFLBOOT:[RST] [, TRACE]!**
In MML, **STOP:OFLBOOT:DATA [,RST] [, TRACE];**

If the RST option is specified, requests will be submitted to restore units that were part of the off-line side.

The TRACE option should be used by authorized technical personnel only. It is for troubleshooting purposes and will generate a significant amount of undocumented trace messages. The default is no trace message generated.

Stepping Through the STOP Command

The following is a sequence of events that occurs when executing the **STOP:OFLBOOT** command. It is not a step-by-step procedure.

1. To restore equipment to the ACT state, the RST option is needed. Enter the **STOP** command on the on-line side. The following message will be printed:

STOP OFLB00T STARTED

2. The off-line CU will be stopped and the equipment assigned to the off-line side will be changed from UNEQIP to GROWTH. This is shown in the CPDP. There will be output on the ROP reporting that the drivers are growing the devices.
3. After growing, the devices are in the OOS state.
4. OFLB00T IP - ONLINE on the CPDP will be cleared and the following message will be printed on the ROP:

STOP OFLB00T COMPLETED

5. If the RST option is specified, the requests to restore the units will be submitted to the Maintenance Input Request Administrator (MIRA). These requests can be verified either by checking the Deferred Maintenance Queue (DMQ) (**OP:DMQ**) or by monitoring the ROP output.
6. If no RST option is given, the user can choose to repeat the off-line boot operation or to restore the equipment within 1 hour after the first **STOP** command by issuing a second **STOP** command with the RST option.
If the one-hour time limit has expired, all units must be restored manually. Manual restore can also be used within the one-hour time limit.
7. After the stop OFLB00T sequence has completed, the user should remove the disk force from the EAI page and restore the message queue size if it has been changed as part of preparation for the OFLB00T procedure.

How to Side Switch During OFLB00T

After the off-line side is booted up and running, a side switch can be performed

making the off-line side the new on-line side and the original on-line side the new off-line side. The ROP and MTTY will port switch to the new on-line side, and SCC will link switch to the on-line side.



CAUTION:

During the side switch, transient calls will be lost and call origination will be suspended. Follow the recommended application procedure for the side switch.

The side switch can be requested in the following ways:

- Through the use of an input message.
- As part of the software release transition process defined by the application.

Only the input message procedure is described in this section.

Using the SW Command

The form of the **SW:OFLBOOT** input message is:

In PDS, **SW:OFLBOOT:[UCL][, TRACE]!**
In MML, **SW:OFLBOOT:DATA [,UCL][, TRACE];**

If the UCL option is specified, some application prechecks will be overridden. The default enforces all prechecks.

The TRACE option should be used by authorized technical personnel only. It is for troubleshooting purposes and will generate a significant amount of undocumented trace messages. The default is no trace message generated.

The off-line side can be booted on a new software release. When a side switch is done, this new software release will become the on-line side and control the system.

Stepping Through the SW Command

The following is a sequence of events that occurs when executing the **SW:OFLBOOT** command. It is not a step-by-step procedure.

1. To override some precheck results, use the UCL option. Refer to Tables 7-2 and 7-3.

If there are two DFCs on the off-line side, an error condition on any one of them will cause a *UNIX* RTR operating system precheck failure. The side switch will not succeed under this condition unless the RETROFIT option is provided with the **EXC:OFLBOOT** command. Refer to Step 4 of the "Stepping Through the **EXC** Command" section for the RETROFIT option impact.

2. Enter the command.
3. Immediately, the MTTY will refresh and display the EAI page and the ROP will stop. This is part of the port switch. The CPDP will change from:

OFLBOOT IP - ONLINE

to

OFLBOOT IP - OFFLINE.

4. The CPDP on the new on-line craft terminal will change from:
`OFLB00T IP - OFFLINE`
to
`OFLB00T IP - ONLINE .`
5. After the port switch completes, the MTTY will be refreshed and will display the CPDP. The ROP will resume printing. Now they are under control of the new on-line CU. The SCC links will be switched to the new on-line side.
6. If the HDWCHK, SFTCHK, or ERRINT option is selected for the **EXC** command, the option will be shown on the EAI page when running on the side that was off-line booted. The options can be removed by executing an appropriate command. For example, to remove inhibit of HDWCHK, enter:
ALW:HDWCHK
7. Dual Serial Channel Computer Interconnect (DCI) units will not be available automatically after a side switch in the new on-line side. If any are present, restore them manually.

Checking OFLBOOT Results

There are several ways to check OFLBOOT results. They can be checked during and after OFLBOOT.

Except for some new output messages and a new log file on the off-line side, the result of OFLBOOT can be interpreted using existing mechanisms (PRMs, error reports, postmortem, log files, and so forth).

⇒ NOTE:

The last six digits of any PRM contain hardware information. These digits are not readable by software processes since they are appended by the hardware that outputs the PRMs. Therefore, when the PRMs are redirected through software to the *OLBLOG* file, the hardware digits cannot be captured and do not form part of the PRM in the *OLBLOG* file.

Available Information During OFLBOOT

During OFLBOOT, the user can monitor progress by following the PRMs. Not all PRMs will be sent to the on-line ROP unless the MONITOR option has been given on the **EXC** command.

When the off-line side is booted, an application precheck will be performed on the off-line side (part of the booting process), and its result will be brought back to the on-line side. Application software will print the result to the ROP. There is no command to explicitly check the boot results.

The off-line side postmortem is brought back to the on-line side with the **STOP** command. This will not be done if the on-line postmortem slots are in use.

There are *OLBLOG* files on the off-line side that logs all the outputs sent to the ROP. These log files can be accessed during OFLBOOT.

Available Information After OFLBOOT

This procedure is valid only if the system disks have not been restored to duplex configuration (the MHDs assigned to the off-line side are still in the OOS state). If an MHD is already restoring, it is too late to use this procedure.

The example described in this section assumes DFC 1 is assigned to the off-line side and *OLBLOG* files are in the */etc/log* directory. If DFC 0 has been assigned to the off-line side, then substitute DFC 1 with DFC 0, and substitute MHD 1 with MHD 0 in the following procedures. If the *OLBLOG* files are not in the */etc/log* directory, check with your technical support personnel to learn where the *OLBLOG* files are located before proceeding.

The log file will be available if **EXC:OFLBOOT** is stopped or terminated after the off-line side has completed the boot procedure (check for the **OFFLINE SIDE BOOT COMPLETE** message). The log file may also be available after the **REPT OFLBOOT IN PROGRESS** message is seen on the ROP if OFLBOOT is stopped or terminated before the boot is completed.

The following procedure uses PDS format commands. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Step	Procedure
1	Verify that OFLBOOT is stopped by checking the output messages. If either of the following messages are printed, go to Step 3: STOP OFLBOOT COMPLETED EXC OFLBOOT TERMINATED
2	If OFLBOOT has not stopped yet, stop it by entering the following command at the craft terminal and return to Step 1: STOP:OFLBOOT
3	Make sure there are no diagnostics running by entering the following command: OP:DMQ If an MHD is already restoring, it is too late to use this procedure. Do not continue this procedure. If diagnostics are running, either wait for them to complete or stop them by entering the following command: STOP:DMQ:ACTIVE Proceed to the next step when the diagnostic queue is empty.
4	Restore unconditionally the associated DFC 1 controller and SBUS 1 by entering the following commands: RST:DFC 1;UCL:CONT RST:SBUS 1;UCL:CONT The restore of SBUS is necessary only if the system has Small Computer System Interface (SCSI) DFCs.

Step	Procedure
5	The <i>OLBLOG</i> files are in <i>/etc/log</i> directory. They will be copied to the same location on the active disk. The corresponding active disk <i>OLBLOG</i> files are of no importance since all the outputs have been captured by the ROP. If a side switch has been performed, that is, the system is stopping on the originally off-line side, perform the next command before proceeding: OP:LOG:LG "OLBLOG" Use the following commands to copy the OOS disk <i>OLBLOG</i> files to the active disk: COPY:OOSDISK;MHD 1, SRC "/etc/log/OLBLOG0",\ DEST "/etc/log/OLBLOG0" COPY:OOSDISK;MHD 1, SRC "/etc/log/OLBLOG1",\ DEST "/etc/log/OLBLOG1"
6	The <i>OLBLOG</i> files are now on the active disk. The following command can be used to print them to the ROP: OP:LOG:LG "OLBLOG" After recovery of <i>OLBLOG</i> files, MHD 1 can be restored together with all other MHDs by entering the following command: RST:DFC 1;UCL

Examples of Performing OFLBOOT

This section describes two example procedures for performing OFLBOOT.

- Without selecting an IOP for the off-line side.
- With one IOP selected for the off-line side.

The side switch will not be described by these two examples because it should be used only as part of the software release transition procedure.

The examples describe straightforward scenarios; no error scenarios are described. The examples are described using the PDS command format. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Performing OFLBOOT Without IOP Selected

An example procedure for performing OFLBOOT without selecting an IOP for the off-line side follows.

Step	Procedure
1	The system must be free of any known hardware problems. If not, repair the system before continuing. Contact the technical support personnel for assistance if needed.
2	From the EAI display page, force the primary boot disk. poke 20
3	Check on the EAI page that the sanity timer is not inhibited. If it is, release it by using the following poke from the EAI page before continuing: poke 25
4	One CU must be ACT, and the other CU must be STBY. If one CU is OOS, restore it before continuing. RST:CU x If there are other devices in the OOS state, restore them to either ACT or STBY state before continuing to Step 5. Refer to 254-303-110, <i>PDS Input Messages Manual</i> , or 254-303-112, <i>MML Input Messages Manual</i> , for the proper restore command.

Step	Procedure
5	Make sure that CU 0 is ACT by checking the CPDP. If it is STBY, perform a switch CU. poke 400
6	Initiate OFLBOOT by entering the following command: EXC:OFLBOOT
7	Check on the ROP that the command has been accepted properly by looking for EXC OFLBOOT STARTED output message. If there is an error message like EXC OFLBOOT STOPPED, check the message and correct the problem accordingly. After correcting the problem, restore the system to the original state and repeat the procedure from Step 4. If there are no error messages, continue to Step 8.
8	Check on the ROP that OFLBOOT is being executed by looking for EXC OFLBOOT IN PROGRESS output message. If there is an error message like REPT OFLBOOT STOPPED or REPT OFLBOOT ABORTED, check the message and correct the problem accordingly. After correcting the problem, restore the system to the original state and repeat the procedure from Step 4. If there are no error messages, continue to Step 9.
9	Check on the ROP for the off-line side boot result by looking for the OFFLINE SIDE BOOT COMPLETE or REPT OFFLINE BOOT TIMED OUT message. Either message indicates the end of the boot procedure for the off-line side.
10	Stop the OFLBOOT procedure by entering the following command: STOP:OFLBOOT The following message will be seen on the ROP: STOP OFLBOOT STARTED
11	Check on the ROP for the following completion message: STOP OFLBOOT COMPLETED

Step	Procedure
12	Check that there are no diagnostics running by entering the following command: OP:DMQ If diagnostics are running, stop them by entering the following command before continuing to Step 13: STOP:DMQ:ACTIVE
13	Restore unconditionally the DFC 1 controller by entering the following command: RST:DFC 1;UCL:CONT
14	If the system is not equipped with SCSI DFCs, skip this step. If the system is equipped with SCSI DFCs, restore unconditionally the SBUS 1 controller by entering the following command: RST:SBUS 1;UCL:CONT
15	Copy the off-line <i>OLBLOG</i> files to the on-line side by entering the following commands: COPY:OOSDISK:MHD 1,SRC "/etc/log/OLBLOG0",\ DEST "/etc/log/OLBLOG0" COPY:OOSDISK:MHD 1,SRC "/etc/log/OLBLOG1",\ DEST "/etc/log/OLBLOG1"
16	Print the <i>OLBLOG</i> files on the ROP by entering the following command: OP:LOG:LG "OLBLOG"
17	Restore the system by entering the following command: STOP:OFLBOOT:RST
18	Check that the command has been accepted and executed by looking for the following messages on the ROP: STOP OFLB00T STARTED STOP OFLB00T COMPLETED

Performing OFLBOOT with One IOP Selected

An example procedure for performing OFLBOOT with one IOP for the off-line side is as follows:

Step	Procedure
1	The system must be free of any known hardware problems. If not, repair the system before continuing. Contact the technical support personnel for assistance if needed.
2	From the EAI display page, force the primary boot disk. poke 20
3	Check on the EAI page that the sanity timer is not inhibited. If it is, then release it by using the following poke from the EAI page before continuing: poke 25
4	One CU must be ACT, and the other CU must be STBY. If one CU is OOS, restore it before continuing. RST:CU x
5	Make sure that CU 0 is ACT by checking the CPDP. If it is STBY, perform a switch CU. poke 400
6	Select one IOP for the off-line side (either IOP 0 or IOP 1). Select one TTY under this IOP to be the pseudo ROP for the off-line side. For this example IOP will be 1, and TTY will be 5.
7	Verify that SCCs can be restored to the ACT state. For this example, SCC will be 0. First restore the SCC that is ACT by entering the following command: RST:SCC 0 After restoring SCC 0, restore the SCC 1 by entering the following command: RST:SCC 1 If both SCCs can be restored, then continue to Step 8; otherwise, do not continue.

Step	Procedure
8	Initiate OFLBOOT by entering the following command: EXC:OFLBOOT:IOP1 MAX, TTY 5
9	Check on the ROP that the command has been accepted properly by looking for the EXC OFLBOOT STARTED output message. If there is an error message, such as EXC OFLBOOT STOPPED, check the message and correct the problem accordingly. After correcting the problem, restore the system to the original state and repeat the procedure from Step 4. If no error messages have been found, continue to Step 10.
10	Check on the ROP that OFLBOOT is being executed by looking for the EXC OFLBOOT IN PROGRESS output message. If there is an error message like REPT OFLBOOT STOPPED or REPT OFLBOOT ABORTED, check the message and correct the problem accordingly. After correcting the problem, restore the system to the original state and repeat the procedure from Step 4. If no error messages have been found, continue to Step 11.
11	Check on the ROP for the off-line side boot result by looking for OFFLINE SIDE BOOT COMPLETE or REPT OFFLINE BOOT TIMED OUT message.
12	Stop the OFLBOOT procedure by entering the following command: STOP:OFLBOOT
13	Check the ROP for the following completion message: STOP OFLBOOT COMPLETED
14	Check that there are no diagnostics running by entering the following command: OP:DMQ If diagnostics are running, stop them by entering the following command before continuing to Step 15: STOP:DMQ:ACTIVE

Step	Procedure
15	Restore unconditionally the DFC 1 controller by entering the following command: RST:DFC 1;UCL:CONT
16	If the system is not equipped with SCSI DFCs, skip this step. If the system is equipped with SCSI DFCs, restore unconditionally the SBUS 1 controller by entering the following command: RST:SBUS 1;UCL:CONT
17	Copy the off-line <i>OLBLOG</i> files to the on-line side by entering the following commands: COPY:OOSDISK:MHD 1,SRC "/etc/log/OLBLOG0",\ DEST "/etc/log/OLBLOG0" COPY:OOSDISK:MHD 1,SRC "/etc/log/OLBLOG1",\ DEST "/etc/log/OLBLOG1"
18	Print the <i>OLBLOG</i> files on the ROP by entering the following command: OP:LOG:LG "OLBLOG"
19	Restore the system by entering the following command: STOP:OFLBOOT:RST
20	Check that the command has been accepted by looking for STOP OFLB00T STARTED message on the ROP.

Log File Review

8

Contents

Introduction	8-1
Using Log Files	8-3
Core Dump Administration	8-5

Introduction

The *UNIX*® Real-Time Reliable (RTR) operating system maintains a history of certain automatic and manual maintenance actions that occur in the system. These histories are located in the */etc/log* directory and can be printed by the **OP:LOG** command. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for the complete command.)

⇒ NOTE:

Do not clear these files without consulting your next level of technical assistance.

The various log files and their contents include the following:

- *PMLOG* – contains postmortem dumps (generated each time the computer goes through an automatic or manual phase), phase source information, and the states of various registers in the computer.
- *MEMLOG* – contains any correctable or noncorrectable memory errors that occur in the on-line or standby computer.
- *DKDRVLOG* – contains error messages generated from the disk driver.
- *ERLOG* – contains all error interrupts not sent to *MEMLOG*.
- *IODRVLOG* – contains all error codes and fault codes generated by the Input/Output (IO) driver.
- *AUDLOG* – contains all audit failures.

- *OLBLOG* – contains all output directed to the Receive-Only Printer (ROP) during OFLBOOT.



NOTE:

The last six digits of any Processor Recovery Message (PRM) contain hardware information. These digits are not readable by software processes since they are appended by the hardware that outputs the PRMs. Therefore, when the PRMs are redirected through software to the *OLBLOG* file, the hardware digits cannot be captured and do not form part of the PRM in the *OLBLOG* file.

- *SIMLOG* – contains problems detected by the system integrity monitor usually dealing with resource overload conditions.
- *SPLLOG* – contains history information and error messages generated by the Coordinator of Spooler Output Processes (CSOP).
- *ULARPLOG* – contains error reports and process restart messages generated by the User-Level Automatic Restart Process (ULARP).
- *UNIXLOG* – contains a record of UNIX shell input entered from operation terminals that are connected to the switch. Each log entry has a header preceding the UNIX shell input which contains the LOGNAME and originating TTY based on the shell environment and a timestamp of when the input was entered.

Refer to 254-303-111, *PDS Output Messages Manual*, and 254-303-113, *MML Output Messages Manual*, for details on data included in the log files.

Using Log Files

Log files are limited in size to avoid excessive consumption of system resources. Two files are actually maintained. The system always writes to a log file with an appended *1*; for example, *PMLOG1*. When the maximum allowable size is reached, the file is copied to a file with the same name but with an appended *0*; for example, *PMLOG0*. The next time *PMLOG1* reaches its maximum size, the copy occurs again and the previous data in *PMLOG0* is lost. Enter the following to obtain a list of these files, their current sizes, and other information:

In PDS, **OP:STATUS:LISTDIR,FN "/etc/log"**!
In MML, **OP:STATUS:DATA,LISTDIR,FN="/etc/log";**

Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for a complete description.

The list may contain some files with the name *rmfxxxx*, where *xxxx* is any number. These files contain incompletely processed log entries. They are generated when the computer is not sane enough to write to the log files. The Recovery Message Formatter (RMF) will clean up these files after it prints out the information contained in them. If any of the *rmfxxxx* files are zero-length, they may be removed using the **CLR: FILESYS: FILE** command. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for the complete command.

NOTE:

Log files should **NOT** be removed using this command.

Log files are useful for the following:

- Detecting and correcting transient errors before they cause system outages. In particular, correctable memory errors can occur at an increasing rate over many days. This problem can be detected by studying log file entries and then corrected before it occurs at a rate high enough to cause automatic recovery actions.

⇒ **NOTE:**

Transient errors are usually not detected by diagnostics because of their transient nature, but sufficient information exists in the log file entry to determine the cause.

- Determining the sequence of events that led to a system phase or maintenance interrupt. Each log file entry has time-of-day information, real-time clock values, and sequence numbers. Study this information to determine possible computer problems. For example, if multiple errors caused an excessive error rate and triggered a phase, the sequence numbers indicate the order and the time stamp verifies that the occurrence was sufficiently short to cause the phase.

Entries from several log files may have to be pieced together to give a chronological order of events. Examine printouts of all the log files to re-create the order of events.

An entry in a log file may be an indication of trouble. When log file entries occur, they are also printed on the ROP and Maintenance Teletypewriter (MTTY) (if available). However, it is difficult to find these entries among the other output and impossible to detect a subtle problem of increasing frequency. Therefore, examine the log files daily and correct problems as soon as they are identifiable.

Log files may be examined using the **OP:LOG** command. Specific entries in these log files may be deleted using the **DEL:LOG** command. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for more information regarding the use of these commands.

⇒ **NOTE 1:**

When using these utilities, messages indicating log file corruption may be printed. These messages indicate problems with the log file and not the file system. These corrupt entries are automatically cleaned up when the **DEL:LOG** command is run.

⇒ **NOTE 2:**

The *OLBLOG* file is available on the off-line side during OFLBOOT. It can be accessed by mounting the off-line disk after OFLBOOT has stopped and before the disk is restored.

Core Dump Administration

A core dump takes place if certain faults occur or if a termination message requests one. Core dumps are images of the run-time state of processes that have been placed on disk in the */cdmp* directory (or in a user-defined directory specified by the termination message).

Generally, a core dump implies an abnormal process termination. A core dump should be investigated to determine the cause of termination. Each site should save the core dump files when log files are saved, or as often as necessary. Core dump files must be removed periodically to allow enough space for future core dumps.

⇒ NOTE:

Files in the *cdmp* directory may not be useful to the office craft. However, these files should be saved (just as log files and ROP output are saved) for forwarding to the next level of technical support, if necessary. Refer to Chapter 2, "Maintenance Plan," for further guidelines.

Contents

Introduction	9-1
Checking File-System Resources	9-1
Manipulating Files	9-2
Using File-System Audits	9-5
■ File-System Block and Link Audits	9-5
■ File-System Compaction Audit	9-8
File Corruption Detection (SIV Only)	9-9
Boot File Verification Through OFLBOOT	9-10
System Backup	9-11
■ 3B20D/3B21D Computers TOP Tape	9-11
■ Common TOP Tape	9-12
Copying to Backup Partitions	9-13
Making a Backup Disk	9-26
■ Method 1 — Copying to a Spare Disk Drive	9-26
Procedure A	9-26
Procedure B	9-29
■ Method 2 — Using a Warm Spare Disk Drive	9-31
Making a Backup Tape Copy of a System Disk	9-34
Initialize Disk Partition with Zeros	9-36
Copying Files to Nonactive Disks	9-37
Comparing and Correcting Disk Data	9-39
■ Operations for Comparing and Correcting Disk Data	9-39

Contents

Read-Only Operation	9-39
Example 1	9-39
Example 2	9-40
Example 3	9-40
Compare Operation	9-40
Example 4	9-40
Example 5	9-41
Example 6	9-41
Example 7	9-41
Correction Operation	9-41
Example 8	9-41
Example 9	9-42
Example 10	9-42
Stop Operation	9-42
Example 11	9-42
■ Limitations of Disk Compare and Correct Commands	9-43
Procedure for Handling Disk Errors	9-44

Introduction

File-system maintenance includes the following:

- Checking file-system resources
- Manipulating files
- Using the file-system audits
- File-system corruption detection [Software Integrity Verification (SIV)]
- Backing up and restoring file systems.

Checking File-System Resources

The `REPT FILESYS OVERFLOW` output message is a warning that a file system is about to run out of space. Immediate action must be taken to provide more space in the file system. Contact your technical assistance organization to determine the files that can be removed to provide file-system space.

File-system free space can be monitored and action taken before the free space is used up. The **OP:STATUS:FREEDISK** command reports the free space and free inodes in the mounted file systems. Monitor these numbers regularly to determine if space or inodes are being depleted.

Running the file-system block audit (refer to “Using File-System Audits” in this chapter) may increase the free space in a file system if the audit finds lost resources that it can recover. No error is reported when lost free disk blocks are recovered.

The allocation of disk space to contiguous files is limited to the size of the largest set of contiguous free disk blocks in the file system. In a fragmented file system, this may be considerably less than total free space. The compaction audit (refer to “Using File-System Audits”) can be used to increase the size of contiguous free disk space.

The */tmp* and */usr/tmp* directories contain temporary files used by many programs and need to be periodically checked and cleaned up. If either is allowed to run out of disk blocks or inodes, the programs may be unable to function correctly. If a file system runs out of inodes, no message is output, but the error number returned by the **creat()** system call will indicate that the file system is out of inodes. The file-system audits may be able to recover lost resources, but most likely old temporary files were not removed after their use by some program. These files will have to be manually identified and removed.

Manipulating Files

Tables 9-1 through 9-3 list the commands that manipulate files in the system.

- The commands listed in Table 9-1 may be used routinely to determine actual or potential troubles in the system.
- The commands listed in Table 9-2 will not cause permanent changes and may be used for troubleshooting if sufficient knowledge of the file system exists.
- The commands listed in Table 9-3 can cause permanent changes in the file structure or process structure of the system and should only be used with concurrence of a technical assistance organization or as directed in a Broadcast Warning Message (BWM).

Table 9-1. File-System Maintenance Commands (Routine)

Command	Description
OP:STATUS:FILESYS	Lists all currently mounted file systems, the directory under which they are mounted, and the time they were mounted.
OP:STATUS:FREEDISK	Lists the mounted file systems with the number of free blocks and free inodes. Used to determine if file-system space is being depleted.
OP:STATUS:LISTDIR	Checks for the existence of a file, its size, and permissions.
OP:STATUS:SUM	Calculates a checksum for a given file. Used to determine if two files are exactly the same.

Table 9-2. File-System Maintenance Commands (Troubleshooting)

Command	Description
DUMP:FILE:ALL	Prints the contents of an American Standard Code for Information Interchange (ASCII) file on the Receive-Only Printer (ROP).
DUMP:FILE:FORMAT	Prints the contents of a file in ASCII, decimal, octal, or hexadecimal.
DUMP:FILE:PARTL	Prints one or more lines of an ASCII file.
OP:STATUS:DISKUSE	Reports the number of blocks used by a file.
VFY:FILE	Verifies the contents of files using cyclic redundancy check algorithm.

Table 9-3. File-System Maintenance Commands (Permanent)

Command	Description
ALW:FILESYS:ACCESS	Changes access permissions of files.
COPY:PTN:ALL	Copies a disk partition to another partition. This command is used to recover mutilated disk partitions from backup disk partitions and to generate partition backup copies.
ALW:FILESYS:MOUNT	Mounts an unmounted file system so that the system can access files in the file system. Mounting a file system "read only" is done routinely as part of system backup to check a backup file system via the file-system audit.
ALW:FILESYS:OWNER	Changes the owner of a file. All <i>UNIX</i> RTR operating system files are owned by <i>root</i> .
CLR:FILESYS:DIR	Removes a directory from the file system.
CLR:FILESYS:FILE	Removes a file from the file system.
COPY:FILESYS:CFILE	Copies a file to a contiguous area of the disk. Contiguous files are transferred from disk to main memory more quickly than regular files.
COPY:FILESYS:FILE	Makes another copy of a file in a different directory. The copied file has the same name as the original file.
IN:FILE:APND	Appends a line to an ASCII file.
IN:FILE:DELT	Deletes one or more lines from an ASCII file.
IN:FILE:REPL	Replaces lines in an ASCII file with user supplied lines.
IN:FILESYS:DIR	Creates a new directory in the file system.
INH:FILESYS:UMOUNT	Unmounts a file system. This command denies system access to the file system. The file system must be inactive to be unmounted. Unmounting a file system is done as part of a system backup after a "read only" mount of a backup file system.

Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for details on any specific input message and for system responses.

Using File-System Audits

The operating system provides two audits, FSBLK and FSLINK, to verify the structural integrity of file systems. A third file-system audit, FSCMPT, is provided to merge some of the free space of a fragmented file system into a large contiguous free space. All three of these audits are capable of being run on either mounted or unmounted file systems. Therefore, it is not necessary to either mount or unmount a file system in order to audit it.

File-System Block and Link Audits

File-System Block (FSBLK) Audit — This audit detects bad blocks and duplicate blocks and verifies the size and mode of each file in a file system. When run in the correcting mode, the FSBLK audit rebuilds the file-system structures that keep track of free blocks and inodes. In some cases, the free space in a file system may be increased if lost resources are recovered by the audit.

File-System Link (FSLINK) Audit — This audit verifies the link integrity among the directories and inodes of a file system and detects mutilated inodes. When run in the correcting mode, the FSLINK audit corrects bad link counts and removes unreferenced inodes and bad directory entries.

Refer to Chapter 14, “System Audits,” for detailed instructions on running these audits and interpreting their output messages.

If either of these audits report an uncorrectable error, the audited file system must be backed up using the procedures given in the “Copying to Backup Partitions” section.

If correctable errors are reported, it is necessary to determine which files have been affected by the errors. The **OP:FNAME** command should be used to convert block or inode numbers listed in audit error reports into filenames.

The file system must be mounted in order to use the **OP:FNAME** command. If it is not already mounted, the following procedure should be used to mount it.

Step	Procedure
1	Create directory <i>/mountfs</i> : In PDS, IN:FILESYS:DIR,FN "/mountfs"! In MML, IN:FILESYS:DATA,DIR,FN="/mountfs"; and wait for a completion message. ⇒ NOTE: It is necessary to create this directory only once. To verify that it was created, enter the following messages: In PDS, OP:STATUS:LISTDIR,FN "/"! In MML, OP:STATUS:DATA,LISTDIR,FN="/";
2	Mount the file system by entering the following messages: In PDS, ALW:FILESYS:MOUNT,FN "/dev/f1",BSDIR "/mountfs",RO! In MML, ALW:FILESYS:DATA,MOUNT,FN="/dev/f1",BSDIR="/mountfs",RO; Where <i>f1</i> is the name of the file system to be mounted.

If the audit error reports list block numbers, enter the following commands to get the filename(s) with the bad blocks:

In PDS, **OP:FNAME:FN "f1",BLOCK b1!**
In MML, **OP:FNAME:DATA,FN="f1",BLOCK=b1;**

Where *f1* is the name of the file system (for example, */dev/root*) and *b1* is a block number reported by the audit.

If the audit error reports list inode numbers, enter the following messages:

In PDS, **OP:FNAME:FN "f1",INODE i1!**
In MML, **OP:FNAME:DATA,FN="f1",INODE=i1;**

Where *i1* is an inode number reported by the audit.

The filenames listed in the **OP:FNAME** output message are the files that have probably been damaged by the file-system errors reported by the audit. If the file system had to be mounted in order to use the **OP:FNAME** command, it should now be unmounted again. Enter:

In PDS, **INH:FILESYS:UMOUNT,FN "/dev/f1"!**
In MML, **INH:FILESYS:DATA,UMOUNT,FN="/dev/f1";**



CAUTION:

After the file-system structural damage has been repaired, contact your technical assistance organization to determine if files that were affected by the file-system errors or removed by the audit must be replaced with backup copies.



WARNING:

The following commands, which run the file system audits in correcting mode, should be used with extreme caution. Know what you are doing or contact a higher level of support.

If the audit was running in a noncorrecting mode when it reported the errors, it must be rerun in a correcting mode to correct the errors. If the error was detected by either the FSBLK or FSCMPT audit, use the FSBLK audit to correct the error by entering one of the following messages:

In PDS, **AUD:FSBLK 1,INS "/dev/f1";CORR!**
In MML, **AUD:FSBLK=1,INS="/dev/f1":CORR;**

If the error was detected by the FSLINK audit, enter one of the following messages:

In PDS, **AUD:FSLINK 1,INS "/dev/f1";CORR!**
In MML, **AUD:FSLINK=1,INS="/dev/f1":CORR;**

File-System Compaction Audit

File-System Disk Space Compaction (FSCMPT) Audit — This audit changes the assignment of disk space to files in order to merge a part of the free space of a file system into a large contiguous free space. Like the FSBLK audit, it detects duplicate blocks and bad blocks and corrects these errors when run in the correcting mode. In the noncorrecting mode, the FSCMPT audit makes no changes to achieve a compacted file system. Rather, it reports the number of contiguous free blocks it expects to be able to create when run in the correcting mode and the number of Input/Output (IO) jobs required to do this.

The FSCMPT audit may be used when there is not enough contiguous free disk space to create a contiguous file of the desired size, even though the total free space in the file system is larger than the desired size.

To handle this resource shortage, the FSBLK audit should be used first to recover any lost disk space. Run the FSBLK audit in noncorrecting mode. If any errors are reported, use the procedures described in “File-System Block and Link Audits” to correct the errors. Then run the FSBLK audit in the correcting mode.

If there is still not enough contiguous free disk space of the desired size, the compaction audit may be used.



CAUTION:

Since the compaction audit involves considerable overhead in time and file manager resources, it should be run only when the system is relatively quiet and when activity in the file system is low.

The FSCMPT should be run first in the noncorrecting mode in order to determine how much contiguous free space it can recover. Two parameters *max/min* and *cond/ucl* are provided to control the compaction effort and the size of the recovered contiguous free space. Refer to Chapter 14, “System Audits,” for detailed instructions on running this audit and interpreting the output messages.

Select the parameters that will give the desired contiguous free space and run the audit in the correcting mode. If the amount of contiguous free space recovered by the audit is not sufficient, a repeated invocation of the audit will increase the size of the contiguous free space until it can do no more. Note that not all the free space of the file system will be merged into a contiguous area since some free blocks are reserved for regular file growth.

File Corruption Detection (SIV Only)

Essential File Verification Utility — This utility, which is part of SIV, verifies any files which are critical to the operation of the system. The **VFY:FILE** command is used to invoke this feature. A Cyclic Redundancy Check (CRC) value should be generated for each essential file prior to running the utility. Failure to follow this procedure will result in reports of failure from the utility. The CRC value which is generated should be stored within the essential file. If this is not possible, the file */.crcvalues* will be created to store the information for essential files such as LBOOT and BBOOT.

The user can specify a set of files that is verified by one command on a timed basis or manually. To run the command manually, refer to **VFY:FILE** in either 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*. To set up the command to run on a timed basis, a new entry (for **VFY:FILE**) must be added to */usr/lib/crontab*. The default set of essential files will contain LBOOT, BBOOT, Volume Table of Contents (VTOC), */appdmert*, and */dmert*. When using the default, the time stamp of the boot image files will be compared with each of their components to verify that they are up to date.

The user can invoke the utility to verify each file in a file system which has the *crc-code* field set in its *fsb* record. The user can also create a specification file consisting of lines containing the filename, followed by white space and the letter "i" or "g" to indicate the location of the cyclic information for that file.

The **VFY:FILE** command can be used to verify the correctness of "TOP" tapes, if the "TOP" tape is mounted on tape drive unit 0. This command should be run periodically to assure that a load disk from the tape operation will succeed.

This utility is applicable to both the on-line and off-line disks. Using the utility, files stored on Physical Disk-to-Tape (PDT) format tape could be read into an off-line disk and verified if the CRCs exist on the tape.

Boot File Verification Through OFLBOOT

Boot Files Verification Utility — The OFLBOOT utility verifies the bootability of files that are involved during a system boot. The **EXC:OFLBOOT** command is used to invoke this feature. This feature temporarily splits the duplex system into two simplex configurations, and then performs a boot on one side.

The OFLBOOT feature allows craft personnel to verify the integrity of files used during a boot procedure while the system is active and handling call processing activities. It allows one side of a duplex system to go through a complete boot procedure exercising the equipment available and the interface between *UNIX*® Real-Time Reliable (RTR) operating system and application subsystems. The feature can be stopped by invoking **STOP:OFLBOOT**.

This feature can also be used as part of the Software Release Transition procedure. It can exercise the new software on the off-line side, verify the sanity before side switching to it, and making it the on-line and active side. The side switching can be done by invoking **SW:OFLBOOT**.



CAUTION:

This feature should be used with caution since it puts the system in a vulnerable simplex configuration during its duration. Refer to Chapter 7, "Off-Line Boot," for more information.

System Backup

The *UNIX* RTR operating system allows for system backups both in immediately accessible form (backup partitions) and in off-line forms (backup disk copies and tape copies).

Each installation should keep backup copies of all disks on hand if recovery action from backup copies is to be initiated. These backup copies should be made on tape and/or disk. To perform a disk backup, an off-line spare disk drive is needed for each nonremovable media disk to be backed up.

It is recommended that the OFLBOOT feature be exercised before each system backup procedure. This is to ensure the integrity of files to be saved on the backup disks.

Disks should always be backed up by copying to backup partitions and by using the PDT writer. Additional backups can be made by using Volume Copy (VCP) to a spare disk drive. The backup copies should **always** be verified by using **VFY:MHD ...** and **VFY:TAPE ...** input messages. Also, these backup copies should be placed in a secure area.

Decisions on when to make backup copies are based on the number of changes applied to the system in a given time. Usually, backups are taken after completing a BWM soak cycle. Writing to tapes (using PDT) does not interfere with the system but is time consuming. The sequence of system recovery is covered in "Recovering the System Manually" and "Recovery Using Software Backup Disk and Alternate Boot Disk Options" sections in Chapter 3, "Recovery."

3B20D/3B21D Computers TOP Tape

To create a "TOP" tape for the 3B20D and 3B21D computers (needed for a deadstart), copy a "TOP" image from the disk onto the tape (with block size of 512 bytes). The **dd** command can be used to copy the "TOP" image on the disk.

To create a new "TOP" tape from the disk image, mount the tape and then enter:

```
In PDS,    COPY:TAPE:TOP, TD "a"!
In MML,    COPY:TAPE:TOP, TD="a";
```

Where a is the tape device name.

A valid tape device name can be found in Chapter 20, "Magnetic Tape Special Device Filenames." This command creates a TOP tape for the current hardware platform only. For example, if the current hardware platform is Very Large Main Memory (VLMM), */etc/topfile64* or */etc/topfile* will be copied to tape.

Common TOP Tape

To create a "COMMON TOP" tape which will run on either the VLMM or the 3B21D computer, copy *topfile64* (with no-rewind option) followed by *topfile21* (with rewind option), strictly in that order, onto the tape (with block size of 512 bytes). The **dd** command can be used to copy the "COMMON" image on the disk.

To create a "COMMON TOP" tape from the disk images, mount the tape and then enter:

```
In PDS,    COPY:TAPE:TOP, TD "a", COM!
In MML,    COPY:TAPE:TOP, TD="a", COM;
```

Where a is the tape device name.

A valid tape device name can be found in Chapter 20, "Magnetic Tape Special Device Filenames." This command copies */etc/topfile64* and */etc/topfile21* from disk to tape, in that order.

To verify the correctness of a "TOP" tape, enter:

```
In PDS,    VFY:FILE, FN "a"!
In MML,    VFY:FILE:DATA, FN= "a";
```

Where a is the tape device name.

Copying to Backup Partitions

The following partitions on the disk have backups (applications may have other partition backups):

Primary Partition	Backup Partition
<i>/dev/boot</i>	<i>/dev/bboot</i>
<i>/dev/root</i>	<i>/dev/broot</i>
<i>/dev/etc</i>	<i>/dev/betc</i>
<i>/dev/db</i>	<i>/dev/bdb</i>

⇒ NOTE:

The partitions */dev/boot* and */dev/bboot* do not have a file-system structure. They are files used during the bootstrap sequence only. They can be copied with the **COPY:PTN** command but cannot be mounted.

A copy can be made from the primary partitions to the backup partitions or from the backup partitions to the primary partitions depending on which is active on the system. To make the copy, perform the following procedure:

Step	Procedure
1	Verify Moving Head Disk (MHD) 0 and MHD 1 are in duplex; enter: In PDS, OP:OOS! In MML, OP:OOS; If MHD 0 or MHD 1 is listed in the output message, restore the listed MHD to service (RST:MHD).
2	Determine if the system is currently running on the primary or backup file systems; enter: In PDS, OP:STATUS:FILESYS! In MML, OP:STATUS:DATA,FILESYS; The output message indicates the currently mounted file systems. If <i>/dev/root</i> and <i>/dev/db</i> are listed, the primary partitions are active. If <i>/dev/broot</i> and <i>/dev/bdb</i> are listed, the backup partitions are active. The partitions are copied from the active (or mounted) partitions to the inactive (or unmounted) partitions.

Step	Procedure
3	Check the integrity of the disks by running verifies on each; enter: In PDS, VFY:MHD 0! In MML, VFY:MHD=0; and wait for a completion message; then enter: In PDS, VFY:MHD 1! In MML, VFY:MHD=1; and wait for another completion message.  NOTE: If the disk drive is a Small Computer System Interface (SCSI) type, there is no need to wait before requesting the second verify. If one MHD fails, remove it from service (RMV:MHD), initialize it (INIT:MHD), and then restore it to service (RST:MHD). If both fail, contact a technical assistance organization. If both MHDs pass, continue to Step 4.
4	Check the integrity of the active partitions by inhibiting routine audits and then running the FSBLK and FSLINK audits on each active partition using Step 4(a) or 4(b). (a) If the primary partitions are active; enter: In PDS, INH:AUD:ALL! In MML, INH:AUD:DATA,ALL; and wait for an acknowledgment; then enter: In PDS, AUD:FSBLK 1,INS "/dev/root";NCG! In MML, AUD:FSBLK=1,INS="/dev/root":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/root";NCG! In MML, AUD:FSLINK=1,INS="/dev/root":NCG;

Step	Procedure
	and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/etc";NCG! In MML, AUD:FSBLK=1,INS="/dev/etc":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/etc";NCG! In MML, AUD:FSLINK=1,INS="/dev/etc":NCG; and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/db";NCG! In MML, AUD:FSBLK=1,INS="/dev/db":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/db";NCG! In MML, AUD:FSLINK=1,INS="/dev/db":NCG; and wait for a completion message. Continue to Step 5. (b) If the backup partitions are active, enter: In PDS, INH:AUD:ALL! In MML, INH:AUD:DATA,ALL; and wait for an acknowledgment; then enter: In PDS, AUD:FSBLK 1,INS "/dev/broot";NCG! In MML, AUD:FSBLK=1,INS="/dev/broot":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/broot";NCG! In MML, AUD:FSLINK=1,INS="/dev/broot":NCG; and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/betc";NCG! In MML, AUD:FSBLK=1,INS="/dev/betc":NCG;

Step	Procedure
	and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/betc";NCG! In MML, AUD:FSLINK=1,INS="/dev/betc":NCG; and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/bdb";NCG! In MML, AUD:FSBLK=1,INS="/dev/bdb":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/bdb";NCG! In MML, AUD:FSLINK=1,INS="/dev/bdb":NCG; and wait for a completion message.
5	Perform Steps 5(a) through 5(f) to create and verify a file named <i>/tmp/primary</i> and a file named <i>/tmp/backup</i> which will be used as the source (SRC) and destination (DEST) filenames in the COPY:PTN command. ⇒ NOTE: If you have the <i>UNIX</i> system available on your machine and are familiar with the ed editor, you may create and verify the files with the ed editor. Simply create a file named <i>/tmp/primary</i> that contains the following lines: <pre>/dev/boot /dev/root /dev/etc /dev/db</pre> Then, create a file named <i>/tmp/backup</i> that contains the following lines: <pre>/dev/bboot /dev/broot /dev/betc /dev/bdb</pre> Also, remember that applications may need additional partitions (one partition name per line) added to <i>/tmp/primary</i> and <i>/tmp/backup</i>.

Step	Procedure
	If you used the <i>UNIX</i> system to create <i>/tmp/primary</i> and <i>/tmp/backup</i>, go to Step 6.
(a)	Verify if the file <i>/tmp/primary</i> already exists; enter: In PDS, DUMP:FILE:ALL,FN "/tmp/primary"! In MML, DUMP:FILE:DATA,ALL,FN="/tmp/primary"; If the file <i>/tmp/primary</i> already exists, go to Step 5(c) to verify the contents; otherwise, continue to Step 5(b).
(b)	Create the <i>/tmp/primary</i> file; enter: In PDS, IN:FILE:APND,FN "/tmp/primary",LINE 0! Press the <CR> key. At the : prompt, type: "/dev/boot"! IN:FILE:APND,FN "/tmp/primary",LINE 1! Press the <CR> key. At the : prompt, type: "/dev/root"! IN:FILE:APND,FN "/tmp/primary",LINE 2! Press the <CR> key. At the : prompt, type: "/dev/etc"! IN:FILE:APND,FN "/tmp/primary",LINE 3! Press the <CR> key. At the : prompt, type: "/dev/db"! In MML, IN:FILE:DATA,APND,FN="/tmp/primary",LINE=0! Press the <CR> key. At the : prompt, type: "/dev/boot"; IN:FILE:DATA,APND,FN="/tmp/primary",LINE=1! Press the <CR> key. At the : prompt, type: "/dev/root"; IN:FILE:DATA,APND,FN="/tmp/primary",LINE=2! Press the <CR> key. At the : prompt, type: "/dev/etc"; IN:FILE:DATA,APND,FN="/tmp/primary",LINE=3! Press the <CR> key. At the : prompt, type: "/dev/db";

Step	Procedure
(c)	Verify the contents of the <i>/tmp/primary</i> file; enter: In PDS, DUMP:FILE:ALL,FN "/tmp/primary"! In MML, DUMP:FILE:DATA,ALL,FN="/tmp/primary"; The file <i>/tmp/primary</i> will be output. Verify that the file has an entry for each primary partition in the order listed as follows: <i>/dev/boot</i> <i>/dev/root</i> <i>/dev/etc</i> <i>/dev/db</i>
(d)	Verify if the file <i>/tmp/backup</i> already exists; enter: In PDS, DUMP:FILE:ALL,FN "/tmp/backup"! In MML, DUMP:FILE:DATA,ALL,FN="/tmp/backup"; If the file <i>/tmp/backup</i> already exists, go to Step 5(f) to verify the contents; otherwise, continue to Step 5(e).
(e)	Create the <i>/tmp/backup</i> file; enter: In PDS, IN:FILE:APND,FN "/tmp/backup",LINE 0/ Press the <CR> key. At the : prompt, type: "/dev/bboot"! IN:FILE:APND,FN "/tmp/backup",LINE 1/ Press the <CR> key. At the : prompt, type: "/dev/broot"! IN:FILE:APND,FN "/tmp/backup",LINE 2/ Press the <CR> key. At the : prompt, type: "/dev/betc"! IN:FILE:APND,FN "/tmp/backup",LINE 3/ Press the <CR> key. At the : prompt, type: "/dev/bdb"! In MML, IN:FILE:DATA,APND,FN="/tmp/backup",LINE=0! Press the <CR> key. At the : prompt, type: "/dev/bboot"; IN:FILE:DATA,APND,FN="/tmp/backup",LINE=1!

Step	Procedure
	Press the <CR> key. At the : prompt, type: "/dev/broot"; IN:FILE:DATA,APND,FN="/tmp/backup",LINE=2! Press the <CR> key. At the : prompt, type: "/dev/betc"; IN:FILE:DATA,APND,FN="/tmp/backup",LINE=3! Press the <CR> key. At the : prompt, type: "/dev/bdb";
(f)	Verify the contents of the <i>/tmp/backup</i> file; enter: In PDS, DUMP:FILE:ALL,FN "/tmp/backup"! In MML, DUMP:FILE:DATA,ALL,FN="/tmp/backup"; The file <i>/tmp/backup</i> will be output. Verify that the file has an entry for each primary partition in the order listed as follows: /dev/bboot /dev/broot /dev/betc /dev/bdb
6	Copy the active partitions to the inactive partitions by performing either Step 6(a) or 6(b). Instead of specifying the partition names, two ordinary files containing the partition names will be specified (one partition name per line) in the command line. (One MHD is automatically removed from service and acts as the source disk for the copy. After the copy completes, the out-of-service MHD is restored to service.) (a) If the primary partitions are active, enter: In PDS, COPY:PTN:ALL,SRC "/tmp/primary",DEST "/tmp/backup"! In MML, COPY:PTN:DATA,ALL,SRC="/tmp/primary",DEST="/tmp/backup"; and wait for a completion message. Then, go to Step 7. Where <i>"/tmp/primary"</i> is the file created in Step 5 that contains the desired active partitions to be copied and <i>"/tmp/backup"</i> is a file also created in Step 5 that contains the inactive backup partitions.

Step	Procedure
	(b) If the backup partitions are active, then enter: In PDS, COPY:PTN:ALL,SRC "/tmp/backup",DEST "/tmp/primary"! In MML, COPY:PTN:DATA,ALL,SRC="/tmp/backup",DEST="/tmp/primary"; and wait for a completion message. Where <i>"/tmp/primary"</i> is the file created in Step 5 that contains the desired active partitions to be copied and <i>"/tmp/backup"</i> is a file also created in Step 5 that contains the inactive backup partitions.
7	Check the integrity of the new inactive partition copies by inhibiting routine audits and then running the FSBLK and FSLINK audits on each new partition using Step 7(a) or 7(b). (a) If the primary partitions are active; enter: In PDS, INH:AUD:ALL! In MML, INH:AUD:DATA,ALL; and wait for an acknowledgment; then enter: In PDS, AUD:FSBLK 1,INS "/dev/broot";NCG! In MML, AUD:FSBLK=1,INS="/dev/broot":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/broot";NCG! In MML, AUD:FSLINK=1,INS="/dev/broot":NCG; and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/betc";NCG! In MML, AUD:FSBLK=1,INS="/dev/betc":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/betc";NCG! In MML, AUD:FSLINK=1,INS="/dev/betc":NCG;

Step	Procedure

and wait for a completion message; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/bdb";NCG!**
In MML, **AUD:FSBLK=1,INS="/dev/bdb":NCG;**

and wait for a completion message; then enter:

In PDS, **AUD:FSLINK 1,INS "/dev/bdb";NCG!**
In MML, **AUD:FSLINK=1,INS="/dev/bdb":NCG;**

and wait for a completion message.

If the file-system audits report errors on any partitions, then continue to Step 8 to copy and verify only the specific partition that failed.

If no file-system audit errors are reported,
STOP. YOU HAVE COMPLETED THIS PROCEDURE.

To resume the scheduling of routine audits, enter:

In PDS, **ALW:AUD:ALL!**
In MML, **ALW:AUD:DATA,ALL;**

(b) If the backup partitions are active, enter:

In PDS, **INH:AUD:ALL!**
In MML, **INH:AUD:DATA,ALL;**

and wait for an acknowledgment; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/root";NCG!**
In MML, **AUD:FSBLK=1,INS="/dev/root":NCG;**

and wait for a completion message; then enter:

In PDS, **AUD:FSLINK 1,INS "/dev/root";NCG!**
In MML, **AUD:FSLINK=1,INS="/dev/root":NCG;**

and wait for a completion message; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/etc";NCG!**
In MML, **AUD:FSBLK=1,INS="/dev/etc":NCG;**

Step	Procedure
	and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/etc";NCG! In MML, AUD:FSLINK=1,INS="/dev/etc":NCG; and wait for a completion message; then enter: In PDS, AUD:FSBLK 1,INS "/dev/db";NCG! In MML, AUD:FSBLK=1,INS="/dev/db":NCG; and wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/db";NCG! In MML, AUD:FSLINK=1,INS="/dev/db":NCG; and wait for a completion message. If the file-system audits report errors on any partitions, then continue to Step 8 to copy and verify only the specific partition that failed. If no file-system audit errors are reported, STOP. YOU HAVE COMPLETED THIS PROCEDURE. To resume the scheduling of routine audits, enter: In PDS, ALW:AUD:ALL! In MML, ALW:AUD:DATA,ALL;
8	This step is only performed if the audits reported a failure in Step 7. Perform Step 8(A) to verify the active partition, Step 8(B) to copy it, and Step 8(C) to re-verify the new backup individual partition. ⇒ NOTE: If the errors occur again while performing Step 8, contact your technical assistance organization for advice on how to correct the problem. (A) In this step, perform Step (a) or (b) to check the integrity of the active partition by inhibiting routine audits and then running the FSBLK and FSLINK audits on each active partition.

Step	Procedure

- (a) If the primary partitions are active; enter:

In PDS, **INH:AUD:ALL!**

In MML, **INH:AUD:DATA,ALL;**

and wait for an acknowledgment; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/a_ptn_name";NCG!**

In MML, **AUD:FSBLK=1,INS="/dev/a_ptn_name":NCG;**

Where "a_ptn_name" is the active_partition_name.

Wait for a completion message; then enter:

In PDS, **AUD:FSLINK 1,INS "/dev/a_ptn_name";NCG!**

In MML, **AUD:FSLINK=1,INS="/dev/a_ptn_name":NCG;**

and wait for a completion message.

- (b) If the backup partitions are active; enter:

In PDS, **INH:AUD:ALL!**

In MML, **INH:AUD:DATA,ALL;**

and wait for an acknowledgment; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/b_ptn_name";NCG!**

In MML, **AUD:FSBLK=1,INS="/dev/b_ptn_name":NCG;**

Where "b_ptn_name" is the backup_partition_name.

Wait for a completion message; then enter:

In PDS, **AUD:FSLINK 1,INS "/dev/b_ptn_name";NCG!**

In MML, **AUD:FSLINK=1,INS="/dev/b_ptn_name":NCG;**

and wait for a completion message.

Step	Procedure
	(B) To copy the partition, enter: In PDS, COPY:PTN:ALL,SRC "/dev/a_ptn_name",\ DEST "/tmp/b_ptn_name"! In MML, COPY:PTN:DATA,ALL,SRC="/dev/a_ptn_name",\ DEST="/tmp/b_ptn_name"; and wait for a completion message. Where "a_ptn_name" is the active_partition_name and "b_ptn_name" is the backup_partition_name. (C) In this step, perform Step (a) or (b) to check the integrity of the new partition copies by inhibiting routine audits and then running the FSBLK and FSLINK audits on each new partition. (a) If the primary partitions are active; enter: In PDS, INH:AUD:ALL! In MML, INH:AUD:DATA,ALL; and wait for an acknowledgment; then enter: In PDS, AUD:FSBLK 1,INS "/dev/b_ptn_name";NCG! In MML, AUD:FSBLK=1,INS="/dev/b_ptn_name":NCG; Where "b_ptn_name" is the backup_partition_name. Wait for a completion message; then enter: In PDS, AUD:FSLINK 1,INS "/dev/b_ptn_name";NCG! In MML, AUD:FSLINK=1,INS="/dev/b_ptn_name":NCG; and wait for a completion message. If the file-system audits report errors on any partitions, then contact your technical assistance organization for advice on how to correct the problem.

Step	Procedure

If no file-system audit errors are reported,
STOP. YOU HAVE COMPLETED THIS PROCEDURE.

To resume the scheduling of routine audits, enter:

In PDS, **ALW:AUD:ALL!**
In MML, **ALW:AUD:DATA,ALL;**

(b) If the backup partitions are active; enter:

In PDS, **INH:AUD:ALL!**
In MML, **INH:AUD:DATA,ALL;**

and wait for an acknowledgment; then enter:

In PDS, **AUD:FSBLK 1,INS "/dev/a_ptn_name";NCG!**
In MML, **AUD:FSBLK=1,INS="/dev/a_ptn_name":NCG;**

Where "a_ptn_name" is the active_partition_name.

Wait for a completion message; then enter:

In PDS, **AUD:FSLINK 1,INS "/dev/a_ptn_name";NCG!**
In MML, **AUD:FSLINK=1,INS="/dev/a_ptn_name":NCG;**

and wait for a completion message.

If the file-system audits report errors on any partitions, then
contact your technical assistance organization for advice on
how to correct the problem.

If no file-system audit errors are reported,
STOP. YOU HAVE COMPLETED THIS PROCEDURE.

To resume the scheduling of routine audits, enter:

In PDS, **ALW:AUD:ALL!**
In MML, **ALW:AUD:DATA,ALL;**

Making a Backup Disk

Disk backups can be made by using one of two methods:

- Method 1 – Copying to a spare disk drive.
 - Procedure A – Copying data from an Out of Service (OOS) MHD to an OOS spare MHD.
 - Procedure B – Copying a specific partition (or list of partitions) from an OOS MHD to an active spare MHD.
- Method 2 – Using a warm spare disk drive.

Method 1 — Copying to a Spare Disk Drive

Procedure A

Procedure A is used to copy data from an OOS disk (for example, MHD 0 or 1) to an OOS spare disk (for example, MHD 2 or 3). The number of blocks specified will determine the data to be copied (for example, if the number of blocks is 50,000, then the first 50,000 blocks on the disk will be copied).

Step	Procedure
1	Determine the MHD number of the disk from which and to which data is to be copied. For this procedure, substitute the MHD number of the MHD to be copied <i>to</i> for the <i>A</i> in MHD A and the number of the MHD to be copied <i>from</i> for the <i>B</i> in MHD B .
2	Verify the integrity of the source disk; enter: In PDS, VFY:MHD B! In MML, VFY:MHD=B; and wait for a completion message.  NOTE: Do not use this disk for a source disk if it has any verify failures.
3	Remove from service the destination MHD; enter: In PDS, RMV:MHD A! In MML, RMV:MHD=A; and wait for a completion message.

Step	Procedure
4	Remove from service the MHD from which data is to be copied; enter: In PDS, RMV:MHD B! In MML, RMV:MHD=B; and wait for a completion message.
5	Format the disk to which data is to be copied; enter: In PDS, INIT:MHD A;VFY! In MML, INIT:MHD=A:VFY; and wait for a completion message. ⇒ NOTE: Do not use this disk if there are any verify failures.
6	Create special files for the copy; enter: In PDS, EXC:ENVIR:UPROC,FN "/etc/dgnnm", ARGV ("MHD", "B")! In MML, EXC:ENVIR:DATA,UPROC,FN="/etc/dgnnm",ARGV="MHD"- "B"; The system responds as follows: /tmp/dkdn - where <i>n</i> is some number. Then enter: In PDS, EXC:ENVIR:UPROC,FN "/etc/dgnnm",ARGV ("MHD", "A")! In MML, EXC:ENVIR:DATA,UPROC,FN="/etc/dgnnm",ARGV="MHD"- "A"; The system responds as follows: /tmp/dkdm - where <i>m</i> is some number.

Step	Procedure
7	Execute the copy; enter: In PDS, EXC:ENVIR:UPROC,FN "/etc/vcp",ARGS ("/tmp/dkdn",\ "/tmp/dkdm","0","XXXXXX","-")! In MML, EXC:ENVIR:DATA,UPROC,FN="/etc/vcp",\ ARGS="/tmp/dkdn"-"/tmp/dkdm"-0"-XXXXXX"-"; Where <i>n</i> and <i>m</i> are numbers obtained in Step 6 and XXXXXX is the total number of blocks to be copied. This number varies with the application. The number of blocks to be copied can be obtained by dumping VTOC or by using the following command; enter: In PDS, OP:STATUS:DISKUSE,FN "a"! In MML, OP:STATUS:DATA,DISKUSE,FN="a"; Where <i>a</i> is the pathname of the directory or file. The system responds with a series of dots followed by <code>vcp</code> completed. The copy may take up to 30 minutes to complete.
8	After the copy completes, remove the special files; enter: In PDS, EXC:ENVIR:UPROC,FN "/etc/udgnnm",ARGS ("MHD", "A")! In MML, EXC:ENVIR:DATA,UPROC,FN="/etc/udgnnm",ARGS="MHD"-A"; and wait for a completion message. Enter: In PDS, EXC:ENVIR:UPROC,FN "/etc/udgnnm",ARGS("MHD", "B")! In MML, EXC:ENVIR:DATA,UPROC,FN="/etc/udgnnm",ARGS="MHD"-B"; and wait for a completion message.

Procedure B

Procedure B copies a specific partition or a list of partitions from an OOS disk to an active spare disk. One of the destination disks (for example, MHD 1) will be removed from service and restored when the copy is completed. Note that for this procedure, special device filenames (destination partitions), which map onto the spare disk, will be required.

To create a special device for a disk, enter:

In PDS, **EXC:ENVIR:UPROC,FN "/etc/mknod",ARGS "v","w","x","y","z"!**

In MML, **EXC:ENVIR:UPROC,FN="/etc/mknod",ARGS="v"- "w"- "x"- "y"- "z";**

v should be replaced with the name of the partition to be created (for example, */dev/newroot*).

w should be replaced with the letter *b* to specify that the partition will be a block device.

x should be replaced with the *dcn* (device control number).

y should be replaced with the partition number.

z should be replaced with the *mdct* (minor device chain table) pointer for the device.

If a partition for a disk already exists, the *dcn* and *mdct* pointer can be obtained using the **Is -I** command. The *dcn* and *mdct* pointer can also be obtained from the *ucb* form in the Equipment Configuration Database (ECD).

Step	Procedure
1	Determine the disk pair associated with the source partitions (for example, <i>/dev/vtoc</i> and <i>/dev/root</i> are in MHD 0 and 1) and the spare disk to which the partitions will be copied.
2	Make sure that the disk pair containing the partitions to be copied is duplexed (for example, both disks are active) and the spare disk is also active.
3	If the destination disk is active, go to Step 6. If the spare disk is a new disk, enter: In PDS, INIT:MHD A;VFY! In MML, INIT:MHD=A;VFY;

Step	Procedure
4	If the spare disk is not active and does not contain a valid VTOC (for example, if a new disk pack is installed), enter: In PDS, COPY:ACTDISK:MHD A,SRC "a",PTN "0"! In MML, COPY:ACTDISK:DATA,MHD=A,SRC="a",PTN="0"; Where A is the unit number of the spare disk and a is the name of the VTOC partition to be copied; for example, <i>"/dev/vtoc"</i> is the VTOC partition for MHD 0 and 1.  CAUTION: <i>Attention all super-users. During partial updates, the VTOC structure for all partitions that are not updated must be identical to the disk VTOC on all tape sequences. If they are not identical, then the starting location of some of the partitions will not match their actual starting location on the disk.</i>
5	Restore the disk; enter: In PDS, RST:MHD A! In MML, RST:MHD=A;
6	Execute the copy; enter: In PDS, COPY:SPDISK;SRC "a",DEST "b"! In MML, COPY:SPDISK:SRC="a",DEST="b"; Where: a specifies a partition name or a regular file which contains a list of partitions to be copied. For example, <i>"/dev/root"</i> specifies the special device name for the root partition. <i>"/tmp/srclist"</i> specifies the location of <i>"srclist"</i>, where each line contains the name of the source partitions. b specifies the special device for the corresponding destination partition or a regular file containing a list of destination partitions. The disk containing the source partitions (for example, MHD 1) will first be removed from service and then the data will be copied from this disk. The number of blocks copied for each partition will be reported. After all partitions are copied, the disk (from which the partitions were copied) will be restored to service.

Method 2 — Using a Warm Spare Disk Drive

This method should be used if a warm spare of a duplex pair is required.

Step	Procedure
1	Determine which duplex pair of MHDs is to have a warm spare. For example, if you want a warm spare of a boot device, make sure that the boot device has a mate. The boot device pair is usually MHD 0 and MHD 1. The remainder of this procedure will use MHD A, B, or C as the names of the disk drives and the appropriate numerical substitutions will have to be made. MHD A and C must be on the same DFC. If the DFC is a SCSI DFC, MHD A and MHD C must be on the same SBUS.
2	Execute a verify command on both MHD A and MHD B. Do not proceed until both disks are error free.
3	Make sure both MHD A and MHD B are in duplex operation.
4	Remove from service MHD A and MHD C to which data is to be copied; enter: In PDS, RMV:MHD A! In MML, RMV:MHD=A; In PDS, RMV:MHD C! In MML, RMV:MHD=C; and wait for a completion message.
5	For SCSI disks: The drive must be powered down before changing the setting of the ID switch. Remove power from both MHD A and MHD C, toggle the settings of each MHD's switch to that of the other MHD, power up both MHDs, and continue to Step 6. For 340-MB disks: The drive must be OOS before removing or inserting the ID plug. No access of the drive must take place from the time the drive was removed from service to the time the ID plug is changed to another value.

Step	Procedure
	Exchange the ID plugs, and then continue to Step 6.
	 CAUTION: <i>Never remove or install ID plugs in an Storage Module Drive (SMD) drive without first removing the drive from service. To change the ID plug when the drive is active could cause data mutilation or corruption.</i> <i>For the ID plug exchange procedure to succeed, there can be no access of the disk drive between removing the drive from service and changing the ID plug. This means that no maintenance commands can be directed to the drive; for example, DGN, VERIFY, DUMP VTOC, and so forth.</i>
	 CAUTION: <i>During the period that the drive IDs are swapped, SCAN POINT and SD POINT information and data (ROP messages, such as POWER DOWN, and so forth) derived from the SCAN POINT information, will NOT be swapped. This could lead to misinformation being output on the ROP.</i>
6	Format the disk drive to which data is to be copied; enter: In PDS, INIT:MHD A;VFY! In MML, INIT:MHD=A:VFY; and wait for a completion message.
7	Restore the MHD to service; enter: In PDS, RST:MHD A! In MML, RST:MHD=A; This makes MHD C a warm spare of MHD A. Wait for a completion message.
8	Remove the disk drive from service; enter: In PDS, RMV:MHD A! In MML, RMV:MHD=A; and wait for a completion message.

Step	Procedure
9	Label the disk drive with the appropriate information: date, MHD number, release issue, last BWM, and any local identification information. This disk drive now becomes the warm spare.
10	If SMD MHD, switch the ID plugs back to the original configuration. If SCSI MHD, the drives must be powered down to switch ID numbers. MHD A is now physically MHD A.
11	Restore the MHD to service; enter: In PDS, RST:MHD A! In MML, RST:MHD=A; This restores the system to the normal duplex configuration.

Making a Backup Tape Copy of a System Disk

The tape writing program is used to copy the image of selected disk partitions to tape. Names of the disk partitions (to be copied) must be specified in the PDT specification file (one partition name per line). By default, the specification filename is */etc/pdtspec*. An optional specification filename and an option to merge generic and database partitions [on the same tape(s)] can be specified using the **COPY:BKDISK** input message. All of the disk partitions specified in a specification file must reside on the same disk (or disk pair). If the application needs to copy from different disks (for example, from MHD 0 and MHD 2), then the tape writer should be executed twice, each time specifying the proper disk containing the partition names specified in the specification file. The contents of the specification file may have to be changed between runs.

For partitions with backup copies on disk, the unmounted partitions are copied to tape. For example, if the system was initialized from the primary partitions (for example, */dev/root* and */dev/db*) and */etc/pdtspec* contains */dev/root* and */dev/db*, then the partitions */dev/broot* and */dev/bdb* will be copied to tape.

To generate a backup tape copy, use the following procedure:

Step	Procedure																											
1	Verify that the unmounted partitions contain the data to be copied to tape. (Refer to the “Copying to Backup Partitions” section.)																											
2	Obtain a sufficient number of tapes to hold the copy. Typical storage capacities for a 2400 foot 9-track tape written with COPY:BKDISK are as follows: <table><tr><th>Density</th><th>Block Size</th><th>Capacity</th></tr><tr><td>1600 bpi</td><td>2048 (default)</td><td>26 Mbytes</td></tr><tr><td>1600 bpi</td><td>6144 (EXT option)</td><td>38 Mbytes</td></tr><tr><td>6250 bpi</td><td>2048 (default)</td><td>77 Mbytes</td></tr><tr><td>6250 bpi</td><td>6144 (EXT option)</td><td>128 Mbytes</td></tr></table> Typical Digital Audio Tape (DAT) unit capacities are as follows: <table><tr><th>Length</th><th>Block Size</th><th>Capacity</th></tr><tr><td>30 meters</td><td>2048 or 6144</td><td>650 Mbytes</td></tr><tr><td>60 meters</td><td>2048 or 6144</td><td>1300 Mbytes</td></tr><tr><td>90 meters</td><td>2048 or 6144</td><td>2000 Mbytes</td></tr></table>	Density	Block Size	Capacity	1600 bpi	2048 (default)	26 Mbytes	1600 bpi	6144 (EXT option)	38 Mbytes	6250 bpi	2048 (default)	77 Mbytes	6250 bpi	6144 (EXT option)	128 Mbytes	Length	Block Size	Capacity	30 meters	2048 or 6144	650 Mbytes	60 meters	2048 or 6144	1300 Mbytes	90 meters	2048 or 6144	2000 Mbytes
Density	Block Size	Capacity																										
1600 bpi	2048 (default)	26 Mbytes																										
1600 bpi	6144 (EXT option)	38 Mbytes																										
6250 bpi	2048 (default)	77 Mbytes																										
6250 bpi	6144 (EXT option)	128 Mbytes																										
Length	Block Size	Capacity																										
30 meters	2048 or 6144	650 Mbytes																										
60 meters	2048 or 6144	1300 Mbytes																										
90 meters	2048 or 6144	2000 Mbytes																										
3	Check the integrity of the disks by running verify on each. If the integrity of the unmounted file systems is questionable, run the file-system integrity audits. (Refer to the “Using File-System Audits” section.)																											

Step	Procedure
4	Mount the first tape, and then enter: In PDS, COPY:BKDISK;START:SRC "/dev/vtoc",TD "x",TPSIZE y! In MML, COPY:BKDISK:START:DATA,SRC="/dev/vtoc",TD="x",TPSIZE=y; Where: x is the pathname of the rewind tape device. y is the tape length. For 9-track tapes, the length is specified in feet ranging from 600 to 2400. For DAT unit tapes (hereafter known as DAT tapes), the length is specified in meters ranging from 30 to 90. ⇒ NOTE: In the above commands, "/dev/vtoc" refers to duplex disk pair MHD 0 and MHD 1. When the copy completes and the tape rewinds, the following message is printed: COPY BKDISK DISMOUNT TAPE, LABEL, AND MOUNT NEXT TAPE
5	Within 10 minutes after the COPY BKDISK message prints, mount the next tape and then enter: In PDS, COPY:BKDISK;ACK:TPSIZE y! In MML, COPY:BKDISK:ACK:DATA,TPSIZE=y; Where: y is the tape length.
6	Repeat Step 5 until all tapes are written. The following message prints when the copy is finished: COPY BKDISK COMPLETED
7	Remove the last tape, label it, and store the tapes in a secure area. ⇒ NOTE: Tapes should be labeled with date, release issue number, last BWM, tape sequence number, and any local identification information.

Initialize Disk Partition with Zeros

The Zero Disk Partition (ZDP) utility is used to clear partitions on a disk selected from a duplex disk pair. The target disk will be removed from service and the application partitions on that disk, determined from a specification file, will be initialized with zeros. After the utility has completed, the maintenance person will be responsible for replacing the disk packs properly and restoring the out-of-service disk.

To use the ZDP utility, the user needs to specify a destination disk unit and the names of the application partitions to be cleared. The disk unit should be one of a duplex disk pair, and the application partitions should be specified in a specification file.

The format of the command is as follows:

```
In PDS,   CLR:PTN;MHD a ;SPECF "b"!  
In MML,   CLR:PTN:MHD=a :SPECF="b";
```

Where: *a* is the disk unit to be initialized with zeros.

b is the complete pathname of a file containing a list of the partition names to be initialized with zeros. The default is the */etc/clearspec* file.

Copying Files to Nonactive Disks

The active/nonactive disk copy utility is used to copy a file from an active to a nonactive disk. The file can be a regular file, a contiguous file (type C or x), or a block device file (type b consists of a partition or a file system).

To use this facility, the user needs to specify the following:

- The number of the destination disk. This disk must be OOS or Off-Line (OFL) and does not have to be a mate of the Active (ACT) disk.
- The full pathname of the source file that exists on an ACT disk. If the source is a regular or contiguous file, then this file must exist on a mounted file system.
- The full pathname or number of the destination partition on the nonactive disk. If a name is specified, then it must exist as a special device file on the active disk.

If the source is a regular or contiguous file, then this partition must be a file system.

- The pathname on the destination partition where the file is to be written. If the destination pathname is not specified, then the pathname of the source file will be used. If the destination pathname starts with a "/", then the mount point will be excluded from the destination pathname. It is not required that all directories in the pathname specified exist.

Examples for Copying Files

To copy a VTOC partition to the nonactive disk (MHD 2), enter:

```
In PDS,  
COPY:ACTDISK :MHD 2 ,SRC "/dev/vtoc" ,PTN "/dev/mhd2"!  
In MML,  
COPY:ACTDISK :DATA ,MHD=2 ,SRC="/dev/vtoc" ,PTN="/dev/mhd2";
```

To copy the root partition from MHD 0 to the corresponding root partition on MHD 1 (OOS disk), enter:

```
In PDS,  
COPY:ACTDISK :MHD 1 ,SRC "/dev/root" ,PTN "/dev/root"!  
In MML,  
COPY:ACTDISK :DATA ,MHD=1 ,SRC="/dev/root" ,PTN="/dev/root";
```

To copy */dev/root* to partition 6 on MHD 2, enter:

In PDS,

COPY:ACTDISK :MHD 2 ,SRC *"/dev/root"* ,PTN "6"!

In MML,

COPY:ACTDISK :DATA ,MHD=2 ,SRC=*"/dev/root"* ,PTN="6";

To copy a contiguous file to the root file system (*/dev/root*) on the nonactive disk, enter:

In PDS,

COPY:ACTDISK :MHD 1 ,SRC *"/appdmert"* ,PTN *"/dev/root"*!

In MML,

COPY:ACTDISK :DATA ,MHD=1 ,SRC=*"/appdmert"* ,PTN=*"/dev/root"*;

To copy a contiguous file from the root file system to a different file system on the nonactive disk (MHD 1), enter:

In PDS,

**COPY:ACTDISK :MHD 1 ,SRC *"/appdmert"* ,PTN "20" ,DEST **
***"odd/appdmert"*!**

In MML,

COPY:ACTDISK :DATA ,MHD 1 ,SRC=*"/appdmert"* ,PTN="20", DEST=
***"odd/appdmert"*!**

In the last example, a destination file is necessary because the partition number associated with the source file (*"/"* is the mount point for */dev/root*) and the destination partition numbers are not the same.

Comparing and Correcting Disk Data

Four commands are used to compare and correct disk data. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for the complete command.)

- **CMPR:MHD**
Compares the contents of two disks or two disk partitions.
- **COPY:DIFF**
Compares disk partitions in order to detect differences. When a difference is found, data is copied from the source partition to the destination partition.
- **STOP:CMPR;MHD** and **STOP:COPY;DIFF**
Both commands stop any in-progress **CMPR:MHD** or **COPY:DIFF** commands.



CAUTION:

*While available for use with all applications, the disk correction operation (**COPY:DIFF**) should only be performed in consultation with your next level of technical support. Improper use of the correction operation can result in disk data mutilation or in excessive use of system resources.*

Operations for Comparing and Correcting Disk Data

The operations associated with the disk compare and correct commands are described with examples in the following subsections.

Read-Only Operation

Similar in nature to the compare operation, the read-only operation ensures the readability of all disk blocks. Examples 1, 2, and 3 show how to use the **CMPR:MHD** command in the read-only mode.

Example 1

This example shows how to perform a read on both disks of a duplex pair. There is no restriction on the major state of the units for this command invocation.

Enter:

```
In PDS,  CMPR:MHD a;MATE:RO!  
In MML,  CMPR:MHD=a:MATE:DATA,RO;
```

Where a is the number of the specified unit.

The command performs reads using a job size of 128 blocks (the default) on a specified unit and its mate unit. A summary message, indicating the number of read errors encountered, is output after every partition is read.

Example 2

This example performs reads for two specified units. This time, however, only the specified partitions are read. An error occurs if the specified partitions do not match in size on the given disks. The job size for each read is specified to be 128 blocks. Enter:

```
In PDS,    CMPR:MHD a,PTN(x-y);MHD b:RO,BLOCKS 128!  
In MML,    CMPR:MHD=a,PTN=x--y:MHD=b:DATA,RO,BLOCKS=128;
```

Where *a* and *b* are the number of the specified units and *x* and *y* represent the specified partition number.

Example 3

This example shows how to perform reads for a specified disk using a job size of 64 blocks. Enter:

```
In PDS,    CMPR:MHD a:RO,BLOCKS 64!  
In MML,    CMPR:MHD=a:DATA,RO,BLOCKS=64;
```

Compare Operation

The compare operation is for use on partitions and disks which are independent of (and unaware of) any other nonmaintenance disk activities or processes involving those partitions and disks. This operation can be performed on any two disks in any major state; these disks do not have to be mates.

Appropriate mismatch messages are generated indicating the range of disk blocks where the mismatch occurred. A maximum of eight mismatch error messages are output for each partition. Nine or more mismatch messages will result in a termination of the operation on that partition. The operation proceeds to the next partition (if any).

Example 4

This example compares a specified disk with its mate. Enter:

```
In PDS,    CMPR:MHD a!  
In MML,    CMPR:MHD=a ;
```

Example 4 shows that the mate unit for the **CMPR:MHD** command does not have to be specified.

Example 5

This example compares a specified disk with its mate using the MATE keyword. Enter:

In PDS, **CMPR:MHD a;MATE!**
In MML, **CMPR:MHD=a:MATE;**

Example 6

This example compares, between two designated disks, the partitions specified with the PTN keyword in a specific sequence. The size of the partitions must match on both disks. Enter:

In PDS, **CMPR:MHD a,PTN (w,x,y,z);MHD b !**
In MML, **CMPR:MHD=a,PTN=w-x-y-z:MHD=b ;**

Example 7

This example compares two partitions of the same size on the same disk. Enter:

In PDS, **CMPR:MHD a,PTN x;MHD a,PTN y !**
In MML, **CMPR:MHD=a,PTN=x:MHD=a,PTN=y ;**

Correction Operation

The correction operation is similar to the compare operation. In the correction mode, when a mismatch is found, a message is generated and the specified disk block(s) is updated from a good source disk.

Example 8

This example shows how to compare a specified disk with its mate and perform the necessary corrections to the destination disk. This example assumes that a compare was run and that a correction was indicated. Enter:

In PDS, **COPY:DIFF;SRC,MHD a;DEST,MATE!**
In MML, **COPY:DIFF:SRC,MHD a:DEST,MATE;**

In this example, there is no default for source or destination. Both units must be specified.

Example 9

This example shows how to correct a specified unit other than the mate of the source disk. This example can be used for updating a backup copy of the source disk. Enter:

```
In PDS,   COPY:DIFF;SRC,MHD a;DEST,MHD b!  
In MML,   COPY:DIFF:SRC,MHD a:DEST,MHD=b;
```

In this example, the destination unit may not be a member of an active duplex pair. If the destination unit is active and has a mate, then the mate cannot be in the active state.

Example 10

This example shows how to correct a backup copy of a partition on the same disk. Enter:

```
In PDS,   COPY:DIFF;SRC,MHD a,PTN x;DEST,MHD a,PTN y!  
In MML,   COPY:DIFF:SRC,MHD a,PTN=x:DEST,MHD=a,PTN=y;
```

In this example, source and destination are on the same unit. As in Example 9, this unit may not be a member of an active duplex pair. If this unit is active and has an active mate, the mate must be removed from service before the command is invoked.

Stop Operation

The **STOP:CMPR;MHD** and **STOP:COPY;DIFF** commands both terminate the execution of a **CMPR:MHD** or **COPY:DIFF** command.

Example 11

To stop a read-only, compare, or correction operation, enter:

```
In PDS,   STOP:CMPR;MHD!  
In PDS,   STOP:COPY;DIFF!  
In MML,   STOP:CMPR:MHD;  
In MML,   STOP:COPY:DIFF;
```

The **STOP** commands respond with **CONFLICT WITH CURRENT SYSTEM STATUS** if there is no read-only, compare, or correct operation in progress.

The use of either of the **STOP** commands results in the termination of the current operation without a display of the status information.

Limitations of Disk Compare and Correct Commands

Only one invocation of a process is allowed to run at any one time. Commands respond with "RETRY LATER" if there is an operation in progress.

These commands cannot be run on any disk currently being reserved for some other maintenance activity.

For additional information on these operations and commands, refer to the following information products (IPs):

- 254-341-200, *Maintenance and Fault Recovery Software Subsystem Description 3B20D Computer*
- 254-303-110, *PDS Input Messages Manual*
- 254-303-111, *PDS Output Messages Manual*
- 254-303-112, *MML Input Messages Manual*
- 254-303-113, *MML Output Messages Manual*.

Procedure for Handling Disk Errors

When the disk driver receives a disk error from the DFC, it increments an error counter for the MHD. Depending on the options specified by the maintenance person, the disk driver may also generate a REPT DFCE or REPT DFCI output message, which is logged in the *DKDRVLOG* file. The following options can be specified by the maintenance person via Disk File System Access Page 120, Poke 677:

- 677,x n – Turn off Enhanced Information Report (EIR).
- 677,x l – Print long EIR format.
- 677,x s – Print short EIR format (default).

Where *x* is the DFC member number.

Every 4 hours, the MHD error counters are checked against the threshold value. In order to change the threshold value, the maintenance person should do the following:

- Access User-Level Automatic Restart Process (ULARP) record in ECD.
- Enter “dkdip” for the “rec_name” field.
- Set threshold by specifying an integer in the “proc_args” field.
- Kill “dkdip” process (ULARP will restart it).

Refer to Chapter 10, “Maintaining Software Integrity,” for ULARP. Applications should determine their own threshold value by trial and error and set it accordingly. When the threshold is exceeded and the MHD is in the ACT state, the “dkdip” process generates the following output message to inform the maintenance person about the MHD errors:

```
REPT DKDRV WARNING, MHD a EXCEEDS LONG TERM  
FAILURE THRESHOLD
```

To obtain information on the bad disk blocks, the maintenance person can execute the **OP:DFCELOG** command. The **OP:DFCELOG** command extracts and formats the short format of type 2222 of the REPT DFCE output messages from the *DKDRVLOG* file.

The information obtained can be used to update the MHD defect table for a fixed media disk drive that supports defect management.

- For SMD disks, refer to the “Writing SMD Disk Drive Defect Tables” section in Chapter 10.
- For SCSI disks, refer to the “Adding to the Defect Table of a SCSI Disk Drive” section in Chapter 10.

If it is desirable to reformat the disk at a later time, the threshold value can be increased. The REPT DKDRV WARNING message will not be printed until the error counters exceed the new threshold.

An MHD error counter can be reinitialized for any type of disk drive by executing the **CLR:MHD;MAEC** command.

For additional information, refer to the following IPs:

- 254-303-110, *PDS Input Messages Manual*
- 254-303-111, *PDS Output Messages Manual*
- 254-303-112, *MML Input Messages Manual*
- 254-303-113, *MML Output Messages Manual*.

Contents

Introduction	10-1
Checking Operating System Resources	10-1
Using Operating System Audits	10-4
Manipulating Processes	10-6
Maintenance Guide for ULARP	10-7
■ ULARP Records in the ECD	10-7
■ Database Modification Procedures	10-9
■ Initialization Procedures	10-9
System Bootstrap	10-9
Boot Progress PRMs	10-10
■ Error Reporting	10-12
■ Use of the OP:ULARP Command	10-15
■ Field Update Procedures for ULARP and Its Child Processes	10-17
Using CRON	10-18
Writing SMD Disk Drive Defect Tables	10-19
Adding to the Defect Table of a SCSI Disk Drive	10-23

Introduction

Maintaining the software integrity of the *UNIX*® Real-Time Reliable (RTR) operating system includes the following:

- Checking operating system resources.
- Using operating system audits.
- Manipulating processes.
- Using the User-Level Automatic Restart Process (ULARP), refer to the “Maintenance Guide for ULARP” section.
- Using Clock Daemon (CRON), refer to the “Using CRON” section.

Checking Operating System Resources

The operating system manages a limited set of resources and reports the depletion of these resources should this occur. Resources include such things as real time, message buffers, swap space, and memory.

The REPT SIMCHK 602 output message reports many types of overload situations. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for the various overload fault codes.

Some overloads occur during normal machine operation. An example of this is when pairs of overload/overload cleared messages and lockout/lockout cleared messages (fault codes 81 through 9A) occur. These messages can be tolerated if they occur occasionally. If they occur at a specific time or in concurrence with a specific event, report it to the next level of technical support. Report any regular occurrence to your technical assistance organization.

Take immediate action in response to overloads involving the following:

- Message buffers—codes AD, AE, and AF
- Kernel process memory—codes A8 and A9
- Disk swap space—code A6
- Segment descriptor table—A3 and A4
- Dispatcher control table—codes A0 and A1
- Module 1 memory—codes 9D and 9E.

When the 100 percent overload limit is reached, the computer takes automatic recovery actions. An off-line dump is required to determine the problem. Refer to the “Creating a Dump Tape” section of Chapter 3. Notify your technical assistance organization immediately when any of the previously mentioned overload messages occur so that a recovery plan can be developed and data can be saved for analysis.

The Message Buffer (MSGBUF) audit gives some indication of whether message buffers are being lost regularly. This audit frees lost message buffers and reports the number freed. This audit is initiated routinely by the System Integrity Monitor (SIM). The number of freed message buffers should always be small (usually zero). Notify your technical assistance organization if the number is regularly more than zero.

Kernel process memory, disk swap space, segment descriptor table entries, and dispatcher control table entries may become overloaded if process administration procedures are not functioning normally. The Process Administration (PROAD) audit detects and corrects process administration abnormalities. This audit is initiated routinely by SIM. A PROAD audit should rarely report an error. Notify your technical assistance organization if this audit begins reporting errors on a regular basis.

When the operating system memory manager detects disk swap space fragmentation, it compacts the swap space and generates an REPT MMGR008 or REPT MMGR009 output message for Generic 2 or *UNIX* RTR Operating System Release 5.0. (This condition is indicated by the REPT MMGR004 output message for releases after *UNIX* RTR Operating System Release 5.0.) The REPT MMGR008 message can be tolerated if it occurs occasionally. However, if any of the following should happen:

- REPT MMGR008 messages occur repeatedly.
- One or more REPT MMGR008 messages are followed within a short time by an REPT SIMCHK 602 message with overload fault code A6.
- A single REPT MMGR009 message occurs.

The swap space compaction audit should be invoked manually by entering the following messages:

In PDS, **AUD:MMGR 9!**
In MML, **AUD:MMGR=9;**

Using Operating System Audits

Operating system audits correct errors in operating system data structures and recover lost operating system resources. Most operating system audits are invoked periodically by the audit control subsystem according to their routine scheduling parameters in the Equipment Configuration Database (ECD). Routine audits produce no output messages unless they detect errors or terminate abnormally. The operating system audits may also be invoked manually when needed to troubleshoot and repair system software problems. Table 10-1 lists the commands that are used to invoke operating system audits.

Table 10-1. Operating System Audit Commands

Command	Description
AUD:CUMEM	Compares the memory of the active and standby 3B20D computer when running duplex. This audit is run only by manual request to troubleshoot hardware problems in the computers. (Refer to the “Interpreting Memory Failures” section in Chapter 4.)
AUD:CUSTAT	Verifies the control unit hardware status and the consistency of ECD information regarding the hardware status. It also performs a limited memory comparison between duplex processors.
AUD:ECD	Audits the structural integrity of the incore ECD and both disk copies, and verifies the integrity of the data in the incore ECD.
AUD:ECDOWN	Verifies the ownership of requests in the ECD manager request table.
AUD:FMGR	Audits various administrative structures within the operating system file manager, times out old file manager tasks, and closes files that had been opened by abnormally terminated processes.
AUD:FSBLK	Verifies the assignment of blocks to files within a file system and rebuilds the file system structures that keep track of free blocks and inodes.
AUD:FSLINK	Verifies the link integrity among the directories and inodes of a file system.

Table 10-1. Operating System Audit Commands (Contd)

Command	Description
AUD:MMGR	Verifies the link integrity among the operating system memory management tables, recovers lost memory segments, and compacts fragmented disk swap space. The swap space compaction audit is run only by manual request. (Refer to the "Checking Operating System Resources" section.)
AUD:MSGBUF	Audits process message queues and message buffer administration structures. Old queued messages are timed out and lost message buffers are recovered with the cooperation of their owner processes.
AUD:PMS	Verifies the structural integrity of the plant measurements database.
AUD:PROAD	Verifies the integrity of operating system process administration procedures, and terminates incompletely created or terminated processes (sometimes referred to as "zombies").

When the **AUD** command is entered, the **CORR** or **NCG** option may be used to specify whether the audit should run in correcting or noncorrecting mode. For detailed instructions on running each audit, refer to the following:

- 254-303-110, *PDS Input Messages Manual*.
- 254-303-112, *MML Input Messages Manual*.
- Chapter 14, "System Audits," in this IP.

Manipulating Processes

For processes that are not executed and monitored by the ULARP, use the **OP:STATUS:PROCESS** command to verify current execution status. For processes that are executed and monitored by ULARP, use the **OP:ULARP:PROCESS** command as described in the section “Maintenance Guide for ULARP.”

The commands listed in Table 10-2 can be used to start or stop the execution of a process. Since these commands can cause permanent changes in the execution status of system processes, they should only be used with concurrence of your technical assistance organization or as directed in a Broadcast Warning Message (BWM).

Table 10-2. Process Execution and Termination Commands

Command	Description
EXC:ENVIR:PROC	Executes a kernel or supervisor process.
EXC:ENVIR:UPROC	Executes a <i>UNIX</i> system level process.
STOP:EXC:ANY	Stops any process from executing. If the process is monitored by ULARP, it will be re-created automatically. (Refer to the “Maintenance Guide for ULARP” section.)
STOP:EXC:USER	Stops a user process specified by ID number or by pathname from executing. Use of this command to stop a monitored process will prevent it from being restarted automatically.

For details on any specific input messages and system responses, refer to the following information products (IPs):

- 254-303-110, *PDS Input Messages Manual*
- 254-303-112, *MML Input Messages Manual*.

Maintenance Guide for ULARP

The ULARP is a *UNIX* system process which is part of the software integrity subsystem. It is created by the SIM and, if it terminates, SIM attempts to restart it. Its functions are to set up the *UNIX* system environment, execute the commands during the bootstrap procedure, and create and monitor critical *UNIX* system processes which include user processes. If one of its child processes terminates, ULARP will determine why and, on most occasions, will attempt to restart the process.

ULARP Records in the ECD

The ULARP child processes and the run commands have been added to ECD records. Also, those processes formerly contained in the *cmon.p* file are included in ECD records. Figure 10-1 shows the *ularp* form for the ECD.

ularp		(1/1)
ULARP ECD Record		(Recent Change and Verify)
1.rec_name: _____	2.exec_no: _____	
3.child_priority: _____	4.boot_prm: _____	5.sleep: _____
6.utilid: _____	7.runtype: _____	8.run_config: _____
9.craft: _____	10.cruc_craft: _____	11.uniq_child: _____
12.wait_for_run: _____		
13.proc_path: _____		
14.proc_args: _____		
15.shell_cmd: _____		

Figure 10-1. ULARP ECD Record

The elements in Figure 10-1 are defined in the following list:

- **rec_name** — The name of the record can be up to 11 characters. This is a key field which will be used to retrieve a specific record. An example is when typing the user input commands **OP:ULARP:PROCESS** and **OP:ULARP:COMMAND**.
- **exec_no** — A key field which determines the order in which run commands and child processes are executed during a bootstrap or user initialization procedure.

- `child_priority` — The priority of a child process if different than normal UNIX system priority (50). If the priority is changed, it must also be changed in the specification file. If the two do not agree, the priority from the specification file will be used.
- `boot_prm` — To be used when a boot progress Processor Recovery Message (PRM) is desired as another means of identifying a process or command, the default value is zero.
- `sleep` — The number of seconds to delay before executing the next process or command in the boot sequence or during a user initialization.
- `utilid` — The utility identification of a child process, use zero for a run command.
- `runtype` — Either child process or run command, the values equal process or command.
- `run_config` — The configuration in which the process or command is to be executed, the values equal `min`, `full`, or `both`.
- `craft` — Any process that is terminated during a craft initialization procedure, the values equal `y` or `n`.
- `cruc_craft` — For user processes crucial to outputting error reports through the spooler instead of PRMs, the values equal `y` or `n`.
- `uniq_child` — To be used when only one copy of this process exists in the system; that is, it does not fork a child process that continues to run under the same utility identification, the values equal `y` or `n`.
- `wait_for_run` — A valid field for commands only. To be used when execution of that command must be completed before the next process or command is started, the values equal `y` or `n`.
- `proc_path` — The full pathname of the process should not exceed 63 characters.
- `proc_args` — Arguments to be passed to the process can be up to 9 but not to exceed 63 characters. A special argument type, "stypc," has been reserved for use by any process that needs to know whether a bootstrap or restart procedure is taking place. This argument is converted to the word "boot" when the process is executed during a bootstrap or craft initialization. It is converted to the word "restart" if that process has been restarted (after having been terminated for any reason other than a craft initialization).
- `shell_cmd` — A run command, to be executed by the UNIX system shell, should not exceed 79 characters. Any entries in those files which exceeded 79 characters were converted to shell scripts.

Database Modification Procedures

Special procedures must be followed for inserting, deleting, and modifying ULARP child process and run command records in the ECD. Four possible actions may cause ULARP database records to change. These actions are listed below along with the procedures to follow for each.

- Insertion of a new process or command: A new database record must be inserted, then, in order to execute the new process or command, the **INIT:ULARP** command must be entered.
- Deletion of a process or command: The database record must be deleted, then, if the process is running, it must be killed.
- Change in execution parameters of a process or command: If the process is running, make the change to the database record and then kill it using the **STOP:EXC:ANY** command. ULARP will automatically restart the process. If the process is not running or the command did not execute properly, make the change and use the **INIT:ULARP** command to execute it.
- Change in the order of execution of a process or command: The *exec_no* field is a key field, and a special procedure must be followed. It is not expected that this will ever be necessary in the field, but it could be done. Delete the database record, then reinsert it with the new value of the *exec_no* field. Kill the process and use the **INIT:ULARP** command to restart it.

Initialization Procedures

System Bootstrap

During a bootstrap procedure, SIM creates the *UNIX* system *init* process which executes */prc/ularp*. When SIM receives the acknowledgment message from the process manager indicating the successful creation of *init*, it logs the process number and waits for the start-up event from ULARP. ULARP will inherit the process number assigned to *init*.

Once it has been created, ULARP will attach to the ECD and loop through its records forking and executing all child processes and run commands which indicate that they are to be run in that configuration. They are executed in the order indicated in the *exec_no* field of their database records. Child processes are executed directly by ULARP, while the run commands are executed by */bin/sh*. As each process and command is executed, a boot progress PRM is output if the *boot_prm* field of its ECD record is nonzero.

Boot Progress PRMs

Boot progress PRMs have been instituted for ULARP child processes and the run commands so that applications will know that the system is coming up.

Some boot progress PRMs may get lost as a result of the TN11 input parameter buffer not being able to process them as fast as they are being issued by the various boot programs.

The following is the format for all PRMs used by ULARP:

EBda *ffxx yyyy yyyy hh hh hh*

Where:

<i>ff</i>	equals the function code of the PRM.
<i>xx</i>	equals the <i>boot_prm</i> field of the ECD record in hexadecimal. For example, operating system user processes contain c1 through c5, diagnostic processes and commands begin with d, run commands begin with b, and other processes begin with e.
<i>yyyy yyyy</i>	equals the utility identification of the process or the record identification of the command in hexadecimal.
<i>hh hh hh</i>	equals hardware status digits.

The PRMs function code of 60 hexadecimal indicates that bootstrap processes and commands are being executed.

A complete list of PRM function codes used by ULARP is provided in the “Error Reporting” section.

INIT:ULARP — When the user shell command **INIT:ULARP** is entered on the maintenance terminal if ULARP is not running, SIM starts it. ULARP then attempts to adopt its former child processes; if the adopt fails, it attempts to restart them. If ULARP is running, it will attempt to start any processes that are not executing or it will attempt to run commands that previously failed.

The **INIT:ULARP** command should be used to restart any child process (including a single-user process) that terminated and was not restarted automatically by ULARP. It should also be used to execute a run command that failed. In either case, the reason for the failure should be found and corrected before using the **INIT:ULARP** command.

The ULARP determines which processes to restart or which run commands to reexecute by looking at the values stored in the status fields of their database records.

The following status values indicate that a process or command should be restarted after special action is taken to correct the problem.

NON_EXEC	This requires action only if the process should have been executed in the current configuration. Look in the <i>run_config</i> field of its ECD record to see if it is supposed to be running.
FORK_FAIL	ULARP failed 10 times in attempting to fork a new process to execute this child process or run command.
EXECFAIL	Process execution failed.
UFATAL	A running process exited because of a fatal error and was not restarted by ULARP.
0 <= value <= 0xff	The exit status of a run command that failed or a process that terminated and was not restarted by ULARP.

Craft init — Selection of the  key (15) results in the termination of all user processes and will cause SIM and/or ULARP to do one of two things.

1. If ULARP is not running, SIM will restart it. Then ULARP will restart user processes and adopt its other running child processes. If an adopt fails, enter the **INIT:ULARP** command.
2. If ULARP is running, it will only start user processes.

This procedure is more drastic than an **INIT:ULARP** and should only be used if craft interface is not responding normally to craft input messages.

In either case, boot progress PRMs will be generated for user processes only. Three levels of user process termination can be invoked from the  key (15). Each level would require a parameter provided by first poking the 42 on the emergency action page. The parameters are 1, 2, and 3 for each level, respectively. Level 1 adds the termination of all user processes of the craft class. Level 2 results in the termination of user processes classified as non-essential killable. Level 3 adds the termination of all killable *UNIX* system processes.

Error Reporting

ULARPLOG Log file — The output message REPT ULARP is the ULARP error and reinitialization reporting mechanism. If the output spooler is running, it will write these messages to the log file, *ULARPLOG*, and to the printer. If the spooler is not running, ULARP will write directly to the log file and generate a PRM.

Some messages contain only a statement of the action taken by ULARP, while others include the process name and process identification of the process that was acted upon. Refer to 254-303-111, *PDS Output Messages Manual*, and 254-303-113, *MML Output Messages Manual*, for a complete list of REPT ULARP message components.

The following messages are produced by ULARP to inform the user of actions that have been taken. No action needs to be taken in response to these messages. PRM function codes are in parentheses; a complete list appears in Table 10-3.

- REPT ULARP FORK FAILED THEN SUCCEEDED
- REPT ULARP ATTEMPTING TO RESTART (0x60)
- REPT ULARP ATTEMPTING TO RE_EXECUTE (0x60)
- REPT ULARP ATTEMPTING CRAFT INIT
- REPT ULARP CHILD FOUND WITHOUT ACCOMPANYING SIGNAL
- REPT ULARP CRAFT INIT COMPLETED
- REPT ULARP CHILD EXEL ABORTED DUE TO DIOP

The following messages indicate database access problems. If any of these messages occur, contact your technical assistance organization.

- REPT ULARP ECD ACCESS FAILED (0x64)
- REPT ULARP OPEN SEQUENCE FAILED (0x64)
- REPT ULARP UPDATE FAILED

The following messages report error conditions that require the user to take action to correct the conditions:

- REPT ULARP FORK FAILED TEN TIMES (0x65 or 0x66)
A symptom of an internal system error, possibly an overload condition. If the error is repeated when the **INIT:ULARP** command is entered, contact your technical assistance organization.
- REPT ULARP CHILD TERMINATED EXECFAIL (0x66)
- REPT ULARP CHILD TERMINATED STATFAIL
There is either a problem with the *pfile* of the process (it is nonexistent, mutilated, or does not have execute permissions), or the database record contains the incorrect pathname and is not immediately detected by ULARP.
- REPT ULARP CHILD TERMINATED UFATAL (0x67)
The child process has detected an error that prevents it from continuing to run. Look for preceding error messages generated by the process itself.
- REPT ULARP NO CHILD PROCESSES EXIST (0x69)
All child processes either failed to execute or terminated UFATAL. Look for preceding error messages and/or PRMs indicating reasons for failure.
- REPT ULARP TOO MANY ARGUMENTS (0x68)
The ECD record of the process contains invalid information and must be corrected using recent change and verify.
- REPT ULARP RUN COMMAND FAILED (0x65)
A completion code other than 0 (success) was returned by the shell. This completion code has been stored in the ECD record and may be displayed by using the **OP:ULARP:COMMAND** input message.
- REPT ULARP PROCESS RESTARTED TOO MANY TIMES (0x67)
The process has been restarted ten times after terminating with an exit status other than EXECFAIL or UFATAL. ULARP has stopped attempting to restart the process to prevent the system from thrashing. The exit status may be displayed by using the **OP:ULARP:PROCESS** input message. Look for preceding error messages for clues as to why the process will not keep running.
- REPT ULARP ADOPT FAILED (0x66)
This is a symptom of an internal system problem. If the process is not running, enter the **INIT:ULARP** command.

To find the procedures for correcting these errors, refer to the REPT ULARP command in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MLL Output Messages Manual*. After the problems have been corrected, enter the **INIT:ULARP** command to restart the process or to reexecute the command named in the output message.

The **OP:LOG:ULARPLOG** and **DEL:LOG:ULARPLOG** commands are used to display and delete information from the log file.

Error PRMs — Boot progress PRMs will always remain PRMs, but error report PRMs will become ordinary spooler output messages as soon as critical craft interface processes are running. The spooler will write ULARP error messages into the log file. If the spooler is not running, ULARP will write directly to the log file and output error report PRMs through the SIM.

The error PRM function codes include those shown in Table 10-3.

Table 10-3. PRM Function Codes

Code	Description
0x60	In addition to being used in boot progress PRMs, this code is output if the spooler is not running and ULARP is attempting to restart a process.
0x61	SIM could not create the <i>UNIX</i> system <i>init</i> process.
0x63	ULARP was not executed successfully by the <i>UNIX</i> system <i>init</i> process.
0x64	ULARP could not attach to the ECD.
0x65	A run command failed to execute.
0x66	A child process failed to execute.
0x67	A child process died UFATAL.
0x68	A record contained invalid information.
0x69	All child processes failed to execute or died UFATAL.

⇒ NOTE:

Actions to be taken in response to each function code are specified on its PRM manual page.

Use of the OP:ULARP Command

This section describes the procedure for using the **OP:ULARP** user command to query ULARP database records for status information on its child processes and the run commands.

Input Commands — The following input commands may be used:

OP:ULARP:EXECSEQ To output the execution sequence of child processes and run commands during a bootstrap or craft initialization process.

OP:ULARP:PROCESS To output the status of one or all ULARP child processes.

OP:ULARP:COMMAND To output the status of one or all run commands.

The following formats are for the **OP:ULARP** commands:

In PDS, **OP:ULARP:EXECSEQ {BOOT | CFTINIT}!**
In MML, **OP:ULARP:DATA,EXECSEQ={BOOT | CFTINIT};**

Where **BOOT** is the order of execution during a bootstrap and **CFTINIT** is the order of execution during a craft initialization.

In PDS, **OP:ULARP:PROCESS {ALL | a}!**
In MML, **OP:ULARP:DATA,PROCESS={ALL | a};**

Where *a* is the name of the ECD record for the child process.

In PDS, **OP:ULARP:COMMAND {ALL | a}!**
In MML, **OP:ULARP:DATA,COMMAND={ALL | a};**

Where *a* is the name of the ECD record for the run command.

A complete description of the procedure for using the input commands is found in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Output Messages — The following output messages will cause ULARP status information to be displayed:

OP ULARP EXECSEQ	Prints the order of execution of child processes and run commands during a bootstrap or craft initialization. Fields include execution sequence number, name of the ECD record, whether it is a process or command, the <i>boot_prm</i> field, the record ID or utility ID, the delay timing field (in seconds), and the <i>wait-for-completion</i> field for commands only (yes or no).
OP ULARP PROCESS	Prints the status of one or all child processes. Fields include the name of the ECD record, the process id, the status of the process, the date and time the status field last changed, the restart count if nonzero, and the date and time ULARP started counting restarts if that field is nonzero.
OP ULARP COMMAND	Prints the status of one or all run commands. Fields include the name of the ECD record, the status of the command, and the date and time the status field last changed.

Field Update Procedures for ULARP and Its Child Processes

When field updating ULARP or one of its child processes, the **STOP:EXC:ANY** command is used to terminate the process. SIM will restart ULARP or ULARP will automatically restart the child process.

The Control/Display Update Page (103) is used as an alternate procedure to terminate ULARP, Coordinator of Spooler Output Processes (CSOP), DAP, or RTS. (Refer to Figure 10-2.)

```
S & B      3B20D      R6.1.24      ttyc-cdC TTY 00 06/18/91 12:32:23 CDT
SYS EMER   CRITICAL   MAJOR      MINOR    BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH      CU      CU PERPH  0S LINKS
CMD< _      103 - C/D UPDATE
```

<u>CMD</u>	<u>FUNCTION</u>	<u>CMD</u>	<u>FUNCTION</u>
800	- UPDATE C/D GLOBAL MENU	806	- RESTART CIA
801	- UPDATE C/D STATE TRANSLATION	807	- RESTART POKER(S)
802	- RESTART ULARP	808	- RESTART CFTSHL(S)
803	- RESTART CSOP		
804	- RESTART RTS	810	- SCC COLOR TERMINAL
805	- RESTART DAP	811	- SCC B/W TERMINAL

```
A  REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / MML ***
<
```

Figure 10-2. Control/Display Update Page

Using CRON

CRON is a process that schedules other processes to run at a date and time specified in the file */usr/lib/crontab*. CRON schedules processes to run every half hour, such as OP OOS, and Routine Exercise (REX) runs every evening.

The file *crontab* consists of lines with six fields each. The fields are separated by either spaces or tabs. The first five fields are integer patterns that specify the minute (0 through 59), hour (0 through 23), day of month (0 through 31), month of year (1 through 12), and day of week (0 through 6, Sunday equals 0). Numbers within a field separated by a minus sign (–) are used to indicate an inclusive range. An asterisk (*) is used to indicate “any legal value.” The sixth field is a command that is executed by the shell at the specified times. A percent character (%), which translates as a new line character, is used to separate the command from the remainder of the string. The remainder of the string is used by the command as standard input.

Each time the file */usr/lib/crontab* is read, an entry is made in the file */etc/cronlog*. The entry includes date and time of the read and a copy of the command executed as a result of the read. To print the contents of either *crontab* or *cronlog*, use the following command:

```
In PDS,  DUMP:FILE:ALL, FN "a"!
In MML,  DUMP:FILE:DATA,ALL,FN="a";
```

Where *a* equals */etc/cronlog* to print *cronlog* or */usr/lib/crontab* to print *crontab*.

Writing SMD Disk Drive Defect Tables

The following procedure is used to write the Manufacturer's Defect Table for 160- or 340-MB disk drives.

Contact your technical support organization for assistance before performing this procedure. This procedure requires the paper defect map supplied with the disk drive.

Use the following commands to write the defect table for Moving Head Disk (MHD) a from the defect map provided with the disk. If the cylinder and head number of a defect on the defect map provided with the disk drive are the same as the cylinder and head number of the previous defect, DO NOT enter that defect. Enter only those defects approved by technical support. There may be flaws not on the paper defect list which were detected by the REX and/or the verify command which are to be included in the defect table.

Step	Procedure
1	Remove any existing file having the same name as the file to be created; enter: In PDS, CLR:FILESYS:FILE, FN "/tmp/MHDa"! In MML, CLR:FILESYS:DATA,FILE, FN="/tmp/MHDa"; Where a equals MHD member number. Wait for a completion message. ⇒ NOTE: The file to be deleted may not exist.
2	Enter: In PDS, IN:FILE:APND,FN "/tmp/MHDa",LINE n! In MML, IN:FILE:DATA,APND,FN="/tmp/MHDa",LINE=n; Where a equals MHD member number and n equals the line number beginning with zero.
3	Enter: "x y z" Where x equals cylinder number, y equals head number, and z equals sector number as provided in the defect map provided with the disk drive or as provided from other valid sources (for example, REX). Wait for a completion message. Repeat Steps 2 and 3, replacing n in Step 2 with next line number for the next defect map line and replacing x y z in Step 3 with data from next approved flaw.
4	If MHD a is active, remove MHD a from service; enter: In PDS, RMV:MHD a! In MML, RMV:MHD=a; Where a equals MHD member number. Wait for a completion message.

Step	Procedure
5	Load MHD a with the new defect table; enter: In PDS, LOAD :MHD a;DEFECT:FN "/tmp/MHDa" In MML, LOAD :MHD=a:DEFECT:DATA,FN="/tmp/MHDa"; Where a equals MHD member number. Wait for a completion message.
6	Verify that the MHD a defect list was written into the Manufacturer's Defect Table; enter: In PDS, DUMP:MHD a;DEFECT:MFGR! In MML, DUMP:MHD=a:DEFECT:DATA,MFGR; Where a equals MHD member number. The Manufacturer's Defect Table will be printed.
7	Format the disk using the newly loaded Manufacturer's Defect Table; enter: In PDS, INIT:MHD a:NEW! In MML, INIT:MHD=a:DATA,NEW; Where a equals MHD member number. Wait for a completion message.
8	Verify that MHD a's combined defect table was updated as a result of the FORMAT command; enter: In PDS, DUMP:MHD a;DEFECT:ALL! In MML, DUMP:MHD=a:DEFECT:DATA,ALL; Where a equals MHD member number. The combined defect table will be printed as part of the output data.
9	Verify that there are no hard errors remaining on MHD a; enter: In PDS, VFY:MHD a! In MML, VFY:MHD=a; Where a equals MHD member number. Wait for a completion message.

Step	Procedure
10	If the message has any verify failures, add the failing cylinder, head and sector to the paper defect list. Repeat Steps 1 through 10. ⇒ NOTE: Flaws are listed in ascending sequence by cylinder, head and sector (lowest numbered cylinder, head and sector first to highest numbered cylinder, head and sector last).
11	Remove the file used to update the Manufacturer's Defect Table; enter: In PDS, CLR:FILESYS:FILE, FN "/tmp/MHDa"! In MML, CLR:FILESYS:DATA,FILE, FN="/tmp/MHDa"; Where a equals MHD member number. Wait for a completion message.
12	Restore MHD a to service; enter: In PDS, RST:MHD a! In MML, RST:MHD=a; Where a equals MHD member number. Wait for a completion message.

Adding to the Defect Table of a SCSI Disk Drive

⇒ NOTE:

This procedure does not apply to the KS23908, L40 disk drive because the drive does automatic reallocation of defective blocks found during read or write operations.

If the LOAD MHD DEFECT command is executed for the KS23908, L40 drive, the command will not be started and will return an error code indicating that defect management is controlled by the drive itself and not with the LOAD MHD DEFECT command.

This procedure can be used to add defects to the combined (GROWN) defect table of a Small Computer System Interface (SCSI) disk drive.

Contact your technical support organization for assistance before performing this procedure.

Use the following commands to add defects to the defect table for MHD a. (a is the MHD member number.)

Step	Procedure
1	Make a list of the defective blocks to be added to the defect list. The block numbers should be in ascending sequence (lowest block number first to highest block number last). Also, only use block numbers obtained since last successful "defect adding" procedure.
2	Remove any existing file having the same name as the file to be created; enter: In PDS, CLR:FILESYS:FILE,FN "/tmp/MHDa"! In MML, CLR:FILESYS:DATA,FILE,FN="/tmp/MHDa"; Where a equals MHD member number. Wait for a completion message.

⇒ NOTE:

The file to be deleted may not exist.

Step	Procedure
3	Enter: In PDS, IN:FILE:APND,FN "/tmp/MHDa",LINE n! In MML, IN:FILE:DATA,APND,FN="/tmp/MHDa",LINE=n; Where <i>a</i> equals MHD member number and <i>n</i> equals the line number beginning with zero.
4	Enter: "w" Where <i>w</i> equals defective block number in the list of defective blocks generated in Step 1. Wait for a completion message. Repeat Steps 3 and 4, replacing <i>n</i> in Step 3 with the next line number for the next defect map line and replacing <i>w</i> in Step 4 with the data for the next defective block from the list of defective blocks. When all the defective blocks are put into the file, record the count of the number of defective blocks.
5	If MHD <i>a</i> is active, remove MHD <i>a</i> from service; enter: In PDS, RMV:MHD a! In MML, RMV:MHD=a; Where <i>a</i> equals MHD member number. Wait for a completion message.
6	Load MHD <i>a</i> with the new defect blocks; enter: In PDS, LOAD:MHD a;DEFECT:FN "/tmp/MHDa"! In MML, LOAD:MHD=a;DEFECT:DATA,FN="/tmp/MHDa; Where <i>a</i> equals MHD member number. Wait for a completion message. Verify that the number of defects which were loaded is the same as the number of defects specified in the input file. Contact your technical support organization.

Step	Procedure
7	Format the disk using the updated defect list; enter: In PDS, INIT:MHD a In MML, INIT:MHD=a Where a equals MHD member number. Wait for a completion message.
8	Remove the file used to update the Manufacturer's Defect Table; enter: In PDS, CLR:FILESYS:FILE, FN "/tmp/MHDa" In MML, CLR:FILESYS:DATA,FILE, FN="/tmp/MHDa"; Where a equals MHD member number. Wait for a completion message.
9	Verify that there are no defects remaining on MHD a; enter: In PDS, VFY:MHD a! In MML, VFY:MHD a; Where a equals MHD member number. Wait for a completion message.
10	If the message has any verify failures, replace the defect list generated in Step 1 with the new list of verify failures. Repeat Steps 3 through 10.
	⇒ NOTE: The defect list used in Step 1 cannot be reused since the defect information changes whenever defects are loaded into a SCSI disk.
11	Restore MHD a to service; enter: In PDS, RST:MHD a! In MML, RST:MHD=a; Where a equals MHD member number. Wait for a completion message.

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers ***UNIX***[®] RTR Operating System

System Maintenance Manual
Volume 2

254-303-106
Issue 1
February 2002



Copyright © 1998 Lucent Technologies. All Rights Reserved.

This information product (IP) is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity, including other Lucent Technologies Business Units or Divisions, without express written consent. For permission to reproduce or distribute, call 1-888-LTINFO6 (1-888-584-6366).

Notice

Lucent Technologies is the successor to the business and assets of AT&T Network Systems Business Unit. Every effort was made to ensure that the information contained in this IP was complete and accurate at the time of printing. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes, and you are cautioned that the hardware, software, features, and capabilities described herein may not be the same as the product you have.

Trademarks

5ESS is a registered trademark of Lucent Technologies.

AUTOPLEX is a registered trademark of Lucent Technologies.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

WE is a registered trademark of Lucent Technologies.

All other trademarks are identified as they appear in the text.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana. Orders for additional copies may be placed from the continental United States by calling 1-888-LUCENT8 (1-888-582-3688) or faxing to 1-317-322-6484. Customers outside the continental United States may place orders by contacting the in-country Lucent Technologies market manager. Lucent Technologies personnel outside the continental United States can order additional copies by faxing to 1-317-322-6699.

Information Product Support Telephone Number

Call 1-888-LTINFO6 (1-888-584-6366) to report errors or to ask questions about this IP.

Acknowledgment

Developed by Lucent Technologies Network Systems Customer Training and Information Products.

Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 2

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiph hotline@lucent.com



Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 2

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiph hotline@lucent.com



Contents

	About This Information Product	xxxi
	■ Purpose	xxxi
	■ Audience	xxxi
	■ Reason for Issue	xxxii
	■ How to Use This Information Product	xxxiii
	■ Conventions Used	xxxvi
	■ Safety Labels	xxxviii
	■ Related Information Products	xxxix
	■ How to Comment on This Information Product	xl
	■ Mandatory Customer Information	xl

1	Hardware Identification	1-1
	■ Introduction	1-1
	■ 3B20D Model 1 Computer Hardware	1-4
	■ 3B20D Model 2 Computer Hardware	1-18
	■ 3B20D Model 3 Computer Hardware	1-36
	■ 3B21D Computer Processor Cabinet	1-52

2	Maintenance Plan	2-1
	■ Introduction	2-1
	■ Site Maintenance Plan	2-2
	■ Automatic Recovery Actions	2-3
	■ Manual Recovery Actions	2-4
	■ Preventive Maintenance	2-4
	■ Responding to Alarms	2-6
	■ Documentation	2-17
	■ Using Automatically Generated Data for Troubleshooting	2-17
	■ Switching Control Center Interface	2-18
	■ Training	2-18
	■ Field Support	2-18

Contents

3	Recovery	3-1
	■ Introduction	3-1
	■ Responding to Automatic Recovery Actions	3-1
	■ Recovering the System Manually	3-8
	■ Using Processor Recovery Messages	3-33
	■ Responding to a Terminal Suspend	3-33
	■ Isolating Emergency Action Page Problems	3-45
	■ Recovery Using Software Backup Disk and Alternate Boot Disk Options	3-51
	■ Recovering from DIOP	3-59
	■ Loading Disk from Tape	3-85

5	Routine Exercise Procedures	5-1
	■ Introduction	5-1
	■ Purpose of REX	5-1
	■ REX Operation	5-1
	■ REX Scheduling	5-2
	■ REX Scheduling Recommendations	5-2
	■ REX Manual Commands	5-3
	■ REX Inhibits	5-7
	■ REX Inhibit Recommendations	5-8
	■ REX Inhibit Implementations	5-9
	■ REX Summary	5-12

6	Diagnostics	6-1
	■ Introduction	6-1
	■ Maintenance Input Request Administrator	6-3
	■ Automatic Diagnostics	6-7
	■ Performing Diagnostics	6-8

Contents

■ Synchronous Data Link Restore Tool	6-54
■ Locating Pins	6-57
■ Processor Recovery Messages	6-59

7	Off-Line Boot	7-1
	■ Introduction	7-1
	■ Using OFLBOOT	7-3
	■ Performing OFLBOOT	7-4
	■ Checking OFLBOOT Results	7-17
	■ Examples of Performing OFLBOOT	7-20

8	Log File Review	8-1
	■ Introduction	8-1
	■ Using Log Files	8-3
	■ Core Dump Administration	8-5

9	File-System Maintenance	9-1
	■ Introduction	9-1
	■ Checking File-System Resources	9-1
	■ Manipulating Files	9-2
	■ Using File-System Audits	9-5
	■ File Corruption Detection (SIV Only)	9-9
	■ Boot File Verification Through OFLBOOT	9-10
	■ System Backup	9-11
	■ Copying to Backup Partitions	9-13
	■ Making a Backup Disk	9-26
	■ Making a Backup Tape Copy of a System Disk	9-34
	■ Initialize Disk Partition with Zeros	9-36
	■ Copying Files to Nonactive Disks	9-37
	■ Comparing and Correcting Disk Data	9-39

Contents

■ Procedure for Handling Disk Errors	9-44

10	Maintaining Software Integrity	10-1
■	Introduction	10-1
■	Checking Operating System Resources	10-1
■	Using Operating System Audits	10-4
■	Manipulating Processes	10-6
■	Maintenance Guide for ULARP	10-7
■	Using CRON	10-18
■	Writing SMD Disk Drive Defect Tables	10-19
■	Adding to the Defect Table of a SCSI Disk Drive	10-23

11	System Update	11-1
■	Introduction	11-1
■	System Update Procedures	11-1
■	SUPR Description	11-5
■	System Update Menu Page Description	11-9
■	System Update Session Examples	11-15
■	Troubleshooting System Update Procedures	11-54
■	Input Messages	11-59

Contents

12	Craft Interface	12-1
	■ Introduction	12-1
	■ Maintenance Interface	12-1
	■ Maintenance Procedures	12-4

13	Field Update	13-1
	■ Introduction	13-1
	■ Update Methods and Processes	13-1
	■ Broadcast Warning Messages	13-6
	■ Messages Files Execution and Display	13-22
	■ Field Update Session Example	13-34
	■ Input Messages	13-73
	■ System Generation Database	13-74

14	System Audits	14-1
	■ Introduction	14-1
	■ System Integrity Subsystem	14-3
	■ Procedures for Modifying Audit Records	14-38
	■ Operating-System Audits	14-50

15	Input/Output Processor Magnetic Tape Unit	15-1
	■ Introduction	15-1
	■ Using the Magnetic Tape System	15-3
	■ Operating, Maintaining, and Diagnosing the Magnetic Tape System	15-4
	■ Writing a File from a Disk	15-7

Contents

16	SC Status and SD Point Operation	16-1
	■ Introduction	16-1
	■ SCSD ECD Records	16-10
	■ Capabilities and User Requirements	16-18
	■ SCSD Commands	16-62
	■ Growth and Degrowth Procedures	16-64
	■ Verifying SCSD Operation	16-76

17	Spare Circuit Packs	17-1
	■ Introduction	17-1
	■ General Guidelines	17-1

18	Maintenance Tools	18-1
	■ Introduction	18-1
	■ Field Test Set	18-1
	■ GRASP/EGRASP	18-3
	■ Software Debugging Tools	18-4

19	SCSI Peripheral System	19-1
	■ Introduction	19-1
	■ Firmware vs. Pumpcode	19-2
	■ Verify the Integrity of Pumpcode	19-4
	■ Installing New Pumpcode	19-5
	■ SCSI Disk Drives	19-7
	■ SCSI Tape Drives	19-9

Contents

20	Magnetic Tape Special Device Filenames	20-1
	■ Introduction	20-1
	■ Tape Device <i>logdev</i> Records	20-1
	■ IOP Magnetic Tape <i>logdev</i> Records (3B20D Computer Only)	20-2
	■ SCSI/IOP Backward Compatible <i>logdev</i> Records	20-4
	■ SCSI <i>logdev</i> Records	20-5

21	Software Update	21-1
	■ Introduction	21-1
	■ Software Update Types	21-2
	■ Space Reclamation	21-12
	■ Error Conditions	21-12
	■ Enhanced SU Capabilities	21-14
	■ SU Installation	21-16
	■ SU Maintenance	21-20
	■ Routine Operations and Maintenance Procedures	21-24
	■ Page 1940 Easy SU Installation	21-44
	■ Page 1941 SU Automation Scheduling	21-49
	■ Page 1942 SU Automation Office Profile	21-53
	■ Page 1943 SU Automation Healthcheck	21-56
	■ Page 1950 SU Maintenance	21-58
	■ Page 1960 SU Installation	21-61

22	DSCH Computer Interconnect Feature	22-1
	■ Introduction	22-1
	■ Installation Procedure	22-5
	■ Maintenance Commands for DCI	22-10
	■ SG/ECD Setup	22-12

Contents

23	UUCP Feature	23-1
	■ Introduction	23-1
	■ Scope of the UUCP Feature Chapter	23-2
	■ Supported/Unsupported UUCP Commands	23-3
	■ Feature Activation/Deactivation	23-4
	■ Operation of UUCP	23-10
	■ UUCP Administration	23-18
	■ UUCP Security	23-39
	■ Troubleshooting	23-41
	■ Error Code Interpretation	23-45
	■ UUCP Administrative Files	23-63
	■ UUCP Manual Pages	23-65

A	3B20D Computer References	A-1
	■ Introduction	A-1
	■ Control Unit Diagnostic Test Sequence	A-2
	■ Diagnostic Hierarchy	A-4
	■ Test Procedures	A-6
	■ Supplementary Phases	A-11
	■ Main Store Failure Data	A-23
	■ Processor Recovery Messages	A-29
	■ Diagnostic Information Tables — CU, DFC, and IOP	A-30

B	3B21D Computer References	B-1
	■ Introduction	B-1
	■ Control Unit Diagnostic Test Sequence	B-2
	■ Diagnostic Hierarchy	B-4
	■ Test Procedures	B-6
	■ Supplementary Phases	B-13
	■ Main Store Failure Data	B-26

Contents

- Processor Recovery Messages B-29
- Diagnostic Information Tables — CU, DFC,
IOP, and DCI B-30

Glossary

GL-1

Index

IN-1

Figures

1

Hardware Identification

1-1.	Typical 3B20D Model 1 Computer Functional Block Diagram	1-5
1-2.	3B20D Model 1 Computer Control Unit Frame	1-7
1-3.	3B20D Model 1 Computer Peripheral Control Frame	1-10
1-4.	3B20D Model 1 Computer 300-MB Moving Head Disk Frame	1-13
1-5.	3B20D Model 1 Computer SCSI Disk Cabinet	1-15
1-6.	3B20D Model 1 Computer Power Distribution Frame	1-17
1-7.	Typical 3B20D Model 2 Computer Functional Block Diagram (SMM Release 1)	1-19
1-8.	Typical 3B20D Model 2 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-20
1-9.	Typical 3B20D Model 2 Computer Functional Block Diagram (VLMM Release 6 Retrofit)	1-21
1-10.	3B20D Model 2 Computer Processor Control Frame	1-22
1-11.	3B20D Model 2 Computer 300-MB Moving Head Disk Frame	1-28
1-12.	3B20D Model 2 Computer Tape/Disk Cabinet	1-30
1-13.	3B20D Model 2 Computer SCSI Disk Cabinet	1-32
1-14.	3B20D Model 2 Computer 160-MB Minimodule Disk Frame	1-34
1-15.	3B20D Model 2 Computer Power Distribution Frame	1-35
1-16.	Typical 3B20D Model 3 Computer Functional Block Diagram (SMM Release 1)	1-37

Figures

1-17.	Typical 3B20D Model 3 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-38
1-18.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SMD DFC Unit	1-39
1-19.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SCSI DFC Unit	1-40
1-20.	3B20D Model 3 Computer Processor Cabinet	1-41
1-21.	3B20D Model 3 Computer Tape/Disk Cabinet	1-48
1-22.	3B20D Model 3 Computer SCSI Disk Cabinet	1-50
1-23.	3B20D Model 3 Computer Power Cabinet	1-51
1-24.	3B21D Computer Processor Cabinet	1-53
1-25.	3B21D Computer Growth Cabinet	1-56

2

Maintenance Plan

2-1.	3B20D Computer TN2, TN3, TN5, and TN6 Power Switches Faceplate	2-10
2-2.	3B21D Computer UN373 (CU Power Switch) Faceplate	2-12
2-3.	3B21D Computer UN375 (CU Power Switch) Faceplate	2-13
2-4.	3B21D Computer UN376 (CU Power Switch) Faceplate	2-14

Figures

3

Recovery

3-1.	Recovery Procedure (System Configuration)	3-30
3-2.	Recovery Procedure (Application Failure)	3-31
3-3.	Recovery Procedure (Post Recovery)	3-32
3-4.	MTTY/3B20D Computer Interface (Model 1)	3-47
3-5.	MTTY/3B20D Computer Interface (Model 2 or Model 3)	3-48
3-6.	MTTY/3B21D Computer Interface with a TN983 MTTYC	3-49
3-7.	MTTY/3B21D Computer Interface with a UN583 MTTYC	3-50
3-8.	Poke 120 — DFC Status Page	3-62

Figures

6

Diagnostics

6-1.	Sample Output from OP:DMQ	6-20
6-2.	Sample Diagnostic Failure	6-26
6-3.	Sample TLP Circuit Pack List	6-30
6-4.	Sample Diagnostic Listing	6-39
6-5.	Sample Output from CU DGN Audit	6-45
6-6.	Sample Output from Non-DGN CU Audit	6-51
6-7.	Backplane Pin Set	6-58

10

Maintaining Software Integrity

10-1.	ULARP ECD Record	10-7
10-2.	Control/Display Update Page	10-17

11

System Update

11-1.	System Update Procedure	11-2
11-2.	System Update Menu Page Screen	11-11
11-3.	Enter Arguments Page 1 Screen	11-13
11-4.	Enter Arguments Page 2 Screen	11-14
11-5.	System Update Menu Page	11-16

12

Craft Interface

12-1.	Maintenance Interface Configuration	12-3
12-2.	MTTY Terminal Critical Indicator Display	12-17
12-3.	Emergency Action Page for TN83	12-20
12-4.	Emergency Action Page for TN983	12-21

Figures

12-5.	Emergency Action Page for UN583 or UN597 without Alternate Boot Disk Feature (pre R21.35)	12-22
12-6.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (pre R21.35)	12-23
12-7.	Emergency Action Page for UN597 without Alternate Boot Disk Feature (R21.35 and later)	12-24
12-8.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (R21.35 and later)	12-25
12-9.	Lock Screen Display for UN597	12-26
12-10.	Page Index for RTR Release 21.3 and Earlier	12-37
12-11.	Page Index for RTR Release 21.5 Through Release 21.15	12-38
12-12.	Page Index for RTR Release 21.17 and Later	12-39
12-13.	Status Summary Area Page	12-43
12-14.	Common Processor Display Page	12-45
12-15.	Common Processor Display Page in DIOP	12-46
12-16.	Common Processor Display Page — MTTY and ROP States	12-47
12-17.	Common Processor Display Page — On-Line	12-48
12-18.	Common Processor Display Page — Off-Line	12-49
12-19.	3B20D Computer Common Processor Display Page with CU FAN Alarms	12-50
12-20.	3B21D Computer Common Processor Display Page with CU FAN Alarm	12-51
12-21.	3B21D Computer Common Processor Display Page with ABD Feature	12-53
12-22.	Control/Display Update Page	12-58
12-23.	Operating System Status Page	12-60
12-24.	CFTMTC Display Page 105	12-65
12-25.	CFTMTC Display Page 105 with Menu 912 Data	12-66

Figures

12-26.	CFTMTC Display Page 106	12-67
12-27.	System Update Menu Page	12-74
12-28.	DCI Status Page	12-76
12-29.	DCI Status Page for RCC	12-77
12-30.	DCI Status Page with DSMLNK	12-78
12-31.	Field Update Page	12-83
12-32.	Typical DFC Index Page	12-89
12-33.	Typical DFC Status Page for a System Equipped with Storage Module Drive (SMD) DFCs 0 and 1	12-91
12-34.	Typical DFC Status Page for a System Equipped with SMD DFCs 2 and 3	12-92
12-35.	Typical DFC Status Page for a System Equipped with SCSI DFCs 0 and 1	12-93
12-36.	Typical DFC Status Page for a System Equipped with SCSI DFCs 2 and 3	12-94
12-37.	Typical DFC Status Page for a System, in Full DIOP, Equipped with SCSI DFCs 0 and 1	12-95
12-38.	Typical DFC Performance Page for a System Equipped with SMD DFCs 0 and 1	12-102
12-39.	Typical DFC Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-103
12-40.	Typical DFSA Index Page	12-106
12-41.	Typical DFSA Status Page for a System Equipped with SMD DFCs 0 and 1	12-108
12-42.	Typical DFSA Status Page for a System Equipped with SMD DFCs 2 and 3	12-109
12-43.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 0 and 1	12-110
12-44.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 2 and 3	12-111

Figures

12-45.	Typical DFSA Status Page for a System, in FULL DIOP, Equipped with SCSI DFCs 0 and 1	12-112
12-46.	Typical DFSA Performance Page for a System Equipped with SMD DFCs 0 and 1	12-118
12-47.	Typical DFSA Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-119

Figures

13

Field Update

13-1.	BWM File Structure	13-8
13-2.	BWM SCANS File	13-10
13-3.	BWM HEADER File	13-11
13-4.	BWM MSGS File (MML Format)	13-14
13-5.	Update Package States	13-18
13-6.	Field Update Page	13-23

14

System Audits

14-1.	Audit-Control Subsystem	14-4

16

SC Status and SD Point Operation

16-1.	Software Structure	16-3
16-2.	Three Levels of Software Interfacing SCSD Hardware	16-4
16-3.	Sample scsdbody Form	16-10
16-4.	3B20D Computer SCSD Point Assignments to Logical Groups Examples	16-12
16-5.	3B21D Computer SCSD Point Assignments to Logical Groups Examples	16-13
16-6.	Structure of the SCSD Option Block in the ECD	16-15
16-7.	UN33 Block Diagram	16-21
16-8.	UN933 Block Diagram	16-22
16-9.	Administrator Message Structure	16-32
16-10.	Error Codes for Administrator and Client Processes	16-34
16-11.	ECD Option Block Record	16-35
16-12.	Power Switch	16-40

Figures

16-13.	3B20D Computer Scan Point Transition Diagram	16-43
16-14.	PSM Data Structure	16-49

17 Spare Circuit Packs

17-1.	Circuit Pack Inventory and Testing Log	17-2

19 SCSI Peripheral System

19-1.	Tape Path Components	19-15
19-2.	Cleaning Points	19-19
19-3.	UN376 (NCR006-3503341), UN376C (NCR006-3300608/Comcode 407545243), and UN376E (KS-24367, L1/Comcode 407771260) SCSI DAT Drives — Front View	19-20
19-4.	UN376 and UN376B Series 1 and 2 Circuit Pack Layout	19-25
19-5.	UN376B Series 3 and Later Circuit Pack Layout	19-26
19-6.	Cassette Write Protect	19-27
19-7.	Tape Load Diagram	19-35

20 Magnetic Tape Special Device Filenames

20-1.	IOP Tape Device Partition Number Descriptor	20-2
20-2.	SCSI/IOP Tape Device Partition Number Descriptor	20-4
20-3.	SCSI Tape Device Partition Number Descriptor	20-6

Figures

21 Software Update

21-1.	Software Update Distribution	21-3
21-2.	Software Update Structure	21-5
21-3.	Example of a PDS Message File	21-6
21-4.	Software Update Page 109	21-15
21-5.	Page 1960 SU Installation Menu	21-17
21-6.	Typical 3B20D/3B21D Computer On-Line SU History	21-20
21-7.	Page 1940 Easy SU Installation Menu	21-45
21-8.	Page 1941 SU Automation Scheduling Menu	21-50
21-9.	Page 1942 SU Automation Office Profile Menu	21-53
21-10.	Page 1943 SU Automation Healthcheck Menu	21-57
21-11.	Page 1950 SU Maintenance Menu	21-58
21-12.	Page 1960 SU Installation Menu	21-62

22 DSCH Computer Interconnect Feature

22-1.	Software Architecture of the DCI Feature	22-3
22-2.	Hardware Architecture of the DCI Feature	22-4
22-3.	Example of DCI ucb Form	22-18
22-4.	Example of HA-OMP DCI ucb Form	22-20
22-5.	Example of HA-OMP Adjunct ucb Form	22-22
22-6.	Example of Normal Operations mdct Form	22-24
22-7.	Example of Diagnostic mdct Form	22-25
22-8.	Example of HA-OMP Adjunct mdct Form	22-26

Figures

22-9.	Example of Diagnostic <i>logdev</i> Form	22-27
22-10.	Example of Non-HA-OMP <i>logdev</i> Form	22-27
22-11.	Example of HA-OMP <i>logdev</i> Form	22-27
22-12.	Example of <i>chopt</i> Form	22-28
22-13.	Example of <i>dciopt</i> Form	22-28
22-14.	Example of <i>device</i> Form	22-29
22-15.	Example of <i>classdef</i> Form	22-30
22-16.	Example of <i>errthld</i> Form	22-31

A 3B20D Computer References

A-1.	BOT and EOT Marker Placement	A-22
A-2.	Sample Memory Failure Histogram	A-25

B 3B21D Computer References

B-1.	Sample Memory Failure Histogram	B-27



Tables



About This Information Product

1. IPs Supporting 3B21D Computer	xxxix



1 Hardware Identification

1-1. 3B20D Model 2 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-25
1-2. 3B20D Model 3 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-44



3 Recovery

3-1. Initialization Levels	3-2

Tables

5

Routine Exercise Procedures

5-1.	Early REX Inhibits	5-10
5-2.	New REX Inhibits	5-11

6

Diagnostics

6-1.	Diagnostic Hierarchy Appendix Index	6-8
6-2.	Suggested Routine Test Procedures	6-11
6-3.	EX Commands	6-13
6-4.	Diagnostic Results	6-27
6-5.	Explanation of Test Skipped Reasons	6-28
6-6.	Meaning of TLP Notes	6-31
6-7.	DIAMON Error Codes	6-43
6-8.	DGN CU Error Codes	6-46
6-9.	Selected System Error Codes	6-48
6-10.	Maintenance Channel Status	6-50
6-11.	DGN Audit IOP, DFC, and DCI Error Codes	6-52

7

Off-Line Boot

7-1.	Options for the EXC:OFLBOOT Input Message	7-6
7-2.	Precheck Override Conditions	7-9
7-3.	Secondary DFC Override Condition	7-9

Tables

9

File-System Maintenance

9-1.	File-System Maintenance Commands (Routine)	9-3
9-2.	File-System Maintenance Commands (Troubleshooting)	9-3
9-3.	File-System Maintenance Commands (Permanent)	9-4

10

Maintaining Software Integrity

10-1.	Operating System Audit Commands	10-4
10-2.	Process Execution and Termination Commands	10-6
10-3.	PRM Function Codes	10-14

12

Craft Interface

12-1.	KS-22921, L1 Maintenance Terminal Configuration	12-5
12-2.	KS-23554, L5 and L6 Maintenance Terminal Configuration	12-6
12-3.	KS-23996, L1 and L5 Maintenance Terminal Configuration	12-8
12-4.	Model 5310 ROP Option Selection	12-12
12-5.	Model 577 ROP Option Selection	12-13
12-6.	Model 602 ROP Option Selection	12-15
12-7.	System Critical Indicators	12-18
12-8.	Emergency Action Page Commands	12-27
12-9.	Emergency Action Page Display Indicators	12-32
12-10.	Emergency Action Page Poke Prompts	12-34

Tables

12-11.	Common Processor Display Page Commands Without ABD Feature	12-52
12-12.	Common Processor Display Page Commands with ABD Feature	12-54
12-13.	Parameters and Options for CFTMTC Commands	12-68
12-14.	Display Page 105 Commands, Parameters, and Options for Menu Numbers	12-71
12-15.	Display Page 106 Commands, Parameters, and Options for Menu Numbers	12-72
12-16.	DCI Status Page Commands	12-79
12-17.	Field Update Scope/Action Combinations	12-85
12-18.	DFC Status Page Commands	12-98
12-19.	DFC Performance Page Menu Commands	12-104
12-20.	Disk File System Access Status Page Menu Commands	12-115
12-21.	DFSA Performance Page Menu Commands	12-120

13

Field Update

13-1.	Allowed Scope/Action Combinations	13-27

14

System Audits

14-1.	Audit Families and Audit Names	14-6
14-2.	Audit Names and Member Numbers	14-8
14-3.	Audit Abort Codes	14-23
14-4.	REPT SIMCHK Error Codes	14-27
14-5.	Optional Hexadecimal Data for REPT SIMCHK Error Conditions	14-31

Tables

14-6.	CUMEM Audit Error Numbers	14-53
14-7.	CUMEM Error Reports	14-53
14-8.	CUSTAT Audit Error Numbers	14-60
14-9.	CUSTAT Error Reports	14-61
14-10.	AUD CUSTAT Error Messages	14-67
14-11.	ECD Audit Error Messages	14-76
14-12.	LLA Audit Error Messages	14-79
14-13.	ECDOWN Audit Error Messages	14-97
14-14.	FMGR Audit Errors	14-104
14-15.	FMGR Audit Error Reports	14-107
14-16.	FSBLK Audit Errors	14-136
14-17.	Special Error Messages	14-137
14-18.	FSLINK Audit Errors	14-147
14-19.	Special Error Messages	14-148
14-20.	FSCMPT Audit Errors	14-161
14-21.	Special Error Messages	14-162
14-22.	MMGR Audit Error Messages	14-171
14-23.	MSGBUF Audit Error Messages	14-193
14-24.	Special Error Messages	14-195
14-25.	PMS Audit Error Messages	14-208
14-26.	PROAD Audit Errors	14-219

15 Input/Output Processor Magnetic Tape Unit

15-1.	Magnetic Tape System Input Commands	15-3
15-2.	MTC Diagnostic Commands	15-6

Tables

16 SC Status and SD Point Operation

16-1.	Phase Alarm Levels	16-31
16-2.	3B20D Computer Logical Group Assignments Examples	16-37
16-3.	3B21D Computer Logical Group Assignments Examples	16-38
16-4.	3B20D Computer Scan Point Transitions and Actions	16-42
16-5.	3B21D Computer Scan Point Transitions and Actions	16-44
16-6.	3B20D Computer Scan Point and Signal Distributor Point Assignments	16-53
16-7.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Large Office	16-54
16-8.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Small Office	16-55

19 SCSI Peripheral System

19-1.	Cleaning Schedules and Equipment	19-13
19-2.	Cleaning Schedule Guidelines	19-16
19-3.	Description of Configuration Option Byte 15 Bit Settings	19-23
19-4.	DAT SCSI Device Identification Switch Settings	19-26
19-5.	KS-23909 9-Track Tape Capacity	19-28
19-6.	DAT Cartridge Tape Capacity	19-29
19-7.	NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT Drive Indicators	19-40
19-8.	NCR006-3503341, L1 3.5-Inch SCSI DAT Drive Indicators	19-41

Tables

20 Magnetic Tape Special Device Filenames

20-1.	IOP 9-Track Operating Parameters	20-2
20-2.	Typical <i>logdev</i> Values	20-7

21 Software Update

21-1.	File Transfer Error Conditions (SCANS/CSCANS Interface)	21-13
21-2.	Predefined SU Automation Healthcheck Conditions	21-56

23 UUCP Feature

23-1.	ED3T076-20 G57 and G58 Signal Connections	23-11
23-2.	UUCP Troubleshooting Corrective Action	23-41
23-3.	Error Codes and Definitions	23-50
23-4.	uucheck Error Messages	23-54
23-5.	uucheck Warning Messages	23-57
23-6.	uucheck Caution Messages	23-60
23-7.	uucheck Internal Error Messages	23-62

A 3B20D Computer References

A-1.	Recommended Frequency of Selected Supplementary Phases	A-11
A-2.	Supplementary Phases for CU	A-12
A-3.	Supplementary Phases for SMD DFC	A-18
A-4.	Supplementary Phases for SCSI DFC	A-19
A-5.	Supplementary Phases for IOP	A-20

Tables

A-6.	Partitioning of Address Spectrum Among Memory Arrays	A-27
A-7.	Application Channel Interface	A-31
A-8.	Central Control	A-32
A-9.	Cache Storage Unit	A-36
A-10.	Enlarged Cache Storage Unit	A-37
A-11.	Disk File Controller	A-38
A-12.	SCSI DFC	A-39
A-13.	Direct Memory Access	A-40
A-14.	Direct Memory Access Channel	A-41
A-15.	Dual Serial Channel	A-43
A-16.	Direct User Interface Controller	A-44
A-17.	Input/Output Processor	A-45
A-18.	Main Store Controller	A-46
A-19.	Default Refresh Data Parity Check Times for MASC Phase 96	A-47
A-20.	Moving Head Disk	A-48
A-21.	Magnetic Tape Controller	A-49
A-22.	Maintenance Teletypewriter Controller	A-49
A-23.	Store Address Translator	A-50
A-24.	Serial Channel	A-50
A-25.	Scan Point and Signal Distributor Controller	A-51
A-26.	Synchronous Data Link Controller	A-51
A-27.	Teletypewriter Controller	A-51
A-28.	Utility Circuit	A-52

B 3B21D Computer References

B-1.	Recommended Frequency of Selected Supplementary Phases	B-14
B-2.	Supplementary Phases for CU	B-15
B-3.	Supplementary Phases for DFC 2	B-21
B-4.	Supplementary Phases for DFC 3	B-22
B-5.	Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape	B-23

Tables

B-6.	Supplementary Phases for IOP 2	B-24
B-7.	Central Control	B-31
B-8.	Main Store Controller	B-34
B-9.	Direct Memory Access	B-36
B-10.	Direct Memory Access Channel	B-37
B-11.	Disk File Controller 2	B-38
B-12.	Disk File Controller 3	B-39
B-13.	SCSI Magnetic Tape	B-40
B-14.	Utility Circuit	B-41
B-15.	Input/Output Processor 2	B-42
B-16.	Dual Serial Channel Computer Interconnect	B-43

About This Information Product

Purpose

The System Maintenance Manual is an information product (IP) that is part of a documentation set used to support the *UNIX*® Real-Time Reliable (RTR) operating system running on Lucent Technologies 3B20D and 3B21D computers. It summarizes the corrective and preventive maintenance procedures used to operate and maintain the 3B20D Model 1, 2, and 3 computers and the 3B21D computer. The procedures range from recovering the system to analyzing the error conditions and performing diagnostics. Software diagnostic capabilities are also discussed.

Other topics are discussed that may not be considered a part of routine maintenance, such as the set up and operation of the maintenance terminal, performing updates, running audits, and the use of peripheral equipment.

Audience

This IP provides information for a variety of audiences. Anyone interested in the *UNIX* RTR operating system, 3B20D/3B21D computers, or a summary of corrective and preventive maintenance will find it useful. The primary audience for this IP includes the following:

- Hardware, software, or training course developers
- Craft or maintenance personnel
- 3B20D/3B21D computer system administrators or documentation users.

Reason for Issue

This IP is being issued to incorporate new information for 3B20D/3B21D computers and to bring the IP into compliance with new Lucent Technologies documentation standards. The major changes are as follows:

- Change the six-digit Select Code number for the *System Maintenance Manual* from 304-046 to the new nine-digit number 254-303-106.
- Add information on alarms and responding to alarms in Chapter 2, and remove references to the UN375C disk pack in Figure 2-3.
- Add circuit pack information for UN582 and UN583 in Chapters 3 and 12.
- Add SOURCE field p and h postmortem descriptions in Chapter 4.
- Add DFC to Tables 6-1, 6-2, and 6-5.
- Add TLP Note 04 for CC, CSU, DMA, and MASC in Table 6-6.
- Add new error codes and meaning to Table 6-11.
- Revise the “Copying to Backup Partitions” procedure in Chapter 9.
- Add new figures associated with the poke commands in Chapter 12.
- Add circuit pack information for UN933 in Chapter 16.
- Add circuit pack information for UN376E in Chapter 19.
- Update Appendixes A and B.

Change bars on the far right-hand side of the text are used to denote changes made to the current issue. Change bars will help you understand at a glance what has changed from the previous issue. The position of deleted text is shown by a single asterisk (*). Not all editorial changes will be marked with change bars. Occasionally, change bars do not work in tables or figures; therefore, those changes will be shown by adding a change bar to only the table or figure title.

Whenever this IP is reissued, the reason(s) for reissue will be listed in this paragraph.

How to Use This Information Product

This IP is composed of independent chapters and reference information (Table of Contents, Appendixes, Glossary, and Index). Specific topics are referenced in the Table of Contents and Index with page number locations. The Glossary defines the terminology used in this IP. This IP is organized in three volumes, as follows:

Volume 1

- Master Table of Contents
- About This Information Product
Provides general information about the IP. Information regarding audience, reason for issue, conventions, safety labels, how to order, and how to comment is included. A description of how the IP is organized and a brief description of the contents of each chapter is also provided.
- Chapter 1 — Hardware Identification
Describes the hardware in the computers and contains illustrations that show the location of specific units, circuit packs, components, and associated frames.
- Chapter 2 — Maintenance Plan
Describes the elements in the plan for maintaining the computers.
- Chapter 3 — Recovery
Provides the following recovery procedures: responding to automatic recovery actions, recovering the system manually, using processor recovery messages, responding to a terminal suspend, isolating emergency action page problems, recovering from a duplex outage, recovering from Disk Independent Operation (DIOP), and loading disk from tape.
- Chapter 4 — Error Analysis
Gives an overview of the Control Unit (CU) error messages and automatic postmortem dumps. It will help the user interpret the different types of error messages generated by the automatic fault recovery features.
- Chapter 5 — Routine Exercise Procedures
Provides a description of the Routine Exercise (REX) processes and procedures. Included are detailed descriptions and examples of the REX craft interface and recommendations for scheduling and inhibiting REX.
- Chapter 6 — Diagnostics
Gives general information for using the diagnostic capabilities of the computers, how and when to perform diagnostics, how to interpret the output, and how to use the information to clear hardware problems.

- **Chapter 7 — Off-Line Boot**
Covers the Off-Line Boot (OFLBOOT) feature for the *UNIX* RTR Operating System Release 21.1 and later. OFLBOOT is used primarily to verify the capability to boot essential files without interrupting the system operation.
- **Chapter 8 — Log File Review**
Gives general information about the history of certain automatic and manual maintenance actions that occur in the system. These histories are located in the */etc/log* directory.
- **Chapter 9 — File-System Maintenance**
Gives procedures for checking file-system resources, manipulating files, using the file-system audits, file-system corruption detection, and backing up and restoring file systems.
- **Chapter 10 — Maintaining Software Integrity**
Gives procedures for checking operating system resources by using operating system audits, the User-Level Automatic Restart Process (ULARP), the Clock Daemon (CRON), and manipulating processes.

Volume 2

- **Master Table of Contents**
- **About This Information Product**
- **Chapter 11 — System Update**
Describes system update concepts and defines procedures for implementing updates. This chapter also provides examples of successful system updates and how to back out and restore an unsuccessful update. It describes other processes and utilities related to performing system updates and lists the system update input messages with a short description of their function.
- **Chapter 12 — Craft Interface**
Provides an overview of the software required to perform the various craft interface functions. The information presented summarizes operation of the Maintenance Teletypewriter (MTTY) terminal.
- **Chapter 13 — Field Update**
Describes field update concepts and defines procedures for implementing updates. This chapter also describes Broadcast Warning Messages (BWMs) and gives some rules for applying BWMs. It also describes BWM components and states.

- Chapter 14 — System Audits
Describes the system audits available to the user and the software integrity subsystem (the system feature that monitors the operating system software reliability and stability). This chapter also provides procedural information dealing with system audits, audit output messages and error reports, and system faults.

Volume 3

- Master Table of Contents
- About This Information Product
- Chapter 15 — Input/Output Processor Magnetic Tape Unit
Describes the magnetic tape system components and gives guidelines for its operation, use, and maintenance.
- Chapter 16 — SC Status and SD Point Operation
Provides information on how to write client processes, communicate with the Scan Point and Signal Distributor Handler (SCSDH), physically connect scan points, develop messages for maintenance personnel, grow and degrow UN33/UN933 boards, and grow and degrow the Input/Output Processor (IOP).
- Chapter 17 — Spare Circuit Packs
Gives guidelines for testing and rotating spare circuit packs.
- Chapter 18 — Maintenance Tools
Describes the available maintenance tools for the 3B20D/3B21D computers: the Field Test Set (FTS) and the Generic Access Package (GRASP)/Enhanced Generic Access Package (EGRASP).
- Chapter 19 — SCSI Peripheral System
Provides information which is helpful in maintaining the Small Computer System Interface (SCSI) peripheral system.
- Chapter 20 — Magnetic Tape Special Device Filenames
Describes the magnetic tape special device filenames.
- Chapter 21 — Software Update
Provides a description of program changes that are possible within the database.
- Chapter 22 — DSCH Computer Interconnect Feature
Provides information on the Dual Serial Channel (DSCH) Computer Interconnect (DCI) feature. The feature is implemented in the RTR Release 21.3 and later software. This chapter presents an overview of the DCI feature, configuring the Equipment Configuration Database (ECD), installation procedures for the *Sun** system, and maintenance commands for the DCI board.

* Sun, Sun Microsystems, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

- Appendix A — 3B20D Computer References
Provides 3B20D computer reference information on the following topics:
CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures;
Supplementary Phases; Main Store Failure Data; Processor Recovery
Messages; and the CU, Disk File Controller (DFC), and IOP Diagnostic
Information Tables.
- Appendix B — 3B21D Computer References
Provides 3B21D computer reference information on the following topics:
CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures;
Supplementary Phases; Main Store Failure Data; Processor Recovery
Messages; and the CU, DFC, IOP, and DCI Diagnostic Information
Tables.
- Glossary
- Index.

Conventions Used

Command, Filename, and Display Notations

The following notations are used to show commands and filenames in the text and displays.

NOTE:

System filenames and command names are case-sensitive, so you must enter them exactly as they are shown.

- Command names in text appear in **bold** type; for example, the **/usr/bin/ls** command. Command names in headings appear in **bold** type.
- Filenames and form names in text appear in *italic* type; for example, the */usr/lib/uucp/System* file or the *scsdbody* form. Filenames and form names in headings appear in ***bold-italic*** type.
- Text that you enter, such as a command or response to a prompt, appears in **bold** type; for example, the **ls -la** command. The convention of making text that you enter **bold** overrides other conventions; for example, enter **scsdbody** for the form name.
- Variables that appear in a command line or file appear in *italic* type; for example, **grep *username* /etc/passwd**. In this example, *username* is a variable indicating a user's name is required.

- Screen displays and system messages appear in constant width type; for example, Please enter your password. Program code listings and file listings are also shown in constant width type. Input messages are shown in **constant-width bold** type.
- Comments and explanations within a display are indented and shown in *italic* type. These are for information only and will not appear on your screen.
- A line in a file or on the computer screen that is too long to be shown as it actually appears in this IP will be shown with a backslash (\) at the end of the first line. This indicates the next line should be read as a continuation of the current line.
- Square brackets around an argument on a command line indicate that the argument is optional; for example, the **lpstat [-t]** command. In this example, the **-t** argument is optional and can be omitted.
- A vertical bar (|) between words in an argument on a command line indicates that one of the arguments is to be selected.
- The key identified on your keyboard as Return, Enter, or a bent arrow (↵) is referred to as the Return key. Occasionally, representations of this Return key will be boxed; for example, Return.
- There is an implied Return at the end of each command and menu response that you enter. Some examples do not explicitly show the Return. Where you may be expected to enter a Return (as in the case where you are accepting a menu default), the symbol <CR> is shown to indicate that you are to press the Return key.
- Key combinations appear in a hyphenated format; for example, Ctrl-d. Press and hold down the first key of a key combination while pressing the second key.
- Ellipses (three dots) on a command line indicate that the previous argument can be repeated; for example, **ls [file ...]**. In this example, multiple files can be listed after the command.
- References to manual pages are followed by their manual page location number in parentheses; for example, **mount(1M)**.

Hexadecimal Notation

Hexadecimal (base 16) numbers are denoted with a **0x** prefix; for example, 0x00A is decimal 10.

Signal Designations

The term “asserted” is used in the descriptions to mean that a signal is driven to its active state. The term “negated” is used in the descriptions to mean that a signal is driven to its inactive state.

Signal names used in the diagrams and descriptions that end in a 0 or 1 indicate the active state of the signal. Names ending in “0” are “active low” signals; names ending in “1” are “active high” signals. For example, CCIOD(31-00)1 describes the Central Control Input/Output Data bits 31 through 00 which are “active high” signals.

Equipment Locations

A coordinate numbering system is used to identify Equipment Locations (EQLs) in units and cabinets. The origin is the lower left front of the cabinet or unit. Vertical increments are measured in inches. Horizontal increments are measured in eighths of an inch. The coordinate location of a circuit pack is expressed as the horizontal and vertical location of the center lines of the connector into which the circuit pack is inserted. The location of a unit in a cabinet is identified by the placement of the lower left corner of the unit in the cabinet.

For example, a connector at EQL 004-080 is located 4 inches above the origin and 10 inches ($80 \times 0.125 = 10.0$) to the right of the origin.

Safety Labels

Safety labels are reminders used to assure the safety of personnel and to minimize service interruptions, loss of data, and damage to equipment, products, and software.

Three types of safety labels are used in Lucent Technologies documentation. The three types, in descending order of priority, are as follows:

1. **DANGER** indicates the presence of a hazard that **will** cause death or severe personal injury if the hazard is not avoided.
2. **WARNING** indicates the presence of a hazard that **can** cause death or severe personal injury if the hazard is not avoided.
3. **CAUTION** indicates the presence of a hazard that **will** or **can** cause minor personal injury or property damage if the hazard is not avoided.

This IP contains safety labels in the form of CAUTION statements.

Related Information Products

Table 1 lists by IP number and Select Code the Lucent Technologies IPs supporting the 3B21D computer. IPs with Select Codes will be converted to nine-digit IP numbers as they are reissued.

Table 1. IPs Supporting 3B21D Computer

NEW IP NUMBER	OLD SELECT CODE	TITLE
254-303-100	-	3B21D Computer Growth/Retrofit Tasks
254-303-101	-	3B21D Computer Routine Maintenance Tasks
254-303-102	-	3B21D Computer Trouble Clearing Tasks
254-303-103	303-007	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Processor Recovery Messages Guide
254-303-104	303-010	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Recent Change and Verify Manual
254-303-105	304-045	3B21D Computer Hardware Reference Manual
254-303-106	304-046	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System System Maintenance Manual
254-303-107	303-072	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Software Troubleshooting Guide
254-303-110	303-080	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Input Messages Manual
254-303-111	303-081	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Output Messages Manual
254-303-112	303-082	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Input Messages Manual
254-303-113	303-083	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Output Messages Manual

IP references will not include full titles; for example, "Refer to 254-303-105, *Hardware Reference Manual*, for more information."

How to Comment on This Information Product

Lucent Technologies welcomes your comments on this IP. Your comments will aid us in improving the quality and usefulness of Lucent Technologies documentation. Please use the Feedback Form provided at the front of this IP. If the Feedback Form is missing, mail your comments to the following address:

Lucent Technologies
Customer Training and Information Products (CTIP)
2400 Reynolda Road
Winston-Salem, NC 27106-4606

Mandatory Customer Information

The 3B21D computer is used as the Administrative Module (AM) in various switching system applications and is not provided as a stand-alone product. Therefore, the application documentation is responsible for providing this information.

Refer to the applicable application documentation for "Mandatory Customer Information."

Contents

Introduction	11-1
System Update Procedures	11-1
■ System Update Steps	11-2
■ System Update Methods	11-3
Backup Partition Method	11-3
Off-Line Disk Method	11-4
SUPR Description	11-5
■ SUPR Processes	11-5
ENTER Request	11-5
PROCEED Request	11-6
COMMIT Request	11-6
BACKOUT Request	11-6
RESTORE Request	11-6
■ SUPR Utilities	11-6
CONTINUE Utility	11-7
STOP Utility	11-7
READLOG Utility	11-7
■ Conditional and Unconditional Execution Mode	11-8
Conditional Execution Mode	11-8
Unconditional Execution Mode	11-8
System Update Menu Page Description	11-9
■ System Update Menu Page Screen	11-10
■ Enter Arguments Page 1 Screen	11-12
■ Enter Arguments Page 2 Screen	11-13
System Update Session Examples	11-15

Contents

■ Successful System Update Session Example	11-15
Display System Update Menu Page	11-16
Specify ENTER Command	11-17
Specify Update Method	11-18
Specify DFC Preference	11-19
Specify Update Backup Partitions Option	11-20
Specify Conditional/Unconditional Mode	11-21
Specify Tape Device	11-22
Display Enter Arguments Page 2 Screen	11-23
Specify First Disk to Be Updated	11-24
Specify First Disk Sequence Option	11-25
Specify Second Disk to Be Updated	11-26
Specify Second Disk Sequence Option	11-27
Specify Third Disk to Be Updated	11-28
Specify Third Disk Sequence Option	11-29
Return to System Update Menu Page	11-30
Execute ENTER Command	11-31
Mount Tape for Disk 770	11-32
Continue Update for Disk 770	11-33
Mount Tape for Disk 772	11-34
Continue Update for Disk 772	11-35
Mount Tape for Disk 775	11-36
Continue Update for Disk 775	11-37
ENTER Process Terminates	11-38
Execute PROCEED Command	11-39
PROCEED Process Terminates	11-40
Boot to New Generic	11-40
Execute COMMIT Command	11-41
COMMIT Process Terminates	11-42
Terminate System Update Menu Page	11-43
■ System Update BACKOUT Example	11-44
Execute BACKOUT Command	11-44
Boot to Old Generic	11-45
Execute the RESTORE Command	11-45
■ System Update Utilities Examples	11-47
Execute STOP Command	11-47
Execute READLOG Command	11-48
■ Unconditional Execution Example	11-49
Specify UCL Argument with Command Line Argument	11-50

Contents

Add UCL Argument to Specified Command Line Argument	11-51
Remove UCL Argument from Specified Command Line Argument	11-51
■ Application Dependent Processing Example	11-52
■ RETRO Argument	11-53
Troubleshooting System Update Procedures	11-54
Input Messages	11-59

Introduction

This chapter describes system update concepts and defines procedures for implementing updates. The procedural information defines the five processes used to execute the update procedure, the three utilities that provide external interactions with those processes, and the two executional modes used to install the system update.

Also provided in this chapter are examples of successful system updates, examples of how to back out and restore an unsuccessful update, and descriptions of other processes and utilities related to performing system updates, such as the RETRO argument and troubleshooting procedures.

This chapter also lists the system update input messages and gives a short description of their function.

System Update Procedures

The system update facilities provide a safe and reliable means for the operator to apply new versions of the *UNIX*® Real-Time Reliable (RTR) operating system and application software while minimizing service disruption. These facilities allow installation of the *UNIX* RTR operating system by one or more partitions, or in its entirety.

System Update Procedure (SUPR) software supports system updates by providing a separate process for each step of the update procedure including a means to reverse the update and reestablish the original system.

System Update Steps

The SUPR is broken down into several distinct steps. This allows the operator to have direct control over the procedure. This control enables the operator to choose the best time to begin each successive step of the update and to cancel the update at any step of the procedure. Figure 11-1 diagrams the steps used in performing a system update.

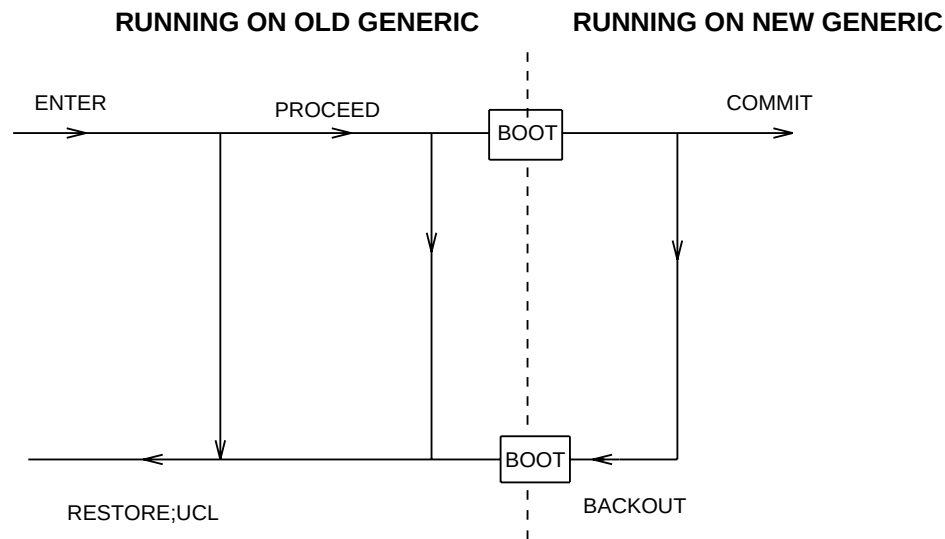


Figure 11-1. System Update Procedure

Under favorable conditions only the following forward steps of SUPR would be utilized, resulting in a successful update:

1. **ENTER:** Installs the new generic while running on the old generic.
2. **PROCEED:** Makes final preparations prior to booting the system from the new generic.
3. **BOOT:** Manually boots the system using the new generic.
4. **COMMIT:** Completes propagation of the new generic into the system after the soak period.

If the new generic is not workable after booting on the new generic, the operator will not commit to the new generic. Instead, the following backout procedure would be started to replace the original system:

1. **BACKOUT:** Makes preparations needed to boot the system from the old generic.
2. **BOOT:** Manually boots the system using the old generic.
3. **RESTORE:** Restores the old generic by removing the new generic from the system.

The operator can halt any update step in progress. However, if SUPR is halted, some manual intervention may be necessary before further update processing can occur. The operator can also restart an aborted or halted update step (except for some noncontinuable errors).

If during the **PROCEED** or **ENTER** step it becomes necessary to abort the SUPR, take the following action instead of continuing to the boot step.

Restore Old Generic: Unconditionally **RESTORE** the old generic by removing the new generic from the system.

System Update Methods

The SUPR supports two update methods: backup partition method and off-line disk method. Using the backup partition method, only those partitions having backups may be updated. Using the off-line disk method, any number of partitions may be updated.

Backup Partition Method

The basis of the backup partition method is that each disk partition containing generic data is "backed up" by another partition containing the same data (unrelated to the duplication associated with duplex disk). One copy is designated as the primary and the other copy as the backup partition. The two partitions form a primary/backup partition pair.

At any given time, only one copy of a partition is in use. The new generic can be written into the inactive partitions resulting in one set of old generic partitions and one set of new generic partitions. These two sets become the same again after completion of the update procedure. When this method is used, no new partitions can be added. Also, partitions cannot be changed in size or location. The format of the disk must remain the same. An example of a successful update using the backup partition method is as follows:

1. **Enter New Generic:** Reads the new generic data into the unused partitions after determining which set of partitions are unused.

2. **Proceed With New Generic:** Prepares for booting from the new generic partitions.
3. **Boot From New Generic Partitions:** Boots system from the new generic partitions using the Emergency Action Interface (EAI).
4. **Commit to New Generic:** Copies the new generic partitions into the old generic partitions resulting in both copies containing the new generic.

Off-Line Disk Method

The off-line disk method involves writing new generic data onto one disk of a duplex disk pair. For each duplex disk pair updated, one disk is removed from normal operation and new generic data is written onto that disk. Upon completion of the update procedure, each pair of duplex disks contain the new generic. The off-line disk method relies upon the existence of duplex disk pairs.

The off-line disk method keeps the disk(s) containing the generic not currently in use in the off-line state to keep them from being restored. The generic data on an off-line disk is protected from being overwritten by the generic data on its active mate. Upon completion of the update procedure, each pair of duplex disks is the same again. Using this method, the new generic may have a completely different disk image than the old generic.

If a system-initiated boot occurs during the update procedure, the system boots from the generic it was running on previously. New generic disks are marked off-line in the incore Equipment Configuration Database (ECD) as well as the ECD on disk. The new generic disks will still be marked off-line after the boot. If the system tries to boot from the disk that is being updated before the new generic data has been completely entered onto the disk (before the "Proceed" step is completed), the boot will fail. The system will then try to boot from the disk that is not being updated. An example of a successful update using the off-line disk method is as follows:

1. **Enter New Generic:** For each duplex disk pair to be updated, read the new generic data onto the disk that SUPR marks off-line. The data will be entered into the primary and backup partitions.
2. **Proceed With New Generic:** Mark the old generic disk(s) off-line in the new generic ECD. At this point, each generic's ECD considers the disk(s) containing the other generic to be off-line.
3. **Boot From New Generic:** Manually boot the system from the system disk containing the new generic using the EAI.
4. **Commit to New Generic:** Copy the new generic disk(s) into the old generic disk(s) resulting in both disks containing the new generic.

SUPR Description

In 3B20D and 3B21D computers, the SUPR supports system updates by providing a separate process for each step of the update procedure. These steps may be executed in either a conditional or an unconditional mode.

After most of the SUPR processing has completed, each SUPR process transfers control to the application hook. The application hook may then perform any application dependent processing, if desired.

Application dependent processing may also be done using the application process. This process may be invoked by using the argument on the menu page or by using its Program Documentation Standards/Man-Machine Language (PDS/MML) input command.

The SUPR utilities also provide for additional interaction with SUPR (that is, **STOP**, for stopping the process). Refer to the "SUPR Utilities" section.

SUPR Processes

The five SUPR processes used to execute the update procedure are **ENTER**, **PROCEED**, **COMMIT**, **BACKOUT**, and **RESTORE**.

ENTER Request

The SUPR **ENTER** process writes the new generic into off-line disks or backup partitions from tape. At the beginning of the update, both disks (of each duplex disk pair) must be active for the off-line disk method only. The **ENTER** input message specifies update method, source (type unit) pathname, and destination Volume Table of Contents (VTOC) pathname.

The **ENTER** process tells the operator when to mount tapes containing new generic, to acknowledge each request, and to continue the update (using the **CONTINUE** utility). If the **CONTINUE** utility is not executed within 30 minutes, the request is repeated once and allows an additional 30 minutes until the request to mount tapes is repeated. If the **CONTINUE** command is still not executed, the **ENTER** process stops with the error code for operator acknowledgment time-out. The **ENTER** process periodically reports its progress as number of blocks written.

PROCEED Request

This procedure is used after the **ENTER** process is successful. The **PROCEED** process prepares for booting the system under the new generic. For the off-line disk method, this process prompts the operator to boot the system on the “primary root” of the off-line disk. For the backup partition method, this process prompts the operator [this prompt appears on the Receive-Only Printer (ROP) for both the **PROCEED** and **BACKOUT** requests] to boot the system on the file systems that the new generic was loaded (*root* or *broot* on the EAI).

COMMIT Request

This procedure is used when the **PROCEED** process (and boot) is successful. Both disks (of each duplex disk pair) must be active for the backup partition method. The **COMMIT** process copies the new generic over the old (onto the original active disks or partitions). This completes the system update (new generic installed). It is advised that testing be done on the new generic before committing, because it is difficult to restore to the old generic.

BACKOUT Request

This procedure is used when the new generic appears to be faulty. Both disks (of each duplex disk pair) must be active for the backup partition method. The **BACKOUT** process prepares for booting the system under the old generic while running on the new generic. This process prompts the operator to boot the system on the original active disk (for the off-line disk method) or the “primary” or “backup” *root*. The *root* is done on the system that was booted before beginning the SUPR (for the backup partition method). The prompt appears on the ROP for both the **PROCEED** and **BACKOUT** requests.

RESTORE Request

This procedure is used when **BACKOUT** (and boot) is successful or when **ENTER** or **PROCEED** is unsuccessful. Both disks (of each duplex disk pair) must be active for the backup partition method. The **RESTORE** process copies the old generic over the new generic. This completes the restoration of the system to the old generic. The restore must always be used with the Unconditional (UCL) option.

SUPR Utilities

In addition to the five SUPR processes used to implement the steps of the update procedure, there are two SUPR utilities (**CONTINUE** and **STOP**) provided for external interactions with those processes, and one utility (**READLOG**) to examine the contents of the SUPR log.

CONTINUE Utility

The **CONTINUE** utility allows operator interaction with SUPR processes in two ways. First, if an SUPR process requires operator actions to be coordinated with its processing, the process issues the appropriate instructions and then waits for an acknowledgment. Second, if an SUPR process stops, the **CONTINUE** utility is used to restart the process at the point where it stopped. The **CONTINUE** utility determines this from log information.

STOP Utility

The **STOP** utility provides a means to gracefully halt an SUPR process by sending a termination signal to the currently running SUPR process and waiting to see if it terminates.

When the currently running SUPR process receives the termination signal, it performs final cleanup processing and then terminates itself. Upon termination, the **STOP** utility receives a signal indicating that the SUPR process has been terminated, and a termination message is displayed on the ROP.

If the currently running SUPR process does not terminate itself and the **STOP** utility does not receive a termination signal within the 60-second time limit, the **STOP** utility sends the SUPR process a second termination signal. The **STOP** utility then waits again (up to 60 seconds) for the SUPR process to terminate itself.

If the second termination signal fails to stop the currently running SUPR process, the next step depends on how the **STOP** utility is executed. If the **STOP** utility is executed conditionally by omitting the UCL option from the **STOP** command line, the **STOP** utility exits with an error code indicating that it is unable to stop the currently running SUPR process. An error message is displayed on the ROP.

If the **STOP** utility is executed unconditionally by using the UCL option on the **STOP** command line, the **STOP** utility uses the **pkill** command to stop the currently running SUPR process. (A message indicating the SUPR process has been stopped is displayed on the ROP.) In this case, neither the SUPR process nor the *UNIX* system performs any cleanup processing. Thus, any reserved devices (such as disks) may still be reserved.

READLOG Utility

The **READLOG** utility provides formatted information pertaining to the execution of SUPR processes. This information is contained in */etc/log/suprlog*. Each SUPR process makes several entries in the SUPR log file. In case of failure (or halting) of a step, the information in the log may be useful. The **READLOG** utility outputs this information to the ROP.

The **COMMIT** and **RESTORE** processes automatically clear the SUPR log 3 minutes after either completes successfully. If the SUPR log output is desired when such completion occurs, the log file */etc/log/suprlog* must be copied to another location.

Conditional and Unconditional Execution Mode

Each step of the update procedure is executed in either a conditional or an unconditional mode. Under normal circumstances, SUPR execution is conditional. However, an unconditional mode of execution is also available within SUPR to increase the flexibility of the system update.

Conditional Execution Mode

Conditional execution implies that the normal update method is being used and there are no abnormalities associated with the update. Therefore, the conditional execution mode includes some context checks to ensure that the command being executed follows the normal SUPR scenario, and the command is being executed on the proper generic. For *UNIX* RTR Operating System Release 6.4 and later, an additional check is made to verify that the proper tape has been loaded for the specified disk; the *packid* (pack identification) fields of the VTOC on the tape and destination disk must match.

Unconditional Execution Mode

The unconditional execution of any SUPR command adds flexibility to the system update by overriding the context checks for that command. However, all other error checking is still done within the unconditional execution mode.

There are many possibilities for using the unconditional execution mode. The following cases are examples in which a step may be executed unconditionally.

- The SUPR **ENTER** process fails because the wrong tape was mounted for the new generic data input. **ENTER** could be executed again, unconditionally this time, with the proper tape mounted.
- The SUPR **ENTER** process is writing new generic data onto disk. Realizing that certain setup procedures were omitted, the operator returns the system to its original state. In this case, the SUPR **STOP** utility is used to halt the **ENTER** process and the SUPR **RESTORE** command is executed unconditionally to return the system to its original state.
- The tape normally used to update the boot disk pair (*/dev/vtoc*) may be used to update a different disk (for example, */dev/MHD4*) which is also configured as a boot disk. For *UNIX* RTR System Release 6.4 and later, **ENTER** should be executed unconditionally to turn off *packid* field checks which would otherwise cause this update to fail. For prior releases, **ENTER** does not have to be executed unconditionally in this situation.

System Update Menu Page Description

All SUPR commands can be executed manually using operator input messages. They can also be executed using the System Update Menu Page. The System Update Menu Page provides a visual and user-friendly procedure for installing system updates. This procedure helps to eliminate problems caused when typing lengthy input commands. In addition to SUPR processes, an application process may be executed from the menu page. The application process can be executed at the same time that a SUPR process is running.

The System Update Menu Page is composed of three screens that are displayed on a craft terminal to allow the user to execute system update and application commands with their appropriate arguments. The main screen is called the "System Update Menu Page." The other two screens are called "Enter Arguments Page 1" and "Enter Arguments Page 2." All actions are performed on the "System Update Menu Page" screen except specifying the **ENTER** command line arguments which are done via the two Enter Arguments screens.

On each of the three screens, a prompt (CMD<) is located in the upper left-hand corner of the screen under the system status information. A blinking underscore cursor is placed after the prompt to designate where to type the desired menu numbers. On the right-hand side of the CMD< line, the title of each screen appears (System Update Menu Page, Enter Arguments Page 1, Enter Arguments Page 2). For the two Enter Arguments pages, the title also indicates how many pages exist. Below the CMD< line, a "Messages Line" exists to report four types of messages: input errors made by the user, warning messages, instructions concerning system update procedures, and information indicating the status of the update. Error, warning, and procedural messages are displayed in reverse video. Status messages are displayed in normal video. All messages that are displayed on the "Messages Line" are also written to the ROP. The last two lines of each screen are reserved for the scrolling line (used to display output) and the manual input line (used to execute PDS/MML input commands manually), respectively.

Each screen has a set of menu numbers that are typed by the user to request various actions. The user may request only the menu numbers that are displayed on the screen that is currently being viewed. Menu numbers from screens not currently being viewed are considered to be invalid. There is an exception to this rule. If the cursor is positioned on the CMD< line, the user may request a menu number from the Index Menu Page (Page 100) to display a page other than the System Update Menu Page.

System Update Menu Page Screen

Menu numbers 701 through 750 are arranged on the screen according to function. (Refer to Figure 11-2.)

- 701 - 709 Specifies which SUPR or application process is to be executed.
- 720 - 735 Performs an action on one of the SUPR or application process command lines.
- 740 - 750 Performs actions concerning System Update Menu Page screens.

The first three SUPR processes indicated on the screen (**ENTER**, **PROCEED**, **COMMIT**) represent the three SUPR processes that must be executed to install a system update. They are listed in the order in which they must be executed. As each of these three processes complete successfully, their menu numbers and associated process names are displayed in reverse video (**ENTER**) to visually indicate which of the mandatory processes have completed successfully. When all three of these processes are displayed in reverse video, the system update is complete and the System Update Menu Page should be terminated.

- 701 Specifies the **ENTER** command is to be executed.
- 702 Specifies the **PROCEED** command is to be executed.
- 703 Specifies the **COMMIT** command is to be executed.

The SUPR processes represented by menu numbers 704 through 709 are used to interact with the system update process.

- 704 Specifies the **CONTINUE** command is to be executed.
- 705 Specifies the **READLOG** command is to be executed.
- 706 Specifies the **APPL PROC** is to be executed.
- 707 Specifies the **RESTORE** command is to be executed.
- 708 Specifies the **BACKOUT** command is to be executed.
- 709 Specifies the **STOP** command is to be executed.

Menu numbers 720 and 721 represent arguments that may be included on some of the SUPR process command lines. Menu number 735 executes command lines.

- 720 Includes the UCL argument on the specified SUPR command line which specifies that the SUPR process is to be executed unconditionally.

- 721 Includes the RETRO argument on the **PROCEED** command line. This argument is not currently used since **PROCEED** does not handle this option. If this menu number is specified, an error message is displayed on the "Messages Line."
- 735 Executes the previously specified SUPR or Application Process command line that is being displayed in the "Input Command Display Area."

Menu numbers 740 and 741 display the screens used to specify the arguments for the **ENTER** command line. Menu number 750 terminates the System Update Menu Page.

- 740 Displays the Enter Arguments Page 1 screen used to input the following **ENTER** command line arguments: update method, Disk File Controller (DFC) preference, update backup partitions preference, conditional or unconditional execution, and the tape device.
- 741 Displays the Enter Arguments Page 2 screen used to input the following **ENTER** command line arguments: disk(s) to be updated and their corresponding tape sequence options.
- 750 Terminates the System Update Menu Page.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _____				< SYSTEM UPDATE MENU PAGE (107) >			
(Messages Line)							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

(Input Command Display Area)

(Scrolling Line)

< (Manual Input Line)

Figure 11-2. System Update Menu Page Screen

Enter Arguments Page 1 Screen

All **ENTER** command line arguments are specified on the Enter Arguments Page 1 screen, except the disks to be updated and their corresponding tape sequence options, which are specified on the Enter Arguments Page 2 screen. To the right of the menu numbers and their descriptions are the possible values that may be specified for each argument along with their default values (if any). Between the menu descriptions and their possible values, input fields are reserved for the user to specify the desired argument values. When a menu number of an argument is requested, an underline is drawn at the appropriate input field. The length of the underline visually represents the maximum length allowed for that argument value. A blinking underscore cursor is placed at the beginning of the underline to indicate where to input the desired argument value. For the tape device argument, the underline is drawn one line below the description. After an argument is specified, that argument is displayed in reverse video.

Menu numbers 760 through 764 request arguments that are to be specified on the **ENTER** command line. (Refer to Figure 11-3.)

760	Specifies the update method to be used: backup partition method or off-line disk method.
761	Specifies the DFC to be used.
762	Specifies whether or not the backup partitions are to be updated when using the off-line disk update method.
763	Specifies if the update is to be executed conditionally or unconditionally.
764	Specifies the full pathname of the tape device on which the tapes containing the new generic are to be mounted.

Menu numbers 792 and 793 control paging.

792	Displays the Enter Arguments Page 2 screen.
793	Displays the System Update Menu Page screen, automatically formats the ENTER input command line and displays it in the "Input Command Display Area."

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< _____ < ENTER ARGUMENTS PAGE 1 of 2 >
(Messages Line)
760 UPDATE METHOD:                (B = BACKUP PARTITION, 0 = OFF-LINE DISK)
761 DFC PREFERENCE:              (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS:    (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:     (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:                 (SPECIFY FULL PATHNAME)

-----
792 DISPLAY ENTER ARGUMENTS PAGE 2  (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE (VIEW ENTER INPUT COMMAND LINE)
-----
                                   (Scrolling Line)
< (Manual Input Line)
```

Figure 11-3. Enter Arguments Page 1 Screen

Enter Arguments Page 2 Screen

The destinations (DEST), also known as the disks to be updated, and their corresponding tape sequence options (SEQ OPT) are specified on this screen in order to be included as arguments on the **ENTER** command line.

Near the top of the screen, destination and sequence option are defined and their possible argument values are specified. (Refer to Figure 11-4.) The following menu numbers are used to specify the destinations and their corresponding tape sequence options:

- 770 – 779 Specifies the full pathnames of the destinations (the disks where the new generic is to be installed into the system).
- 780 – 789 Specifies the tape sequence options (the number of tape sequences that contain the new generic that is being installed into the system). The following are the possible tape sequence options that can be specified:
 - S = single sequence (default for nonboot disks)
 - D = double sequence (default for boot disks)
 - T = triple sequence.

After one of these arguments has been requested, an underline is drawn immediately after its description. A blinking underscore cursor at the beginning of the line indicates where to input the desired argument value. The length of the line visually represents the maximum length allowed for that argument value. After an argument is specified, the argument is displayed in reverse video.

791	Displays the Enter Arguments Page 1 screen.
793	Displays the System Update Menu Page screen, automatically formats the ENTER input command line and displays it in the "Input Command Display Area."

```

SYS EMER      CRITICAL    MAJOR      MINOR      BLDG/PWR    BLD INH     CKT LIM     SYS NORM
TRAFFIC      SYS INH      CU       CU PERPH   OS LINK
CMD< _____ < ENTER ARGUMENTS PAGE 2 of 2 >
(Message Line)
DEST = DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT = SEQUENCE OPTION FOR DESTINATION (S = SINGLE, D = DOUBLE, T = TRIPLE)
          DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:                                780 SEQ OPT:
771 DEST:                                781 SEQ OPT:
772 DEST:                                782 SEQ OPT:
773 DEST:                                783 SEQ OPT:
774 DEST:                                784 SEQ OPT:
775 DEST:                                785 SEQ OPT:
776 DEST:                                786 SEQ OPT:
777 DEST:                                787 SEQ OPT:
778 DEST:                                788 SEQ OPT:
779 DEST:                                789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1          (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE        (VIEW ENTER INPUT COMMAND LINE)
-----

(Scrolling Line)

< (Manual Input Line)

```

Figure 11-4. Enter Arguments Page 2 Screen

System Update Session Examples

This section gives an example of how to use the system update facility via the System Update Menu Page(s) to install a new generic on the 3B20D and 3B21D computers. Included are examples of:

- A successful system update session
- System update backout
- System update utilities
- Unconditional executions
- Application dependent processing.

Successful System Update Session Example

This example includes many elements that occur during a “live site” system update session. Since this is only an example, no problems or errors will be encountered during this update session.

In this example, the following conventions are used:

- Operator-entered data is shown in **bold** (menu numbers and arguments).
- **video** is shown as white print on a black background.
- All input commands are indicated in PDS format.
- When two menu numbers are entered on the CMD< line, the RETURN key (designated <CR>) must be operated after each menu number.
- The mandatory **ENTER** command line arguments have not been specified; therefore, **ENTER** command line arguments will be specified on the Enter Arguments Page 1 and Enter Arguments Page 2 screens.
- The update method used will be the off-line disk method, and the DFC preference will be DFC1. Where the backup partition method would be different, those changes will be noted.
- The backup partitions will not be updated.
- The pathname of the tape device containing the new generic will be **/dev/mt08**.
- Three disks will be updated: one boot disk and two nonboot disks.
- The tape sequence option for each disk will be single.

Display System Update Menu Page

From the MTTY terminal, our system update session example would be started by using the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key. (The System Update Menu Page cannot be accessed through the EAI Page.)
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.	
3	At the CMD< prompt, enter: 107<CR> to display the System Update Menu Page. The system responds by displaying the System Update Menu Page as shown in Figure 11-5.

SYS EMER TRAFFIC CMD< ____	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLDG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
< SYSTEM UPDATE MENU PAGE (107) >							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			
<							

Figure 11-5. System Update Menu Page

Specify ENTER Command

To specify that the **ENTER** command is the command to be executed, type **701<CR>** on the **CMD<** line. The “Enter Arguments Page 1 of 2” screen will automatically appear since the update method and the tape device arguments have not been previously specified.

⇒ NOTE:

If all the mandatory **ENTER** command line arguments have not been previously specified when 701 is requested, the appropriate Enter Arguments Page will automatically be displayed. If the update method or the tape device were not specified, the Enter Arguments Page 1 screen is displayed. If the boot disk to be updated was not specified or a sequence option was specified without a corresponding disk being specified, the Enter Arguments Page 2 screen is displayed. No DFC preference is required if the backup partition method is used.

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLDG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD< 701<CR>				< SYSTEM UPDATE MENU PAGE (107) >			

701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER	ARGS1
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER	ARGS2
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM	PAGE

Specify Update Method

To specify the update method, type **760**<CR> on the CMD< line. At the blinking underscore cursor, type **O**<CR> for the off-line disk method. This is a mandatory **ENTER** command line argument and therefore must be specified. If the backup partition method is being used, **B** should be entered.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 760<CR>                                     < ENTER ARGUMENTS PAGE 1 of 2 >
```

```
760 UPDATE METHOD:   O                      (B = BACKUP PARTITION, O = OFF-LINE DISK)
761 DFC PREFERENCE:                      (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS:             (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:              (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:                          (SPECIFY FULL PATHNAME)
```

```
-----
792 DISPLAY ENTER ARGUMENTS PAGE 2          (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE         (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Specify DFC Preference

To specify the DFC preference, type **761**<CR> on the CMD< line. At the blinking underscore cursor, type **1**<CR> for DFC 1. If the default value is chosen (no preference), SUPR software will choose DFC 1.

⇒ NOTE:

A DFC preference can only be specified when using the off-line disk update method. If the backup partition update method has been previously specified or if an update method has not yet been specified, and a DFC preference is attempted to be specified, an error message will appear on the "Messages Line." Make sure that the off-line disk update method is specified before a DFC preference is requested. No DFC preference is required if the backup partition method is used.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 761<CR>                                     < ENTER ARGUMENTS PAGE 1 of 2 >

760 UPDATE METHOD:  0                      (B = BACKUP PARTITION, 0 = OFF-LINE DISK)
761 DFC PREFERENCE: 1                      (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS:              (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:              (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:                            (SPECIFY FULL PATHNAME)

-----
792 DISPLAY ENTER ARGUMENTS PAGE 2          (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE         (VIEW ENTER INPUT COMMAND LINE)
-----
<
```

Specify Update Backup Partitions Option

To specify the update backup partitions option, type **762**<CR> on the CMD< line. At the blinking underscore cursor, type **N**<CR> for “NO.” The backup partitions will not be updated.

⇒ NOTE:

The option of updating the backup partitions can only be specified when using the off-line disk update method. If the backup partition update method has been previously specified or if an update method has not yet been specified and the update backup partition option is attempted to be specified, an error message will appear on the “Messages Line.” Make sure that the off-line disk update method is specified before the backup partition option is requested.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 762<CR>                                     < ENTER ARGUMENTS PAGE 1 of 2 >

760 UPDATE METHOD:  0          (B = BACKUP PARTITION, 0 = OFF-LINE DISK)
761 DFC PREFERENCE: 1          (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS: N (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:      (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:      (SPECIFY FULL PATHNAME)

-----
792 DISPLAY ENTER ARGUMENTS PAGE 2      (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE     (VIEW ENTER INPUT COMMAND LINE)
-----

<
```

Specify Conditional/Unconditional Mode

To specify the executional mode, type **763**<CR> on the CMD< line. At the blinking underscore cursor, type **N**<CR> for “NO.” This is the preferred method for a normal update.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 763<CR>                                     < ENTER ARGUMENTS PAGE 1 of 2 >
```

```
760 UPDATE METHOD:  0          (B = BACKUP PARTITION, 0 = OFF-LINE DISK)
761 DFC PREFERENCE: 1          (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS: N (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:  N (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:          (SPECIFY FULL PATHNAME)
```

```
-----
792 DISPLAY ENTER ARGUMENTS PAGE 2          (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE        (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Specify Tape Device

To specify the tape device, type **764**<CR> on the CMD< line. At the blinking underscore cursor (beneath the tape device option), type **/dev/mt08**<CR>. This is the full pathname of the tape device on which the tape containing the new generic is to be mounted. This is a mandatory **ENTER** command line argument and therefore must be specified.

All the **ENTER** command line arguments from the Enter Arguments Page 1 screen have now been specified.

⇒ NOTE:

When specifying the tape device, an error message will appear on the "Messages Line" if the full pathname of the tape device is not specified, a nonexistent tape device is specified, or a tape device is specified that does not have read permissions. In the *UNIX RTR* Operating System Release 6 and later, the option of specifying "mt0," "MT0," "mt1," or "MT1" may be used to specify the following tape devices only:

mt0 or MT0	/dev/mt00 or /dev/mt08
mt1 or MT1	/dev/mt10 or /dev/mt18

This option allows the user to specify only the tape unit number without knowing the tape density. SUPR will support different tape densities even if a full tape name is specified. If a full tape name is specified with the wrong density, the other density is used and all other characteristics if the original tape name are maintained (for example, no rewind).

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 764<CR>				< ENTER ARGUMENTS PAGE 1 of 2 >			
760 UPDATE METHOD: 0				(B = BACKUP PARTITION, 0 = OFF-LINE DISK)			
761 DFC PREFERENCE: 1				(0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)			
762 UPDATE BACKUP PARTITIONS: N				(Y = YES <DEFAULT>, N = NO)			
763 UNCONDITIONAL EXECUTION: N				(N = NO <DEFAULT>, Y = YES)			
764 TAPE DEVICE:				(SPECIFY FULL PATHNAME)			
/dev/mt08							
792 DISPLAY ENTER ARGUMENTS PAGE 2				(SPECIFY DISKS AND SEQUENCE OPTIONS)			
793 DISPLAY SYSTEM UPDATE MENU PAGE				(VIEW ENTER INPUT COMMAND LINE)			

<							

Display Enter Arguments Page 2 Screen

To display the Enter Arguments Page 2 screen, type **792**<CR> on the CMD< line. The disks to be updated along with their sequence options may now be specified.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 792<CR>                                     < ENTER ARGUMENTS PAGE 1 of 2 >

760 UPDATE METHOD:   0                      (B = BACKUP PARTITION, 0 = OFF-LINE DISK)
761 DFC PREFERENCE:  1                      (0 = DFC0, 1 = DFC1, 2 = NO PREF <DEFAULT>)
762 UPDATE BACKUP PARTITIONS:  N            (Y = YES <DEFAULT>, N = NO)
763 UNCONDITIONAL EXECUTION:  N            (N = NO <DEFAULT>, Y = YES)
764 TAPE DEVICE:
      /dev/mt08                             (SPECIFY FULL PATHNAME)

-----
792 DISPLAY ENTER ARGUMENTS PAGE 2          (SPECIFY DISKS AND SEQUENCE OPTIONS)
793 DISPLAY SYSTEM UPDATE MENU PAGE        (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Specify First Disk to Be Updated

Specify the first disk to be updated by typing **770<CR>** on the **CMD<** line. At the blinking underscore cursor, specify the pathname of the boot disk by typing **/dev/vtoc<CR>**.

⇒ NOTE:

When specifying the disk to be updated, an error message will appear on the “Messages Line” if the full pathname is not specified, a nonexistent disk is specified, or a disk is specified that does not have read and write permissions. Multiple disks may be specified in any order and on any DEST line of the Enter Arguments Page 2 screen (disk 775 may be specified before disk 770). It is mandatory that at least one disk that is specified must be a boot disk (**/dev/vtoc**). The maximum number of disks allowed on the **ENTER** command line is ten which allows for up to nine nonboot disks to be specified.

```
SYS EMER   CRITICAL   MAJOR    MINOR    BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH     CU       CU PERPH  OS LINK
CMD< 770<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =    SEQUENCE OPTION FOR DESTINATION  (S = SINGLE, D = DOUBLE, T = TRIPLE)
              DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:    /dev/vtoc                               780 SEQ OPT:
771 DEST:                                         781 SEQ OPT:
772 DEST:                                         782 SEQ OPT:
773 DEST:                                         783 SEQ OPT:
774 DEST:                                         784 SEQ OPT:
775 DEST:                                         785 SEQ OPT:
776 DEST:                                         786 SEQ OPT:
777 DEST:                                         787 SEQ OPT:
778 DEST:                                         788 SEQ OPT:
779 DEST:                                         789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
```

<

Specify First Disk Sequence Option

To specify the sequence option for disk 770, type **780<CR>** on the **CMD<** line.
At the blinking underscore cursor, specify “single” tape sequence option by typing **S<CR>**.

⇒ NOTE:

The tape sequence option specified must correspond to the DEST that was previously specified on that same line before the next DEST is specified. An error message appears on the “Messages Line” if a sequence option is specified without previously specifying its corresponding disk. Single, double, and triple are valid sequence options for both boot disks and nonboot disks. The default values for a boot disk and a nonboot disk are double and single, respectively.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH     CU      CU PERPH  OS LINK
CMD< 780<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT = SEQUENCE OPTION FOR DESTINATION  (S = SINGLE, D = DOUBLE, T = TRIPLE)
          DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:  /dev/vtoc                               780 SEQ OPT:  S
771 DEST:                                     781 SEQ OPT:
772 DEST:                                     782 SEQ OPT:
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:                                     785 SEQ OPT:
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----

<
```

Specify Second Disk to Be Updated

Specify the second disk to be updated by typing **775<CR>** on the **CMD<** line. At the blinking underscore cursor, specify the pathname of the nonboot disk by typing **/dev/mhd4<CR>**.

⇒ NOTE:

Multiple disks may be specified in any order and on any DEST line of the Enter Arguments Page 2 screen (disk 775 may be specified before disk 770).

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINK
CMD< 775<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =   SEQUENCE OPTION FOR DESTINATION (S = SINGLE, D = DOUBLE, T = TRIPLE)
           DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:   /dev/vtoc                                     780 SEQ OPT: S
771 DEST:                                     781 SEQ OPT:
772 DEST:                                     782 SEQ OPT:
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:   /dev/mhd4                                    785 SEQ OPT:
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Specify Second Disk Sequence Option

To specify the sequence option for disk 775, type **785<CR>** on the **CMD<** line.
At the blinking underscore cursor, specify "single" tape sequence option by typing **S<CR>**.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH OS LINK
CMD< 785<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =   SEQUENCE OPTION FOR DESTINATION (S = SINGLE, D = DOUBLE, T = TRIPLE)
            DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:   /dev/vtoc                               780 SEQ OPT: S
771 DEST:                                     781 SEQ OPT:
772 DEST:                                     782 SEQ OPT:
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:   /dev/mhd4                               785 SEQ OPT: S
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----

<
```

Specify Third Disk to Be Updated

Specify the third disk to be updated by typing **772<CR>** on the **CMD<** line. At the blinking underscore cursor, specify the pathname of the nonboot disk by typing **/dev/mhd2<CR>**.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR  BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINK
CMD< 772<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =   SEQUENCE OPTION FOR DESTINATION  (S = SINGLE, D = DOUBLE, T = TRIPLE)
           DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:  /dev/vtoc                               780 SEQ OPT:  S
771 DEST:                                     781 SEQ OPT:
772 DEST:  /dev/mhd2                               782 SEQ OPT:
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:  /dev/mhd4                               785 SEQ OPT:  S
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Specify Third Disk Sequence Option

To specify the sequence option for disk 772, type **782**<CR> on the CMD< line.
At the blinking underscore cursor, specify "single" tape sequence option by
typing **S**<CR>.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINK
CMD< 782<CR>                                     < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT =   SEQUENCE OPTION FOR DESTINATION  (S = SINGLE, D = DOUBLE, T = TRIPLE)
           DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:  /dev/vtoc                               780 SEQ OPT:  S
771 DEST:                                     781 SEQ OPT:
772 DEST:  /dev/mhd2                               782 SEQ OPT:  S
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:  /dev/mhd4                               785 SEQ OPT:  S
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----
```

<

Return to System Update Menu Page

Now that all the mandatory **ENTER** command line arguments have been specified, it is necessary to display the System Update Menu Page so that the **ENTER** command may be executed. To display the System Update Menu Page, type **793<CR>** on the CMD< line. The System Update Menu Page will be displayed. The **ENTER** command line will be automatically formatted in the appropriate PDS/MML format and displayed in the display area of the System Update Menu Page.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINK
CMD< 793<CR>                                < ENTER ARGUMENTS PAGE 2 of 2 >

DEST =      DESTINATION - DISK TO BE UPDATED (FULL PATHNAME)
SEQ OPT = SEQUENCE OPTION FOR DESTINATION (S = SINGLE, D = DOUBLE, T = TRIPLE)
          DEFAULTS: BOOT DISK = D, NON-BOOT DISK = S

770 DEST:  /dev/vtoc                               780 SEQ OPT:  S
771 DEST:                                     781 SEQ OPT:
772 DEST:  /dev/mhd2                               782 SEQ OPT:  S
773 DEST:                                     783 SEQ OPT:
774 DEST:                                     784 SEQ OPT:
775 DEST:  /dev/mhd4                               785 SEQ OPT:  S
776 DEST:                                     786 SEQ OPT:
777 DEST:                                     787 SEQ OPT:
778 DEST:                                     788 SEQ OPT:
779 DEST:                                     789 SEQ OPT:

-----
791 DISPLAY ENTER ARGUMENTS PAGE 1                (SPECIFY OTHER ENTER ARGUMENTS)
793 DISPLAY SYSTEM UPDATE MENU PAGE                (VIEW ENTER INPUT COMMAND LINE)
-----

<
```

Execute ENTER Command

To execute the **ENTER** command, type **735<CR>** on the **CMD<** line. The message **THE ENTER PROCESS IS EXECUTING** is displayed on the "Messages Line" to indicate that the **ENTER** process is executing.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 735<CR>                                     < SYSTEM UPDATE MENU PAGE (107) >
THE ENTER PROCESS IS EXECUTING

701 ENTER   704 CONTINUE   707 RESTORE   720 UCL       740 ENTER ARGS1
702 PROCEED 705 READLOG    708 BACKOUT   721 RETRO     741 ENTER ARGS2
703 COMMIT  706 APPL PROC  709 STOP PROC 735 EXECUTE   750 TERM PAGE
```

```
UPD:GEN;ENTER:OFLDISK 1,NOBKPR,T,UCL
SRC "/dev/mt08",
DEST ("/dev/vtoc",
      "/dev/mhd2",
      "/dev/mhd4"),
SEQOPT (SGL, SGL, SGL)!
```

<

Mount Tape for Disk 770

The system update halts the **ENTER** step of the update when it is time to mount a tape. The message MOUNT THE TAPE FOR THE APPROPRIATE DISK AND CONTINUE THE UPDATE is displayed on the "Messages Line" when it is time to mount the tape containing the new generic for the first disk to be updated. The tape that is mounted must correspond to the disk currently being displayed in reverse video. Mount the tape for disk 770 (pathname `/dev/vtoc`) and continue the update.

⇒ NOTE:

The continue operation will be repeated for each tape in the sequence.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _____				< SYSTEM UPDATE MENU PAGE (107) >			
UPDATE							
701 ENTER	704 CONTINUE	707 RESTORE		720 UCL	740 ENTER ARGS1		
702 PROCEED	705 READLOG	708 BACKOUT		721 RETRO	741 ENTER ARGS2		
703 COMMIT	706 APPL PROC	709 STOP PROC		735 EXECUTE	750 TERM PAGE		


```
UPD:GEN;ENTER:0FLDISK 1,NOBKPRT,
SRC "/dev/mt08",
  ("/dev/vtoc",
    "/dev/mhd2",
    "/dev/mhd4"),
SEQOPT (SGL, SGL, SGL)!
```


<

Continue Update for Disk 770

Once the tape has been mounted, the process must be continued. Continue the **ENTER** step by typing **704<CR>** on the **CMD<** line. Note the input command message displayed in the display area of the System Update Menu Page.

To execute the **CONTINUE** command, type **735<CR>** on the **CMD<** line. Note the message on the "Messages Line" indicating that the **CONTINUE** process is executing. A completion message will be displayed for only a few seconds when the **CONTINUE** process completes successfully. After this message is displayed, the **ENTER** input command will be automatically redisplayed in the display area and the message **THE CONTINUE PROCESS IS EXECUTING** will be redisplayed on the "Messages Line."

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 704<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE CONTINUE PROCESS IS EXECUTING							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			
UPD:GEN;CONTINUE!							
<							

Mount Tape for Disk 772

After the **ENTER** process finishes updating the first disk, the message **MOUNT THE TAPE FOR THE APPROPRIATE DISK AND CONTINUE THE UPDATE** will reappear so that the second disk may be updated. The pathname of this disk appears in reverse video.

Mount the tape for disk 772 (pathname `/dev/mhd2`) and continue the update.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _____				< SYSTEM UPDATE MENU PAGE (107) >			
UPDATE							
701 ENTER	704 CONTINUE	707 RESTORE		720 UCL	740 ENTER ARGS1		
702 PROCEED	705 READLOG	708 BACKOUT		721 RETRO	741 ENTER ARGS2		
703 COMMIT	706 APPL PROC	709 STOP PROC		735 EXECUTE	750 TERM PAGE		


```
UPD:GEN;ENTER:OFLDISK 1,NOBKPR,  
SRC "/dev/mt08",  
DEST ("/dev/vtoc",  
      "/dev/mhd2",  
      "/dev/mhd4"),  
SEQOPT (SGL, SGL, SGL)!
```

<

Continue Update for Disk 772

Continue the **ENTER** step by typing **704<CR>** and **735<CR>** on the **CMD<** line. Note the messages displayed on the “Messages Line” and in the display area of the menu page.

A completion message will be displayed for only a few seconds when the **CONTINUE** process completes successfully. After this message is displayed, the **ENTER** input command will be automatically redisplayed in the display area and the **THE CONTINUE PROCESS IS EXECUTING** message will be redisplayed on the “Messages Line.”

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 704<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE CONTINUE PROCESS IS EXECUTING							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

UPD:GEN;CONTINUE!

<

Mount Tape for Disk 775

After the **ENTER** process finishes updating the second disk, the message MOUNT THE TAPE FOR THE APPROPRIATE DISK AND CONTINUE THE UPDATE will reappear so that the third and final disk may be updated. The pathname of this disk appears in reverse video.

Mount the tape for disk 775 (pathname `/dev/mhd4`) and continue the update.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _____				< SYSTEM UPDATE MENU PAGE (107) >			
UPDATE							
701 ENTER	704 CONTINUE	707 RESTORE		720 UCL	740 ENTER ARGS1		
702 PROCEED	705 READLOG	708 BACKOUT		721 RETRO	741 ENTER ARGS2		
703 COMMIT	706 APPL PROC	709 STOP PROC		735 EXECUTE	750 TERM PAGE		


```
UPD:GEN;ENTER:OFLDISK 1,NOBKPR,  
SRC "/dev/mt08",  
DEST ("/dev/vtoc",  
      "/dev/mhd2",  
      "/dev/mhd4"),  
SEQOPT (SGL, SGL, SGL)!
```

<

Continue Update for Disk 775

Continue the **ENTER** step by typing **704<CR>** and **735<CR>** on the **CMD<** line.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 704<CR>      735<CR>          < SYSTEM UPDATE MENU PAGE (107) >
THE CONTINUE PROCESS IS EXECUTING
```

701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE

UPD:GEN;CONTINUE!

<

***ENTER* Process Terminates**

After the third disk has been successfully updated, a completion message is displayed on the "Messages Line" to indicate that the **ENTER** process terminated successfully. Also, the menu number and description (**ENTER**) is displayed in reverse video to show that the process completed successfully.

```

SYN EMER    CRITICAL    MAJOR    MINOR    BLDG/PWR    BLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU       CU PERPH  OS LINKS
CMD< _____ < SYSTEM UPDATE MENU PAGE (107) >
  THE ENTER PROCESS COMPLETED SUCCESSFULLY

```

ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE

```

<

```

Execute PROCEED Command

To execute the **PROCEED** command, type **702<CR>** and **735<CR>** on the CMD< line. Note the messages displayed on the “Messages Line” and in the display area.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 702<CR>    735<CR>                < SYSTEM UPDATE MENU PAGE (107) >
THE PROCEED PROCESS IS EXECUTING

ENTER      704 CONTINUE      707 RESTORE      720 UCL      740 ENTER ARGS1
702 PROCEED 705 READLOG      708 BACKOUT     721 RETRO    741 ENTER ARGS2
703 COMMIT  706 APPL PROC    709 STOP PROC   735 EXECUTE  750 TERM PAGE

UPD:GEN;PROCEED!

-----
<
```

PROCEED Process Terminates

A completion message is displayed on the "Messages Line" when the **PROCEED** process terminates successfully. Also, the menu number and description (**PROCEED**) is displayed in reverse video to show that the process completed successfully.

```
SYS EMER   CRITICAL   MAJOR   MINOR   BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< _____ < SYSTEM UPDATE MENU PAGE (107) >
THE PROCEED PROCESS COMPLETED SUCCESSFULLY

  ENTER    704 CONTINUE    707 RESTORE    720 UCL    740 ENTER ARGS1
  PROCEED  705 READLOG    708 BACKOUT   721 RETRO  741 ENTER ARGS2
 703 COMMIT 706 APPL PROC   709 STOP PROC 735 EXECUTE 750 TERM PAGE

-----
<
```

Boot to New Generic

After the **ENTER** and **PROCEED** commands have been successfully executed, the operator must manually boot the system to the new generic. Boot the system using the following steps:

1. Terminate the System Update Menu Page before booting by typing **750<CR>** on the **CMD<** line. If not terminated, the boot will automatically terminate the menu page anyway.
2. Boot to the new generic. The system **MUST** boot off the Moving Head Disk (MHD) that contains the new generic. This information is written by the **PROCEED** step and is found on the ROP.

Execute COMMIT Command

When the system is running on the new generic, bring up the System Update Menu Page by typing **107**<CR> on the CMD< line. The menu page will remember the last step performed in the update session and update the visuals on the System Update Menu Page.

⇒ NOTE:

Normally the **COMMIT** is not performed until after it has been determined that the new generic is running satisfactorily.

To execute the **COMMIT** command, type **703**<CR> and **735**<CR> on the CMD< line. Note the messages displayed on the “Messages Line” and in the display area.

When using the backup partition method, the Out of Service (OOS) disk of a duplex pair (resulting from the boot) should be restored to the active state before executing the **COMMIT** process.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 703<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE COMMIT PROCESS IS EXECUTING							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

UPD:GEN;COMMIT!

<

COMMIT Process Terminates

A completion message is displayed on the "Messages Line" when the **COMMIT** process terminates successfully. Also, the menu number and description (**COMMIT**) are displayed in reverse video to show that the process completed successfully. The system update has been completed once the three mandatory processes are displayed in reverse video.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _____				< SYSTEM UPDATE MENU PAGE (107) >			
THE COMMIT PROCESS COMPLETED SUCCESSFULLY							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

<

Terminate System Update Menu Page

The System Update Menu Page needs to be terminated after each update session. If the menu page is not terminated, the process that runs the menu page will execute indefinitely until it is terminated. The menu page can be terminated at any time as long as a system update process or the application process is not currently executing. To terminate the System Update Menu Page, type **750**<CR> on the CMD< line.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 750<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE COMMIT PROCESS COMPLETED SUCCESSFULLY							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

<

After the System Update Menu Page interface process sends a message to the Display Administration Process (DAP) to terminate the System Update Menu Page, the interface process writes a completion message to the ROP and to the terminal to inform the operator that the menu page terminated successfully.

System Update **BACKOUT** Example

If the new generic appears to be faulty or some other problem occurs during the update session, the update must be backed out and the old generic restored.

Execute **BACKOUT** Command

Execute the **BACKOUT** command by typing **708<CR>** and **735<CR>** on the CMD< line. Note the messages displayed on the “Messages Line” and in the display area.

When using the backout partition method before executing the backout process, the OOS disk of a duplex pair (resulting from the boot) should be restored to the active state.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 708<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE BACKOUT PROCESS IS EXECUTING							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

UPD: GEN; BACKOUT !

<

When the **BACKOUT** process terminates, a completion message or an error message is displayed on the “Messages Line” depending on whether the **BACKOUT** process completed successfully or unsuccessfully.

Boot to Old Generic

After the **BACKOUT** command has been successfully executed, the operator must manually boot the system to the old generic. Boot the system using the following steps:

1. Terminate the System Update Menu Page before booting by typing **750<CR>** on the **CMD<** line. If not terminated, the boot will automatically terminate the menu page anyway.
2. Boot to the old generic. Boot from the MHD that contains the old generic. This information is written by the **BACKOUT** procedure and is found on the ROP.

Execute the RESTORE Command

When the system is running on the old generic, bring up the System Update Menu Page by typing **107<CR>** on the **CMD<** line. The menu page will remember the last step performed in the update session and update the visuals on the System Update Menu Page.

NOTE:

If the **ENTER** and **PROCEED** commands were previously executed and completed successfully, their menu numbers and descriptions are displayed in reverse video. When the **RESTORE** process completes successfully, the menu numbers and descriptions for the **ENTER** and **PROCEED** commands are returned to normal video. These commands are no longer in the completed state. The update must be started from the beginning of the **ENTER** command.

RESTORE will restore the old generic by removing the new generic from the system. To prevent a misentered command from restoring the old generic, **RESTORE** must always use the UCL option.

Execute the **RESTORE** command by typing **707<CR>**, **720<CR>**, and **735<CR>** on the CMD< line. Note the messages on the “Messages Line” and in the display area.

When using the backup partition method (before executing the restore process), the OOS disk of a duplex pair (resulting from the boot) should be restored to the active state.

```
SYS EMER   CRITICAL   MAJOR    MINOR    BLDG/PWR  BLD INH  CKT LIM  SYS NORM
TRAFFIC    SYS INH    CU      CU PERPH  OS LINKS
CMD< 707<CR> 720<CR> 735<CR>          < SYSTEM UPDATE MENU PAGE (107) >
THE RESTORE PROCESS IS EXECUTING

  ENTER  704 CONTINUE      707 RESTORE      720 UCL      740 ENTER ARGS1
  PROCEED 705 READLOG      708 BACKOUT     721 RETRO     741 ENTER ARGS2
703 COMMIT 706 APPL PROC   709 STOP PROC   735 EXECUTE   750 TERM PAGE
-----
UPD:GEN;RESTORE;UCL!

<
```

When the **RESTORE** process terminates, a completion message or an error message is displayed on the “Messages Line” depending on whether the **RESTORE** process completed successfully or unsuccessfully.

System Update Utilities Examples

Three SUPR utilities interact with the five SUPR processes used to implement the steps of the update procedure. The **CONTINUE** and **STOP** utilities provide for external interactions with those processes, and the **READLOG** utility is used to examine the contents of the SUPR log. The **CONTINUE** utility is discussed in the "Successful Update Session Example."

Execute STOP Command

If a system update command (**ENTER**, **PROCEED**, **COMMIT**, **RESTORE**, or **BACKOUT**) is currently executing and needs to be stopped, type **709<CR>** and **735<CR>** on the **CMD<** line. The PDS/MML input command line for the **STOP** command will automatically be formatted and displayed in the display area of the System Update Menu Page. This will overwrite the displayed input command of the currently running system update process. On the "Messages Line," a message will appear in normal video indicating that the **STOP** process is executing (overwriting the execution message of the currently running process).

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 709<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE STOP PROCESS IS EXECUTING							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

STOP:GEN!

<

When the **STOP** process terminates, the currently running system update command, an error message will be displayed in reverse video on the "Messages Line" indicating that the system update process terminated unsuccessfully. This means that the **STOP** command successfully terminated the system update process.

If the **STOP** process is unable to stop the currently running system update process, an error message is displayed on the "Messages Line" stating that the **STOP** command failed and the currently running process is still executing. The PDS/MML input command line for that system update process will automatically be redisplayed to show the user the command line that is still being executed.

Execute READLOG Command

The **READLOG** command can be executed at any time during a system update by typing **705<CR>** and **735<CR>** on the **CMD<** line. If the **READLOG** command is executed while a system update process is currently executing, the execution message on the "Messages Line" for the **READLOG** process will temporarily overwrite the execution message for the currently running system update process. Also, the PDS/MML input command for the **READLOG** process that is displayed will temporarily overwrite the PDS/MML input command for the currently running system update process.

When the **READLOG** process terminates, a completion message or an error message is displayed on the "Messages Line" depending on whether the **READLOG** process completed successfully or unsuccessfully. The executing message for the currently running system update process is then automatically redisplayed on the "Messages Line" to inform the operator that this process is still executing. The PDS/MML input command for the system update process is also automatically formatted and redisplayed in the display area.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 705<CR>				735<CR>		< SYSTEM UPDATE MENU PAGE (107) >	
THE READLOG PROCESS IS EXECUTING							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

OP: GEN; READLOG!

<

Unconditional Execution Example

The **ENTER**, **PROCEED**, **COMMIT**, **RESTORE**, **BACKOUT**, and **STOP** commands can be executed unconditionally (ignoring some error checking). These are the only system update commands that accept the UCL argument. If the UCL argument is attempted to be specified for any other input command, an error message will be displayed on the "Messages Line." The UCL argument can be specified along with the system update command or after the system update command argument has already been specified.

Specify UCL Argument with Command Line Argument

To specify the UCL argument for the **PROCEED** command, type **702,720<CR>** on the CMD< line. The UCL argument can also be specified by typing **702,UCL<CR>** on the CMD< line. (The “UCL” may be entered in either uppercase or lowercase letters.) The PDS/MML input command line for the desired command will automatically be formatted with the UCL argument and displayed in the display area of the System Update Menu Page.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 702,720<CR>				< SYSTEM UPDATE MENU PAGE (107) >			

ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE

UPD:GEN;PROCEED;UCL!

<

Add UCL Argument to Specified Command Line Argument

If a command has already been specified without the UCL argument, it can be added prior to executing the command by typing **720<CR>** on the **CMD<** line. The UCL argument will automatically be added to the PDS/MML input command that is currently being displayed on the System Update Menu Page.

SYS EMER TRAFFIC CMD< 720<CR>	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLDG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
< SYSTEM UPDATE MENU PAGE (107) >							
ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			
UPD: GEN; PROCEED; UCL !							
<							

Remove UCL Argument from Specified Command Line Argument

If the UCL argument has been specified on the input command line but is no longer desired, it can be "removed" by typing the menu number of the desired command on the **CMD<** line without the UCL argument (**702<CR>**). The PDS/MML input command line will automatically be reformatted and redisplayed without the UCL argument.

Application Dependent Processing Example

The application process can be executed at any time during a system update, even when a system update process is currently executing. To specify that an application process is to be executed, type **706**<CR> on the CMD< line. The PDS/MML input command line for the application process will automatically be formatted and displayed on the System Update Menu Page. To specify and execute an argument to the application process, type **706,myarg**<CR> and **735**<CR> on the CMD< line. Note the message displayed on the "Messages Line."

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 706,myarg<CR> 735<CR>				< SYSTEM UPDATE MENU PAGE (107) >			
THE APPLICATION PROCESS IS EXECUTING							
701 ENTER	704 CONTINUE	707 RESTORE	720 UCL	740 ENTER ARGS1			
702 PROCEED	705 READLOG	708 BACKOUT	721 RETRO	741 ENTER ARGS2			
703 COMMIT	706 APPL PROC	709 STOP PROC	735 EXECUTE	750 TERM PAGE			

UPD:GEN;APPLPROC:ARG "myarg"!

<

When the application process terminates, a completion message or an error message is displayed on the "Messages Line" depending on whether the application process completed successfully or unsuccessfully. Also, three bell tones are sounded to further alert the operator that the application process has completed.

The argument "*myarg*" represents an argument string that the application process will accept. Any character can be specified in the argument string except a comma (,) which is invalid. If an argument is specified for the application process but is not really desired, it can be "erased" by typing **706**<CR> (without the argument). The PDS/MML input command line for the application process will be redisplayed without an argument, just as if one were never specified.

If a system update command is executing at the same time the application process is executed, the execution message for the application process temporarily overwrites the execution message for the system update command. Also, the PDS/MML input command for the application process temporarily overwrites the PDS/MML input command for the currently executing system update process that was displayed. After a few seconds, the execution message for the application process disappears and the execution message for the system update process that is currently running is redisplayed on the "Messages Line". The PDS/MML input command for the currently running system update process is also redisplayed to indicate to the user which system update process is executing.

If a system update command is executing when the application process completes, the execution message on the "Messages Line" for that system update command is temporarily overwritten by the completion message for the application process. After a few seconds, the completion message for the application process disappears and the execution message for the currently running system update process is automatically redisplayed on the "Messages Line." The message indicates to the user that a system update command is still executing.

RETRO Argument

Prior to *UNIX* RTR Operating System Release 1, the RETRO option (Poke **721**) was used with the **PROCEED** command to indicate that the new generic ECD is incompatible with the old generic "Recent Change" utility. The option indicated that the **PROCEED** command should use a specially provided version of "Recent Change" to update the new generic ECD. This option is no longer used and should not be specified.

For *UNIX* RTR Operating System Release 1 and later, the **PROCEED** command uses a specially provided "Recent Change" program, if one has been installed in */usr/bin/retrcvecd*, before beginning the retrofit. This would normally be installed, if needed, as part of the retrofit Broadcast Warning Message (BWM) package. Otherwise, the default "Recent Change" program (*/usr/bin/rcvecd*) is used.

Troubleshooting System Update Procedures

Unless otherwise noted, all problems in this section pertain to the two system update methods used by SUPR as follows:

- Off-line disk
- Backup partition.

If an error occurs during SUPR processing, a message is displayed on the ROP along with a six-digit error code. Refer to 254-303-111, *PDS Output Messages Manual*, and 254-303-113, *MML Output Messages Manual*, for listings of each error code along with possible causes and recovery action which should be the first resource examined if an error occurs. The following are recovery procedures for other possible problem situations:

Problem A

Problem A is specifically for the backup partition method.

Problem	Updating a disk(s) using the backup partition method that has a different disk image than that of the tape you are using in the system update.
Solution	After the ENTER process fails, restore unconditionally and start the process over using the correct tape/disk.

⇒ NOTE:

The tapes used in the SUPR process were made with the same disk image as MHD 0 and MHD 1. MHD 3 has a different disk image than that of the tape.

On the Enter Arguments Page 1 screen, the backup partition method and other options are specified.

On the Enter Arguments Page 2 screen, */dev/vtoc* (MHD 0 and MHD 1) and */dev/mhd3* are specified as the disks to be updated. The update will be done using a single tape sequence for both destinations.

ENTER and **CONTINUE** are both successful when updating MHD 0 and MHD 1, but the **CONTINUE** process fails for MHD 3. Executing the **ENTER** command unconditionally will not continue this update. At this point, you need to restore unconditionally. This will restore the old generic on MHD 0 and MHD 1 to which the new generic has been loaded.

Problem B

Problem	Updating a disk(s) and the ENTER process fails because of a discrepancy in the <i>packid</i> fields.
Solution	Reexecute the ENTER command unconditionally.

⇒ NOTE:

The tapes used in this SUPR process were made from MHD 0 and MHD 1 and were used to update MHD 0, MHD 1, and MHD 2.

On the Enter Arguments Page 1 screen, the off-line method and other options are specified.

On the Enter Arguments Page 2 screen, */dev/vtoc* and */dev/mhd2* are specified as the disks to be updated. The update will be done using a double tape sequence.

ENTER and **CONTINUE** are both successful for MHD 0 and MHD 1. The **CONTINUE** process fails for MHD 2. At this point, you should reexecute the **ENTER** command unconditionally to continue the update. To do this, Poke **720**, which will append the UCL option onto the **ENTER** command, then continue the process.

Problem C

Problem	Updating a disk using the wrong tape device on the Enter Arguments Page 1 screen.
Solution	Restore unconditionally and start the SUPR process over with the correct tape device specified.

⇒ NOTE:

In this example, the tapes used in the SUPR process were medium density tapes, but the tape device originally specified is used for high density tapes.

On the Enter Arguments Page 1 screen, the method and the tape device (*/dev/mt00*) are specified.

On the Enter Arguments Page 2 screen, */dev/vtoc* is specified as the disk pair to be updated. The update will be done using a single tape sequence.

The **ENTER** process is successful, but the **CONTINUE** process fails. At this point, changing the tape device on the Enter Arguments Page 1 screen to a medium density tape device and reexecuting the **ENTER** process unconditionally will fail. You now need to restore unconditionally and start the SUPR process over with the correct tape device specified.

Problem D

Problem	A SUPR log file already exists.
Solution	Scenario 1: A system update is in progress. Execute the RESTORE process unconditionally to ensure you are getting a clean copy of the existing system before doing your system update. Scenario 2: A system update is in progress, but the RESTORE process fails which indicates the SUPR log file may be corrupted. Follow the instructions under the solution for Problem E to terminate the SUPR Menu Page (also known as the System Update Menu Page), to remove the log file <i>/etc/log/suprlog</i>, and to start your system update. Scenario 3: A system update is not in progress. Terminate the SUPR Menu Page and remove the log file to start your system update.

NOTE:

The SUPR Menu Page is selected to start a system update and the **ENTER** and **PROCEED** processes are highlighted. MHD 1 is still off-line.

Execute the **RESTORE** process unconditionally. Upon completion, terminate the SUPR Menu Page and begin your system update.

The SUPR Menu Page is selected to start a system update and the **ENTER** process is highlighted. MHD 1 and MHD 0 are both active.

Terminate the SUPR Menu Page and remove the log file to start your system update.

Problem E

Problem E is specific to the off-line disk method.

Problem	SUPR stops unexpectedly with an uncontinuable error during the ENTER process leaving the disk onto which the new generic will be written as OFL (off-line).
Solution	Before the ENTER process can be executed again, the OFL disk must be in the active state. To do this, mark the disk as OOS in the ECD using Recent Change and Verify. (See your system administrator if you need assistance in this procedure.) Then restore it using the RST:MHD command. After it has been successfully restored, determine why SUPR stopped before restarting the ENTER process.

NOTE:

The **ENTER** process is reading the first generic tape onto the off-line disk when the active Control Unit (CU) takes a recovery, killing the **ENTER** process. Because it did not have a chance to clean up before it quit, the **ENTER** process leaves the off-line disk marked OFL in the ECD. At this point, mark the disk as OOS in the ECD, restore the OOS disk, and restart the system update.

Problem F

Problem	You would like to start a system update over from the beginning.
Solution	Restore any disks marked OFL to the active state using the instructions from the Problem E solution. Terminate the SUPR Menu Page, remove the SUPR log file <i>/etc/log/suprlog</i> , and start your system update.

Problem G

Problem	In either the "Enter Arguments Page 1 of 2" or "Enter Arguments Page 2 of 2" screen, you want to exit to the SUPR Menu Page without specifying anything.
Solution	Type 107 <CR> at the CMD< line to exit either screen and to return you to the SUPR Menu Page.

⇒ NOTE:

You have entered SUPR to do a system update, but you find after you have gotten into the "Enter Arguments Page 1 of 2" screen that you are missing some information. You would now like to get back to the SUPR Menu Page so you can exit SUPR. Typing **793** will fail at this point because the Enter Arguments Page 1 screen requires that the update method and tape device be specified before exiting the page. Typing **107** will return you to the SUPR Menu Page. Similarly, the Enter Arguments Page 2 screen requires that the destination and the sequence option be specified before returning to the SUPR Menu Page.

Input Messages

System update input messages (commands) function in a variety of ways depending on the options specified in the message. The following is a list of system update input messages and a summary of their functions:

UPD:GEN;ENTER	Enters the new generic into the system; reads the data and writes it onto a disk.
UPD:GEN;PROCEED	Prepares the system for booting from the new generic.
UPD:GEN;COMMIT	Commits to the new generic. Completes the propagation of the new generic into the system after the soak period.
UPD:GEN;CONTINUE	Continues the execution of a system update process.
OP:GEN;READLOG	Requests output of the system update event log. Used if problems occur while updating the system to determine the current state of affairs or for general information about the update.
UPD:GEN;RESTORE	Restores the old generic by removing the new generic from the system. This command is typically used to restore the old generic after a successful BACKOUT and reboot occurred. RESTORE must always use the UCL option.
UPD:GEN;BACKOUT	Prepares the system for booting from the old generic if it is necessary to back out of the new generic.
STOP:GEN	Stops the system update command in progress. Terminates the process associated with a previous UPD:GEN input message.

Refer to the following information products (IPs) for details:

- 254-303-110, *PDS Input Messages Manual*
- 254-303-112, *MML Input Messages Manual*.

Contents

Introduction	12-1
Maintenance Interface	12-1
Maintenance Procedures	12-4
■ How to Maintain System from User Operating Position	12-4
■ How to Set Options on Maintenance Terminals	12-5
■ How to Set Options on the Receive-Only Printer	12-11
■ How to Operate the ROP in a High Humidity Environment	12-11
■ How to Interpret the MTTY Display	12-17
■ How to Access the Emergency Action Pages	12-19
■ How to Access the Common and Application System Display Pages	12-36
■ How to Specify the Initial Display Pages for C/D Terminal in the ECD	12-40
■ How to Access the Status Summary Area Page	12-42
■ How to Access the Common Processor Display Page	12-44
Page Description	12-44
Major and Minor State Display Field Values	12-55
■ How to Access the Control/Display Update Page	12-57
■ How to Access the Operating System Status Page	12-59
■ How to Interpret the Operating System Status Page	12-61

Contents

■ How to Access the Craft Field Maintenance Command Display Pages	12-63
■ How to Execute CFTMTC Commands via Menu Numbers	12-64
■ How to Access the System Update Menu Page	12-73
■ How to Access the DSCH Computer Interconnect (DCI) Status Page	12-75
Major and Minor State Display Field Values for DCI	12-80
■ How to Access the Field Update Page	12-80
Page Description	12-81
Functional Description	12-84
■ How to Access and Interpret the DFC Index Page	12-88
■ How to Access the DFC Status Page	12-90
■ How to Interpret the DFC Status Page	12-96
■ How to Access the DFC Performance Page	12-101
■ How to Interpret the DFC Performance Page	12-104
■ How to Access and Interpret the Disk File System Access Index Page	12-105
■ How to Access the DFSA Status Page	12-107
■ How to Interpret the DFSA Status Page	12-113
■ How to Access the DFSA Performance Page	12-117
■ How to Interpret the DFSA Performance Page	12-120
■ How to Use the Craft Shell and Dialogue Shell with the Input Message Catalog	12-121
Input Message Catalog Database	12-121
Capability Modes	12-122
Shell Configuration	12-123
<i>splinfo</i> Record in the ECD	12-123
ECD Changes for the Dialogue Shell	12-124
Pathname File	12-124
■ How to Field Update the Input Message Catalog	12-125
■ How to Invoke the CFTSHL Help Facility	12-126
■ How to Use the Flexible Input Message Formats	12-126
■ How to Specify the Incore Size of the Message Spooler's Output Message Database in the ECD	12-127

Contents

■ How to Field Update the Output Message Database	12-128
■ How to Abort Commands or Messages	12-129
■ How to Retire Alarms	12-129
■ How to Reinitialize the Craft Interface	12-130
■ How to Switch Devices to Alternate MTTYC	12-131

Introduction

This chapter provides an overview of the software required to perform the various craft interface functions provided by the *UNIX*® Real-Time Reliable (RTR) operating system. The information presented summarizes operation of the Maintenance Teletypewriter (MTTY) terminal and maintenance capabilities for the *UNIX* RTR Operating System Releases 1 and 6.

Maintenance Interface

Maintenance personnel interact with the 3B20D and 3B21D computers primarily through the maintenance terminal. The maintenance interface is normally connected in a duplex configuration. The hardware configuration for the maintenance interface is shown in Figure 12-1. The major components of the maintenance interface configuration are described as follows:

Control Unit (CU): The CUs are equipped with Maintenance Channels (MCHs), Emergency Action Interfaces (EAls), and Direct Memory Access (DMA) circuits.

Maintenance Channel (MCH): The MCH provides a direct link between the two CUs over which system control and maintenance take place. Through the MCH, the on-line computer can access the internal buses of the off-line computer, load instructions, and diagnose portions of the CU even if the microcontrol is not operational. Messages for keeping the off-line CU inactive and for signaling a CU switch take place over this link.

Emergency Action Interface (EAI): The EAI circuit provides access to the computer via the system status register. The EAI is used to force CU configuration, to select and initiate various levels of recovery, to initialize hardware, to disable check circuits, and to provide some CU status information.

Input/Output Processor (IOP): The IOP serves as an interface between a variety of peripheral units and the Central Control (CC). The IOP is connected to the CC via a dual serial channel associated with a DMA controller in the DMA Input/Output (IO) unit.

Maintenance Teletypewriter Controller (MTTYC): The MTTYC resides within the IOP in preassigned locations (slot 0 of IOP 0 and IOP 1) and takes on an appearance similar to other peripheral controllers in the IOPs. Dual port random access memory is used to allow the peripheral interface controller of the IOP to share memory with the microprocessor of the MTTYC. This allows for the reception of jobs and transmittal of data through the IOP to the computer.

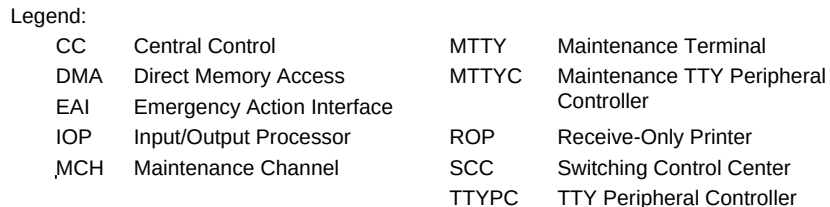
Switching Control Center (SCC): The SCC provides for synchronous data link communication using the BX.25 protocol for error detection and correction. The local terminal or the remote SCC terminal (at the SCC) interfaces with the computer CU through the EAI or the DMA via the IOP and the MTTYC.

Port Switch: The port switch provides a switching interface between the two MTTYCs and one or both switchable devices [an MTTY and/or the Receive-Only Printer (ROP)].

MTTY Display and Keyboard: The display screen of the MTTY terminal provides a split screen. The upper portion of the screen displays information necessary for the user to maintain the system. The lower portion displays a scroll of consecutive input and output messages. Upon request, additional status displays can be presented to augment the information in the normal status display. (Terminal options must be selected before these functions are operable. Refer to the "How to Set Options on Maintenance Terminals" section.)

The MTTY terminal keyboard is used to enter input messages and to select input commands from the menu (lists of commands) on each display page. Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Receive-Only Printer (ROP): A medium-speed line printer (up to 600 lines per minute) can be connected to the MTTYC to record all message output directed to it.



Maintenance Procedures

How to Maintain System from User Operating Position

The user interacts with the 3B20D and 3B21D computers primarily through the MTTY terminal. The MTTY terminal provides message IO facilities. It also provides graphic displays of system status and alarms and allows numeric menu selection for maintenance command input.

The MTTY has two types of displays, EAI and system. In each display, the system critical indicators are displayed across the top of the MTTY terminal screen. The critical indicator status shown is the last updated state if software sanity is lost.

EAI Display: The EAI display is used for recovery of the processor complex during emergencies. The user has direct control of the processor via the EAI. To access the EAI display, press the **EA DISP** special function key, located above the numeric pad on the MTTY terminal keyboard.

System Display: These displays are used primarily for display of hardware status indicators, system critical indicators, and/or menu commands. To access the system displays, press the **NORM DISP** special function key, located above the numeric pad on the MTTY terminal keyboard.

The MTTY terminal has two modes of operation available on any display: command mode and message mode.

Command/Message Mode: In the command/message mode, the MTTY terminal is used primarily for menu command input and for message IO, respectively. The user accesses the desired MTTY mode by pressing the **CMD MSG** special function key, located above the numeric pad on the MTTY terminal keyboard. Pressing this key transfers control between the two input modes.

The MTTYC contains four independent IO channels, one of which has two ports. Connections are as follows:

Channel-0		MTTY
Channel-1		ROP
Channel-2	Port 0	EAI-0
Channel-2	Port 1	EAI-1
Channel-3		SCC

How to Set Options on Maintenance Terminals

The maintenance terminal options can be set only when the “Set-Up Page” is displayed. These options are provided on the maintenance terminal to turn the terminal features on or off. To set these options, refer to the manual for your particular terminal. Tables 12-1, 12-2, and 12-3 show the default option configurations for each of the following KS maintenance terminals:

Table 12-1	KS-22921, L1 (U.S.)
Table 12-2	KS-23554, L5 (U.S.) KS-23554, L6 (International)
Table 12-3	KS-23996, L1 (U.S.) KS-23996, L5 (International)

Table 12-1. KS-22921, L1 Maintenance Terminal Configuration

Option/Function	Desired Selection
Characters per line	80
Scrolling	Jump (line at a time)
Auto-repeat	Off
Screen	Dark background
Cursor	Underline or block
Margin bell	Disabled
Key click	On
Control mode	ANSI*
Line protocol	ASCII DC1/DC3
Automatic wraparound	Disabled
New line	Off
Parity	Enabled
Parity sense	Even
Bits per character	7
Automatic DC1/DC3	Enabled
# (shifted)	0
Interface	Off
Tab stops	None
Local echo	Disabled
Display flashing characters	Enabled
Power	60 Hz
Transmit speed	9600 Asynchronous
Receive speed	9600 Asynchronous

* American National Standards Institute

Table 12-2. KS-23554, L5 and L6 Maintenance Terminal Configuration

Option Name	Factory Default
Date	DD MMM YYYY
Time	HH
Diskette Drive(s)	None
Keyboard Type	KS-22921, L1 Style
Screen Resolution	High
Numeric Coprocessor	Not Installed
Operating Mode	Terminal Only (One Session)
Parallel Port	Terminal (Printer)
Printer Hot-Key	Disabled
Printer Type	Model 5310 (Version 3.0)
DOS Enabled	Disabled
First-Boot Device	Hard Disk
Second-Boot Device	Hard Disk
Third-Boot Device	RAM file
Fourth-Boot Device	CARD file
RAM file Write Protect	Unprotected
Speaker Volume	Loud
Application Interrupts	Enabled
Screen Saver	Disabled
Assignment	Terminal
Communication Speed	9600 Baud
Data Bits/Parity	7 Bits, Even Parity
Handshaking	XON/XOFF
Interface	RS-232
Stop Bits	1-Stop Bit
Communication Buffer	64 Bytes
Transmit Limit	Unlimited Transmit
Assignment	Terminal
Communication Speed	9600 Baud
Data Bits/Parity	7 Bits, Even Parity
Handshaking	XON/XOFF
Interface	RS-232
Stop Bits	1-Stop Bit
Communication Buffer	64 Bytes
Transmit Limit	Unlimited Transmit
Background Color	Black
Cursor Color	Green
Standout Color	Green
Text Color	White
Title Color	Yellow

Table 12-2. KS-23554, L5 and L6 Maintenance Terminal Configuration (Contd)

Option Name	Factory Default
Connection	On-Line
Terminal Mode	KS-22921 Mode
VT100 Mode	VT100 ID
User Defined Keys Lock	Unlocked
User Features Lock	Unlocked
Character Set Mode	Multinational
Margin Bell	No Margin Bell
Warning Bell	Enable Warning Bell
New Line	No New Line
Local Echo	No Local Echo
Columns/Rows	24 Rows by 80 Columns
Scroll	Jump Scroll
Auto Wrap	No Auto Wrap
Video Mode	Light Text, Dark Screen
Control Mode	Interpret Controls
Text Cursor Display	Text Cursor On
Cursor Style	Block Cursor
Status Line	No Status Line
Tabs	No Tabs
Keyboard Language	North American
Keypad Mode	Numeric Keypad
Cursor Key Mode	Normal Cursor Keys
Keys	Typewriter Keys
Lock	Caps Lock
Key Repeat	No Auto Repeat
Key Click	Enable Key Click
Break Key	Enable Break Key
Auto Answer Back	No Auto Answer Back
Printer to Host	No Printer Input to Host
Printer Mode	Normal Print Mode
Print Screen Mode	Print Full Page
Print Mono/Color	Monochrome
Print Page Width	8½ Inches
Print Width	Same as Current Screen Width
Print Terminator	No Terminator

Table 12-3. KS-23996, L1 and L5 Maintenance Terminal Configuration

For List 1 and List 5 Terminals	
Option Name	Factory Default
Terminal Emulation	VT100
EIA/AUX Stop Bits	1 Bit
EIA/AUX Baud Rate	9600 Baud
EIA/AUX Data Bits	7 Bits
EIA/AUX Parity	Even Parity
Session	One
Host	EIA (RS232)
Printer	None
Enhanced	Off
COMM Mode	Full Duplex
Terminal Emulation	VT100
Host	EIA (RS232)
Enhanced	Off
Color Mode	Direct
Auto Scroll	On
Auto Font Load	On
Screen Save	Off
Session	One
Printer	None
Warning Bell	On
Auto Wrap	Off
Bell Volume	9 (Maximum)
Auto Page	Off
Monitor Mode	Off
Columns	80
Display Cursor	On
Scroll	Jump
Width Change Clear	Off
Speed	Normal
Rows	24
Cursor Style	Steady Block
Screen Length	26
Screen Video	Normal
Auto Adjust Cursor	On

Table 12-3. KS-23996, L1 and L5 Maintenance Terminal Configuration (Contd)

For List 1 and List 5 Terminals	
Option Name	Factory Default
Character Set	ANSI
Margin Bell	Off
Key Click	On
Key Mode	ASCII
Key Repeat	Off
Key Lock	Caps
Enter/Return Key	<CR>
ALT Key	ALT
Pound Key	US
~&'	~&'
Backspace	BS/BS
Escape	Esc
F1 to F5	Function Key
,< & .>	,< & ,
EIA/AUX Baud Rate	9600 Baud
EIA/AUX Parity	Even Parity
EIA/AUX Parity Check	Off
EIA/AUX Break	250
EIA/AUX Mode Control	Off
EIA/AUX DISC	60 msec
EIA/AUX Data Bits	7 Bits
EIA/AUX Stop Bits	1 Bit
EIA/AUX Transmit	X-ON X-OFF
EIA/AUX Receive	X-ON X-OFF
EIA/AUX Transmit Pace	Baud
COMM Mode	Full Duplex
RECEIVE <DEL>	DEST BS
Send Region	Screen
RECEIVE <CR>	<CR>
Null Suppress	On
Local Echo	No Local Echo
Send ACK	Off
Send END	Region
Send Block Term	None

Table 12-3. KS-23996, L1 and L5 Maintenance Terminal Configuration (Contd)

For List 1 and List 5 Terminals	
Option Name	Factory Default
Print Mode	Normal
Print Reg	Screen
Secondary Receive	Off
Print Block Term	None
Status Line	Off
Cursor Keypad	Cursor
UPSS	ANSI Sup
Feature Lock	Off
Function Key Lock	On
Numeric Keypad	Numeric
Print	Multinational
ANSI ID	VT100
Send Data	All
Auto Init Tabs	Off
Answer Back	Off
PROG	F1 (NO OP)
Lucent Technologies Defaults	Default Term
Any other terminal function that may inhibit any requirement	Off

How to Set Options on the Receive-Only Printer

To set the options provided on the Receive-Only Printer (ROP), refer to the user's guide for your particular printer. Set the options, if available, as listed in Tables 12-4, 12-5, and 12-6.



NOTE 1:

Systems equipped with the Teletype 5310 teleprinter have the capability of suspending output to the ROP temporarily. Pressing the **BREAK** key on the printer will suspend the output for 2 minutes or until the **BREAK** key is pressed again, whichever occurs first. The **HERE IS** key will do the same as the **BREAK** key. This operation is not recommended because messages may be lost. This capability has been disabled on systems with TN983 MC4C132A1B, Issue 1 and later, and systems with the UN583/UN597.



NOTE 2:

Ensure that the parity is set to even. While even parity has always been specified, flow control on systems equipped with TN983 MC4C132A1B, Issue 1 and later, or the UN583/UN597 will only work correctly if the parity is set to even.

How to Operate the ROP in a High Humidity Environment

High humidity causes the printer paper to change dimensionally and to lose its tear resistance. Therefore, the following precautions should be taken:

- Check tractor spacing and readjust, if necessary.
- Prevent back pressure from being exerted on the paper as it exits the printer by doing one of the following:
 - Remove the printed paper frequently from the collection basket.
 - Remove the paper collection basket from the pedestal and place it on the floor.

Table 12-4. Model 5310 ROP Option Selection

Option/Function	Desired Selection
Characters per inch	10*
Line per inch	6
Form length (in-line)	66
Left margin	1*
Right margin	80*
Top margin	1
Bottom margin	66
Character set	ASCII
Alternate font	ASCII
Wrap at right margin	Yes
Print with controls	No
Control sequence emulation	ANSI Emulation
Character view	Disable (Manual)
Auto LF on CR	No
Auto CR on LF	No
Horizontal tab settings	1,9,17,25,33,41,49,57,65,73
Vertical tab settings	N/A
Line protocol	EIA (RS-232-C)
Parity	Even
Speed	1200 Baud
Flow control	DC3/DC1 Enabled
Secondary request to send	N/A
DC2/DC4 control	No
Disconnect on paper out	Yes
Disconnect on EOT received	No
Low Paper	No
Local echo	No
Send answer-back on auto answer	No

* By changing to the following options, ten more characters can be printed per line and the left margin can accommodate holes for putting the printout in binders:

Characters per inch	12
Left margin	7
Right margin	96

Table 12-5. Model 577 ROP Option Selection

Option/Function		Desired Selection	
00	Emulation mode	02	DEC * emulation
01	Emulation level	01	LA210 mode
02	EEPROM store	01	Enable
10	Form length	66	Lines
12	Line spacing	01	6 LPI
13	Character pitch	10	CPI
14	Print quality	01	DP
17	Left margin		0.2 inch
18	Right margin		8.2 inches
19	Print direction	02	Bidirectional
21	Font for zero	02	Slashed zero
22	Skip perforation	01	None
23	Wraparound	02	Auto wrap
24	Data error	01	Reverse question mark
25	LQ CG selection	01	Built-in CG
26	Define GL,G0	01	U.S. ASCII
27	Define G1	01	U.S. ASCII
28	Define GR,G2	01	U.S. ASCII
29	Define G3	01	U.S. ASCII
30	CG selection	01	Built-in CG
31	Alarm tone	01	On
33	Top margin	01	None
34	TOF position	01	Disable
35	Switch tone	01	Enable
40	Interface	02	RS232C
41	Data bits	01	7 bits
42	Protocol	06	XON/XOFF (ON/OFF-LINE & PE)
43	Stop bits	01	1 bit
44	Parity	04	Even
45	Baud rate	05	1200 BPS
46	RTS	01	ON/OFF-LINE
47	OFF-LINE Deselect	01	Deselect
50	PE detection	01	PE Valid

* DEC is a trademark of Digital Equipment Corporation.

Table 12-5. Model 577 ROP Option Selection (Contd)

Option/Function		Desired Selection	
62	Overrun buffer	03	1024 bytes
63	Initial state	01	On-line
70	A.S.F.	01	Disable
71	Auto loading	02	Disable
72	FF speed	01	12 IPS
73	ON-LINE paper feed	02	Enable
74	Reset + TEAR FORM	01	Normal
80	Automatic CR on LF	01	LF only
81	Automatic LF on CR	01	CR only
82	CAN code	01	Normal
90	Hex dump	01	Normal mode
91	Test print pattern	02	Sliding pattern
92	Test print character	H	

Table 12-6. Model 602 ROP Option Selection

Option/Function		Desired Selection
Font	Print Mode	Utility
Font	Pitch	10 CPI
Font	Proportional Spacing	No
Font	Style	Normal
Font	Size	Single
Symbol Sets	Character Set	Set I
Symbol Sets	Language Set	American
Symbol Sets	Zero Character	Slashed
Symbol Sets	Code Page	USA
Rear Feed	Line Spacing	6 LPI
Rear Feed	Form Tear-Off	Off
Rear Feed	Skip Over Perforation	No
Rear Feed	Page Length	11 inches
Rear Feed	Gap Control	Auto Gap
Bottom Feed	Line Spacing	6 LPI
Bottom Feed	Form Tear-Off	Off
Bottom Feed	Skip Over Perforation	No
Bottom Feed	Page Length	11 inches
Bottom Feed	Gap Control	Auto Gap
Top Feed	Line Spacing	6 LPI
Top Feed	Bottom Margin	Valid
Top Feed	Page Length	11 inches
Top Feed	Gap Control	Auto Gap
Top Feed	Wait Time	1 second
Set-Up	Graphics	Uni-directional
Set-Up	Receive Buffer Size	64K
Set-Up	Paper Out Override	No
Set-Up	Print Registration	0
Set-Up	Operator Panel	Full Operation
Set-Up	Reset Inhibit	No
Set-Up	Print Suppress Effect	Yes
Set-Up	Auto LF	No
Set-Up	Auto CR	No
Set-Up	SI Select Pitch (10 CPI)	17.1 CPI
Set-Up	SI Select Pitch (12 CPI)	20 CPI
Set-Up	Time Out Print	Valid
Set-Up	Auto Select	No
Set-Up	Printhead Gap Adjust	0

Table 12-6. Model 602 ROP Option Selection (Contd)

Option/Function		Desired Selection
Parallel I/F	I-Prime	Buffer Print
Parallel I/F	Pin 18	+5v
Serial I/F	Parity	Even
Serial I/F	Serial Data 7/8 Bits	7 Bits
Serial I/F	Protocol	X-ON/X-OFF
Serial I/F	Diagnostic Test	No
Serial I/F	Busy Line	RTS
Serial I/F	Baud Rate	1200 BPS
Serial I/F	DSR Signal	Valid
Serial I/F	DTR Signal	Ready On Select
Serial I/F	Busy Time	200 ms
Printer Control	Emulation Mode	IBM* PPR

* IBM is a registered trademark of International Business Machines Inc.

How to Interpret the MTTY Display

The MTTY header line, shown in Figure 12-2, displays the following fields:

- Office name and type from the Equipment Configuration Database (ECD)
- Release and issue
- Terminal identification
- Date (in format specified in *eaiopt* record in the ECD) and 24-hour clock.

The second and third lines of the MTTY screen display the state of the system critical indicators. These lines are present on all system displays.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _				ENTER 100 FOR PAGE INDEX			

Figure 12-2. MTTY Terminal Critical Indicator Display

Table 12-7 lists the critical system indicators, a description of each of the critical indicators, and the suggested user action.

Table 12-7. System Critical Indicators

Indicator	Description	Action
SYS EMER	Indicates (when flashing) that the 3B20D/3B21D computers have lost sanity.	Emergency action required.
CRITICAL	Indicates a total system loss or that a major portion of the system is lost.	Emergency action required.
MAJOR	Indicates a partial loss of system capability or a failure of a type such that a similar failure could result in a critical condition.	Immediate action required.
MINOR	Indicates a minor loss of system capability or some condition requiring attention.	Requires action but not immediately.
SYS NORM	Indicates that the system is running normally.	None.
Other Indicators	Application definable.	Refer to local application procedures.

How to Access the Emergency Action Pages

The **EA DISP** function key gives access to the Emergency Action Page display. (Refer to Figures 12-3, 12-4, 12-5, 12-6, 12-7, and 12-8.) The MTTYC automatically displays this page on initialization and when it detects a loss of sanity in the active CC. This page may be requested at any other time by pressing the **EA DISP** function key. Pressing the **EA DISP** function key when the Emergency Action Page is already displayed erases the page and then redisplay it.

Emergency Action Page functions are invoked by executing the commands displayed as menu items on the Emergency Action Page. Commands are executed by entering a two-digit number followed by a carriage return. A list of Emergency Action Page commands and a description of their actions are shown in Table 12-8.



NOTE 1:

The command prompt for the TN83 remains CMD: _.

The command prompt for the TN983/UN583/UN597 is CMD< _.



NOTE 2:

A lighted EMERGENCY ACTION ENABLED lamp on the TN10/TN11 EAI circuit pack (located in the main store mounting unit adjacent to the power switch circuit pack) indicates that an emergency action function is in effect.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION			DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM	
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS				
CMD: _				_____EMERGENCY ACTION PAGE _____				
				MTTY_4				
CU-0	FOFL	EAI-0_ASW	PRM-0	E800	0000	0000	0044	79 EA 04
CU-1_ACT	RUN FONL	EAI-1_ASW	PRM-1	E800	0000	0000	0044	79 DA 14
SCCS_								
SET CLR		CU-0 CU-1	SET CLR					
10 FONL-0	20 21 PRI-DISK	30 31 BACKUP-ROOT	50 APPL					
11 FONL-1	22 23 SEC-DISK_SET SET	32 33 MIN-CONFIG_SET	51 INIT					
12 FONL-ACT	24 25 INH-TIMER_INH INH	34 35 INH-HDW-CKL	52 BOOT					
13 CLR-FDNL	26 27 PRM-TRAP	36 37 INH-SFT-CHK_INH	53 BOOT:ECD					
		38 39 INH-ERR-INT_	54 BOOT:MEM					
14 CLR-EAI	28 PRM-DUMP	40 41 INH-CACHE	55 LDTAPE-0					
15 CFT-INIT		42 43 APPL-PARAM	56 LDTAPE-1					

Figure 12-3. Emergency Action Page for TN83

OFFICE NAME AND TYPE	GENERIC AND ISSUE	TERMINAL IDENTIFICATION	DATE AND 24-HOUR CLOCK
SYS EMER CRITICAL MAJOR MINOR BLDG/PWR BLD INH CKT LIM			SYS NORM
TRAFFIC SYS INH CU CU PERPH OS LINKS			
CMD<_			
EMERGENCY ACTION PAGE			
MCRT 3			
CU 0	ACT RUN FONL	EAI 0 ASW	PRM 0 E841 0002 891C 0004 79 EA 04
CU 1	FOFL	EAI 1 ASW	PRM 1 0000 0000 0000 0000 00 00 00
SCCS			
SET CLR			
10 FONL 0	20 21 PRI DISK	CU 0 CU 1	SET CLR
11 FONL 1	22 23 SEC DISK	30 31 BACKUP ROOT	50 APPL
12 FON1 ACT	24 25 INH TIMER	32 33 MIN CONFIG SET	51 INIT
13 CLR FON1	26 27 PRM TRAP	34 35 INH HDW CHK	52 BOOT
		36 37 INH SFT CHK	53 BOOT+ECD
		38 39 INH ERR INT	54 BOOT+MEM
14 CLR EAI		40 41 INH CACHE	55 LDTAPE 0
15 CFT INIT	LAST APPL PARAM	42 43 APPL PARAM	56 LDTAPE 1
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ****			
<			

Figure 12-4. Emergency Action Page for TN983

OFFICE NAME AND TYPE	GENERIC AND ISSUE	TERMINAL IDENTIFICATION	DATE AND 24-HOUR CLOCK
SYS EMER CRITICAL MAJOR MINOR BLDG/PWR BLD INH CKT LIM			SYS NORM
TRAFFIC SYS INH CU CU PERPH OS LINKS			
CMD<_			
EMERGENCY ACTION PAGE			
MCRT 3			
CU 0	ACT RUN FONL	EAI 0 ASW	PRM 0 E841 0002 891C 0004 79 EA 04
CU 1	FOFL	EAI 1 ASW	PRM 1 0000 0000 0000 0000 00 00 00
SCCS			
SET CLR			
10 FONL 0	20 21 PRI DISK	CU 0 CU 1	SET CLR
11 FONL 1	22 23 SEC DISK	30 31	BACKUP ROOT 50 APPL
12 FON1 ACT	24 25 INH TIMER	32 33	MIN CONFIG SET 51 INIT
13 CLR FON1	26 27 PRM TRAP	34 35	INH HDW CHK 52 BOOT
14 CLR EAI		36 37	INH SFT CHK 53 BOOT+ECD
15 CFT INIT		38 39	INH ERR INT 54 BOOT+MEM
16 EAI INIT	LAST APPL PARAM	40 41	INH CACHE 55 LDTAPE 0
		42 43	APPL PARAM 56 LDTAPE 1
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ****			
<			

Figure 12-5. Emergency Action Page for UN583 or UN597 without Alternate Boot Disk Feature (pre R21.35)

```

OFFICE NAME & TYPE  RELEASE INFO  TERMINAL ID  DATE & 24 HR CLOCK
SYS EMER  CRITICAL  MAJOR  MINOR  BLDG/PWR  BLD INH  CKT LIM  SYS NORM
TRAFFIC  SYS INH  CU  CU PERPH  OS LINKS
CMD<
MCRT 3
CU 0 ACT RUN  EAI 0 ASW  PRM 0 E841 0002 891C 0004 79 EA 04
CU 1  EAI 1 ASW  PRM 1 0000 0000 0000 0000 00 00 00
SCCS
Set Clr  CU 0 CU 1  Set Clr
10 Fonl CU 0  20 21 Pri Disk  30 31 Backup Root  50 APPL
11 Fonl CU 1  22 23 Sec Disk  32 33 Min Config  51 INIT
12 Fonl Act CU  24 25 Inh Timer  34 35 Inh Hdw Chk  52 BOOT
13 Clr CU Fonl  26 27 PRM Trap  36 37 Inh Sft Chk  53 BOOT+ECD
14 Clr EAI  28 29 Boot MHDs 0/1 1/0  38 39 Inh Err Int  54 BOOT+MEM
15 CFT Init  40 41 Inh Cache  55 Ldtape 0
16 EAI Init  Last Appl Param  42 43 Appl Param  56 Ldtape 1
-----

```

REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE/PDS ***

<

Figure 12-6. Emergency Action Page for UN597 with Alternate Boot Disk Feature (pre R21.35)

```

OFFICE NAME & TYPE  RELEASE INFO      TERMINAL ID      DATE & 24 HR CLOCK
SYS EMER  CRITICAL  MAJOR    MINOR  BLDG/PWR  BLD INH  CKT LIM  SYS NORM
TRAFFIC   SYS INH   CU      CU PERPH  OS LINKS
CMD<
                                MCRT 3
CU 0 ACT RUN FONL          EAI 0 ASW  PRM 0 E841 0002 891C 0004 79 EA 04
CU 1          FOFL        EAI 1 ASW  PRM 1 0000 0000 0000 0000 00 00 00
SCCS
10 Fonl CU 0      Set Clr          CU 0 CU 1 Set Clr
11 Fonl CU 1      20 21 Pri Disk    30 31ackup Root          50 APPL
12 Fonl Act CU    22 23 Sec Disk    32 33in Config      SET 51 INIT
13 Clr CU Fonl    24 25 Inh Timer    34 35nh Hdw Chk          52 BOOT
14 Clr EAI        26 27 PRM Trap    36 37nh Sft Chk          53 BOOT+ECD
15 CFT Init                          38 39nh Err Int          54 BOOT+MEM
16 EAI Init                          40 41nh Cache            55 Ldtape 0
18 Lock MTTY                          42 43ppl Param            56 Ldtape 1
-----
                                REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE/PDS ***
<

```

Figure 12-7. Emergency Action Page for UN597 without Alternate Boot Disk Feature (R21.35 and later)


```

OFFICE NAME & TYPE  RELEASE INFO      TERMINAL ID      DATE & 24 HR CLOCK
SYS EMER  CRITICAL  MAJOR    MINOR    BLDG/PWR  BLD INH  CKT LIM  SYS NORM
TRAFFIC   SYS INH   CU      CU PERPH OS LINKS
CMD<
----- EMERGENCY ACTION PAGE -----
MCRT 3
CU 0 ACT RUN FONL EAI 0 ASW PRM 0 E841 0002 891C 0004 79 EA 04
CU 1          FOFL EAI 1 ASW PRM 1 0000 0000 0000 0000 00 00 00
SCCS
10 Fonl CU 0 SET CLR CU 0 CU 1 SET CLR
11 Fonl CU 1 20 21 Pri Disk 30 31 Backup Root 50 Appl
12 Fonl Act CU 22 23 Sec Disk 32 33 Min Config SET 51 Init
13 Clr CU Fonl 24 25 Inh Timer 34 35 Inh Hdw Chk 52 Boot
14 Clr EAI 26 27 PRM Trap 36 37 Inh Sft Chk 53 Boot+ECD
15 CFT Init 28 29 Boot MHDs 0/1 1/0 38 39 Inh Err Int 54 Boot+MEM
16 EAI Init 40 41 Inh Cache 55 Ldtape 0
18 Lock MTTY Last Appl Param 42 43 Appl Param 56 Ldtape 1
-----

```

REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE/PDS ***

<

Figure 12-8. Emergency Action Page for UN597 with Alternate Boot Disk Feature (R21.35 and later)

```
OFFICE NAME & TYPE   RELEASE INFO      TERMINAL ID          DATE & 24 HR CLOCK
SYS EMER    CRITICAL    MAJOR        MINOR     BLDG/PWR   BLD INH   CKT LIM   SYS NORM
TRAFFIC     SYS INH         CU           CU PERPH  OS LINKS
CMD<

*****
*                                     *
*                                     *
*                                     *
* Login:                               *
* Password:                            *
*                                     *
*****
* ENTER LOGIN *
*                                     *
*                                     *
*****

-----

REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE/PDS ***
<
```

Figure 12-9. Lock Screen Display for UN597

Table 12-8. Emergency Action Page Commands

Command	Description
10	Inhibits automatic processor recovery switch capability. Forces Control Unit 0 (CU 0) to be the on-line processor and CU 1 to be the off-line processor. This will result in recovery action if CU 0 was not on-line at the time of the force. (Refer to Note 1.)
11	Same as command 10 except that CU 1 is forced on-line and CU 0 is forced off-line. (Refer to Note 1.)
12	Same as command 10 except the currently active processor is forced on-line and the other processor is forced off-line. (Refer to Note 1.)
13	Removes on-line and off-line forces and allows automatic processor recovery action to determine the on-line and off-line CUs.
14	Clears the CU forces, disk forces, and Inhibit Sanity Timer. This is equivalent to entering commands 13, 21, 23, and 25. For systems equipped with TN983s/UN583/UN597s, the next state information is also cleared. This is equivalent to entering commands 31, 33, 35, 37, 39, 41, and 43. For UN597 with Alternate Boot Disk (ABD), the selection of the alternate boot disks are reset to MHD 0 and MHD 1. This is the equivalent of entering a command of 29.
15	Refer to Note 5. If application parameter is 1: ▪ Terminates and restarts all craft processes. ▪ Restarts MTTY controllers. If application parameter is 2: ▪ Terminates and restarts all killable nonessential user and supervisor processes. ▪ Restarts all MTTY and TTY controllers. If application parameter is 3: ▪ Terminates and restarts all killable essential and nonessential user and supervisor processes. ▪ Reboots the IOPs.

See notes at end of table.

Table 12-8. Emergency Action Page Commands (Contd)

Command	Description
16	Clears the EAI hardware on a specific CU (for UN583/UN597 only).
18	(UN597 R21.35 and later) Locks the MTTY. Locking the MTTY prevents access to the terminal until a user successfully logs in. (Refer to Note 6 and Figure 12-9.)
20	Inhibits automatic processor recovery disk unit selection and forces both processors to access their primary disk units on a boot. The primary disk for CU 0 is MHD 0. The primary disk for CU 1 is MHD 1. (Refer to Notes 1 and 4.)
21	Removes force on primary disk unit select.
22	Same as command 20 except it forces the processors to access their secondary disk units. The secondary disk for CU 0 is MHD 1. The secondary disk for CU 1 is MHD 0. (Refer to Notes 1 and 4.)
23	Removes force on secondary disk unit select.
24	Inhibits the sanity timer from expiring and initiating automatic recovery action.
25	Removes the sanity timer inhibit.
26	Releases any trapped failure Processor Recovery Messages (PRMs), and causes the next failure PRM to be displayed in reverse video and to remain displayed until released.
27	Releases any trapped failure PRMs and allows further PRMs to be displayed.
28	(TN83 ONLY) Dumps to the display and printer the contents of the buffer in the active CU containing nonfatal error PRMs generated during the most recent processor recovery. (Refer to Note 2.) (UN597 With ABD Feature ONLY) Select an alternate boot disk.
29	(UN597 With ABD Feature ONLY) Clear selection of an alternate boot disk. Reset boot disks back to MHD 0 and MHD 1.
30	Forces the processor to initialize from the backup root file system. (Refer to Note 3.)
31	Allows the processor to initialize from the primary root file system. (Refer to Note 3.)

See notes at end of table.

Table 12-8. Emergency Action Page Commands (Contd)

Command	Description
32	Forces the processor to initialize only the <i>UNIX</i> RTR operating system. The application software is not initialized.
33	Allows the processor to initialize both the <i>UNIX</i> RTR operating system and application software.
34	Inhibits hardware checks from initiating automatic recovery action.
35	Allows the hardware checks to initiate automatic recovery action.
36	Inhibits software checks from initiating automatic recovery action.
37	Allows software checks to initiate automatic recovery action.
38	Inhibits error interrupts.
39	Allows error interrupts.
40	Inhibits use of cache memory.
41	Allows use of cache memory.
42	Allows the setting of a parameter which is made available to the application software or to the craft initialization processing. (Refer to Note 5.) When set for application software processing, the valid values for the parameter will be application dependent.
43	Clears the application parameter.
50	Notifies the application software to initialize. (Refer to Notes 3 and 5.) The handling of the notification will be application dependent (that is, applications may choose to ignore this notification).
51	Forces initialization of the <i>UNIX</i> RTR operating system (level 1 initialization). (Refer to Note 3.)
52	Forces bootstrap and reloads the <i>UNIX</i> RTR operating system from disk (level 2 initialization). (Refer to Note 3.)
53	Same as command 52 plus reloads equipment configuration data (level 3 initialization). (Refer to Note 3.)
54	Same as command 53 plus clears the memory (level 4 initialization). (Refer to Note 3.)

See notes at end of table.

Table 12-8. Emergency Action Page Commands (Contd)

Command	Description
55	Loads selected disk from tape unit 0. (Refer to Note 4.)
56	Loads selected disk from tape unit 1. (Refer to Note 4.)

Notes:

1. Manually forcing a CU (commands 10, 11, or 12) will prevent diagnostics from running against either unit. Attempts to run diagnostics will result in messages indicating Conditional All Tests Passed (CATP). Also, forcing a CU or MHD (commands 10, 11, 12, 20, or 22) could inhibit system recovery upon detection of a fault. This condition can be cleared by entering command 14.
2. PRM-DUMP Command is supported on the **TN83 Only**.
3. Commands 30 through 43 on systems equipped with TN83s builds the next state information which is sent to the EAI/CU when commands 50 through 56 are entered. Commands 30 through 43 entered on a device (that is, MTTY or SCC terminal) connected to one MTTYC will not be seen on a device connected to the other MTTYC.

On systems equipped with TN983/UN583/UN597, commands 30 through 43 are sent to the EAI/CU immediately, which allows the other MTTYC to read this information and update its next state information. Therefore, commands entered on devices connected to different MTTYCs will be seen by both devices. Like the TN83, commands 50 through 56 also send the next state information to the EAI/CU.

Commands 30 through 43 builds the next state information only. The state does not take affect until an initialization command (51 through 54) is entered. They remain in effect during subsequent automatic bootstraps until they are manually removed and the system is manually initialized again.
4. Commands 55 and 56 require a CU to be forced on-line (commands 10, 11, or 12) and a disk unit to be selected (commands 20 or 22).

Table 12-8. Emergency Action Page Commands (Contd)

Notes:

5. If the MTTYC is a TN983/UN583/UN597, it is possible to have the application initialization (Poke **50**) or the craft initialization (Poke **15**) command canceled. This can happen if the user enters another poke before the *UNIX* system process reads the information from the EAI circuit pack. This time is normally 5 seconds or less. The user can safely enter another poke without canceling the command after the application parameter field is cleared, a PRM is displayed which indicates that the command has been accepted by the *UNIX* system process, or the application parameter is displayed in the last *Appl Param* field (for application initialization only).

For an application initialization poke, it is recommended to wait until the application parameter is displayed in the last *Appl Param* field before entering another poke. This is recommended because if another poke is entered too soon, the application parameter field is cleared in the same way as if the command was accepted by the *UNIX* system process. Thus, an incorrect indication is given. For a craft initialization poke, the application parameter is not moved to last *Appl Param* field. With TN983 MC4C132A1, Issue 4, the application parameter is not cleared after it is read by the *UNIX* system process. Therefore, the safest procedure after entering a craft initialization poke is to check for the craft initialization PRM "EBda 7400 ..." before entering another poke.

6. This command is used to prevent unauthorized access to the MTTY. Once in a locked state, access to the MTTY is blocked until a user successfully logs in. With a successful login, the MTTY is returned to a state of normal operation with access to DAP display pages, EAI, and craft commands.

The command is acknowledged with NG when MCC Login Protection is not activated in the ECD or when the capability is activated, but there are no personal records in the Commands Restriction authority file.

In addition to the manual request to lock the MTTY, the MTTY may also lock up following a time period of keyboard inactivity. The automatic locking of the MTTY is part of the MCC Login Protection capability. This functionality is driven by a value in the ECD that is evaluated when MCC Login Protection is in effect. The value may be zero or in the range of five to thirty minutes. A value of zero indicates that the MTTY will never time out.

When a command is entered on the Emergency Action Page (refer to Figures 12-3, 12-4, 12-5, 12-6, 12-7, and 12-8), it is parsed by the MTTYC. If the parsing shows that the command is valid, an OK acknowledgment is output; otherwise, NG (command not good) is sent. For valid inputs, the MTTYC generates commands to both EAI. After these commands are executed by the EAI, the MTTYC requests status and uses this status to update the EAI display

indicators. Table 12-9 lists the display indicators and describes their meaning.

Table 12-9. Emergency Action Page Display Indicators

Indicator	Meaning
MTTY (TN983/TN83/UN583/UN597)	This single digit increments once every 2 seconds to indicate the sanity of the MTTYC and its ability to update the display. If this digit is not incrementing, none of the indicators on the display can be considered valid.
EAI-0 and -1	Reverse video (light characters on dark background) indicates certain states of the EAI as follows: ASW Indicates EAI is communicating with the MTTYC and that internal audits are passing. ERR Indicates EAI is communicating with the MTTYC but that internal audits are failing. OOS Indicates EAI is not communicating with the MTTYC. In this condition, the state of the rest of the display indicators associated with the corresponding CU cannot be considered valid.
CU-0 and -1	When CU-0 or CU-1 are in reverse video, it indicates certain states of the processors as follows: ACT Implies the indicated processor is on-line. RUN Implies that a processor is executing main store instructions. FONL Verifies the forced on-line condition. FOFL Verifies the forced off-line condition. RCVRY Implies that the indicated processor is executing recovery microcode.

Table 12-9. Emergency Action Page Display Indicators (Contd)

Indicator	Meaning
PRM-0 and -1	Normal video shows dynamic PRMs. Reverse video shows the first failing PRM received after execution of command 26. (Refer to 254-303-103, <i>Processor Recovery Messages Guide</i> , for interpretation of these codes.)
SCCS	ACT in reverse video indicates that the SCC channel of the MTTYC, which is connected to the MTTY terminal, has established BX.25 level 2 protocol communication over the synchronous data link. If the SCC connected to one MTTYC is active but the MTTY terminal is connected to the other MTTYC, this indicator is not present even though SCC communication is possible.
SET or INH	When SET or INH is shown in reverse video, it indicates the current state of the associated function. For commands 20 through 27, separate indicators are provided for each CU. For commands 30 through 41, the SET or INH indicators apply to the active CU.
30 — 43	When 30-43 is shown in reverse video, it indicates the “next-state” information which will be sent to the processor on the next execution of commands 50 through 56.

In addition, the TN983/UN583/UN597 protects the system from inadvertent user error. Certain critical EAI pokes will require user verification. Refer to Table 12-10. For each poke, the user will be asked to verify their intentions. The question asked will be poke-specific.

Table 12-10. Emergency Action Page Poke Prompts

Poke	Question
10 FONL-0	Force CU? (y/n)
11 FONL-1	Force CU? (y/n)
15 CFT-INIT	Craft initialize? (y/n)
16 EAI-INIT	Reset EAI? (0/1)
	Reset EAI x (y/n)
	Where x is 0 or 1
18 Lock MTTY	Lock? (y/n)
24 INH-TIMER	Inhibit sanity timer? (y/n)
28 Set Boot MHD	Select boot device? (0-15)
	Set boot devices 0/x? (y/n)
	Where x is 1,3,5,7,9,11,13,15
	Set boot devices z/1? (y/n)
	Where z is 0,2,4,6,8,10,12,14
29 Clr Boot MHD	Set boot devices 0/1? (y/n)
50 APPL	Initialize? (y/n)
51 INIT	Initialize? (y/n)
52 BOOT	Boot? (y/n)
53 BOOT+ECD	Boot? (y/n)
54 BOOT+MEM	Boot? (y/n)
55 LDTAPE-0	Boot? (y/n)
56 LDTAPE-1	Boot? (y/n)

With the TN983 MC4C132A1B, Issue 1 and later, or the UN583/UN597, the setting of the Backup Root and Min Config parameters are tested when a boot command (52, 53, or 54) is entered. The question displayed reflects their settings.

Backup Root	Min Config	Question
31 (Clr)	33 (Clr)	Boot? (y/n)
31 (Clr)	32 (Set)	Min Config Set! Boot? (y/n)
30 (Set)	33 (Clr)	Backup Root Set! Boot? (y/n)
30 (Set)	32 (Set)	BkRt/MinCfg Set! Boot? (y/n)

Only if the user responds **y** or **Y** to the confirm the question will the specified action be taken and the OK acknowledgment printed. In all other cases, including intercharacter time-out, the action will not be performed. The confirm question will be displayed on the CMD line of the Emergency Action Page. For example:

TN983/UN583/UN597 displays:

CMD<

User types: CMD< 54

TN983/UN583/UN597 responds:

CMD< 54 - Boot? (y/n)

User types: CMD< 54 - Boot? (y/n) **y**

TN983/UN583/UN597 responds:

CMD< 54 - Boot? (y/n) **OK**

If the user responds with something other than **y** or **Y**, the TN983/UN583/UN597 will respond with:

CMD< 54 - Ignored

If the user does not respond before the intercharacter timer expires, the TN983/UN583/UN597 responds:

CMD< 54 - Boot? (y/n) ?T

How to Access the Common and Application System Display Pages

The common and application system display pages may be accessed by pressing the **NORM DISP** function key. Under normal operating conditions, pressing this key erases the emergency action page and displays one of the system pages. The system page displayed depends upon the state of the system and the display at the time the request for display is made.

Information to be displayed on the MTTY terminal is divided into display pages. Each page is assigned a unique page identifier. To access a desired system display page, use the following procedure:

Step	Procedure
1	Select a system display by pressing the NORM DISP function key.
2	Place the terminal in the command mode by pressing the CMD MSG function key.
	 NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 100<CR> to obtain the Page Index display. (If you have RTR Release 21.3 or earlier, refer to Figure 12-10.) (If you have RTR Release 21.5 through Release 21.15, refer to Figure 12-11.) (If you have RTR Release 21.17 or later, refer to Figure 12-12.)
4	Access the desired page by entering the three-digit command for the desired page followed by <CR> .

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<				100 - PAGE INDEX			
CMD PAGE TITLE				CMD PAGE TITLE			
101 - STATUS SUMMARY AREA				107 - SYSTEM UPDATE			
102 - COMMON PROCESSOR DISPLAY				109 - FIELD UPDATE			
103 - C/D UPDATE				110 - DFC INDEX PAGE			
104 - OS STAU PAGE				198 - RCV/SG			
105 - CRAFT FM 01				199 - RCV/ECD			
106 - CRAFT FM 01							
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-10. Page Index for RTR Release 21.3 and Earlier

⇒ NOTE:

The version of the Page Index shown in Figure 12-10 is only valid for RTR Release 21.3 and earlier. The Page Index is also referred to as an Index Page.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				100 - PAGE INDEX			
CMD PAGE TITLE				CMD PAGE TITLE			
101 - STATUS SUMMARY AREA				107 - SYSTEM UPDATE			
102 - COMMON PROCESSOR DISPLAY				109 - SOFTWARE UPDATE INDEX PAGE			
103 - C/D UPDATE				110 - DFC INDEX PAGE			
104 - OS STAUUS PAGE				198 - RCV/SG			
105 - CRAFT FM 01				199 - RCV/ECD			
106 - CRAFT FM 01							
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-11. Page Index for RTR Release 21.5 Through Release 21.15

NOTE:

The version of the Page Index shown in Figure 12-11 is only valid for RTR Release 21.5 through Release 21.15. For more information, refer to Chapter 21, "Software Update."

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				100 - PAGE INDEX			
CMD	PAGE TITLE			CMD	PAGE TITLE		
101	- STATUS SUMMARY AREA			107	- SYSTEM UPDATE		
102	- COMMON PROCESSOR DISPLAY			108	- DCI STATUS PAGE		
103	- C/D UPDATE			109	- SOFTWARE UPDATE INDEX PAGE		
104	- OS STAUUS PAGE			110	- DFC INDEX PAGE		
105	- CRAFT FM 01			198	- RCV/SG		
106	- CRAFT FM 01			199	- RCV/ECD		
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-12. Page Index for RTR Release 21.17 and Later

⇒ NOTE:

The version of the Page Index shown in Figure 12-12 is only valid for RTR Release 21.17 and later. The 108, 109, and 110 CMDs will not appear on this Index Page until the RTR process that is responsible for the page is running.

Commands may also be input on the normal system pages. These commands generally consist of three- or four-digit sequences followed by a carriage return. Valid inputs are acknowledged with OK, and invalid inputs are acknowledged with NG. If additional processing occurs as the result of the menu input [for example, if the Display Administration Process (DAP) must fork and execute a process], valid inputs are acknowledged by OK IP.

How to Specify the Initial Display Pages for C/D Terminal in the ECD

In the *UNIX* RTR Operating System Release 1, a capability is provided to allow applications to specify in their ECD *getty* forms the page to be displayed automatically on each C/D device following a terminal restore (craft initialization, system boot, remove/restore, and so forth). The process that initializes a page specified in the ECD must be connected to a port. Therefore, not every page in the system can be displayed automatically following a terminal restore.

The following list contains the page names and port numbers of the *UNIX* RTR operating system pages that may be specified in an ECD *getty* form:

Page Name	Port Number
CPDP	17
INDEX	17
CDUP	17

For applications not desiring a specific display upon a terminal restore, the Common Processor Display Page (CPDP) will be the default page for all MTTY terminals except the SCC terminal. The default page for the SCC terminal is the Status Summary Area (SSA) page.

Once another system page is requested and displayed, pressing the NORM DISP function key causes the last page requested to be displayed again.

Use the following procedure to specify the initial page for a C/D device:

Step	Procedure
1	Invoke Recent Change and Verify (RC/V) by entering 199 <CR> at the CMD< prompt.
2	Fill in the necessary information to initiate an update of the <i>getty</i> form for the desired C/D device [on a CFTSHL (craft shell) – controlled terminal] or TTY device [on a DLGSHL (dialogue shell) – controlled terminal].
	 NOTE: The procedures for updating database forms are explained in 254-303-104, <i>Recent Change and Verify Manual</i> .
3	When the <i>getty</i> form is displayed, enter the name of the initial page desired in the <i>pagename</i> field and the port number of the process that initializes the page in the <i>portid</i> field.
4	After updating the <i>getty</i> form, remove and restore the device for which the <i>getty</i> form was updated.
5	To change message mode while an <i>rcv</i> session is in progress, press the CMD/MSG key. This will bring the cursor to the input text messages area, then press the CMD/MSG twice, and you are back into the <i>rcv</i> session already in progress.

How to Access the Status Summary Area Page

If multiple lines of output from an input message are expected, the SSA Page should be requested. This page has a larger message scrolling area than the other display pages and allows the user to see more of the output messages. In addition to the scrolling area, the system critical indicators are shown at the top of the display.

Use the following procedure to access the SSA Page on the MTTY terminal:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 101<CR> to obtain the SSA Page display. The display will be similar to the display shown in Figure 12-13.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< _				ENTER 100 FOR PAGE INDEX			

Figure 12-13. Status Summary Area Page

How to Access the Common Processor Display Page

To access the CPDP on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.  NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 102<CR> to obtain the CPDP display. It will be similar to one of the displays shown in Figures 12-14 through 12-21.

Page Description

The CPDPs (Figures 12-14 through 12-21) display the configuration of 3B20D/3B21D computers hardware units and provide menu commands to manipulate the hardware configuration.

To illustrate a more generic CPDP, the fan alarm units are intentionally not shown in Figures 12-14 through 12-18. The fan alarm indicators are available for RTR Release 21.13 and later only. Figures 12-19 and 12-20 show the CU FAN alarm units on the CPDP for the 3B20D and 3B21D computers. Figure 12-21 shows the 3B21D computer CPDP with the ABD feature.

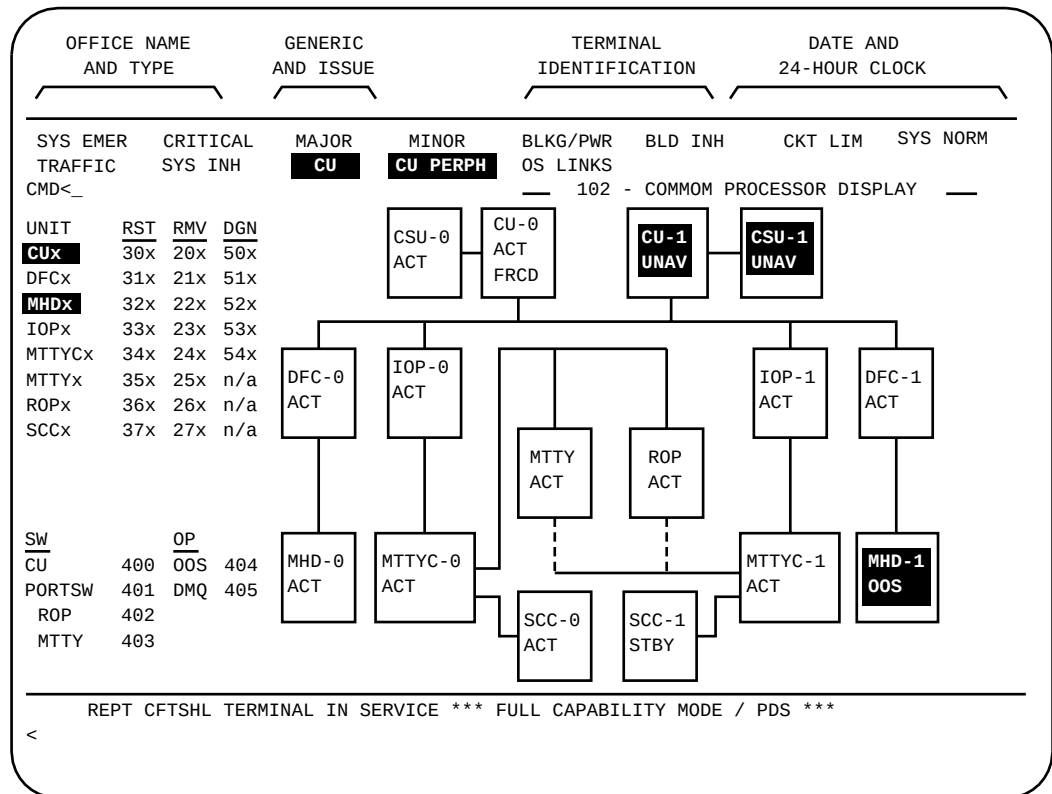


Figure 12-14. Common Processor Display Page

NOTE:

Although CSU 0 and CSU 1 are appropriately linked to the CU 0 and CU 1 units, respectively, the ACT CU may or may not run with the Cache Storage Unit (CSU).

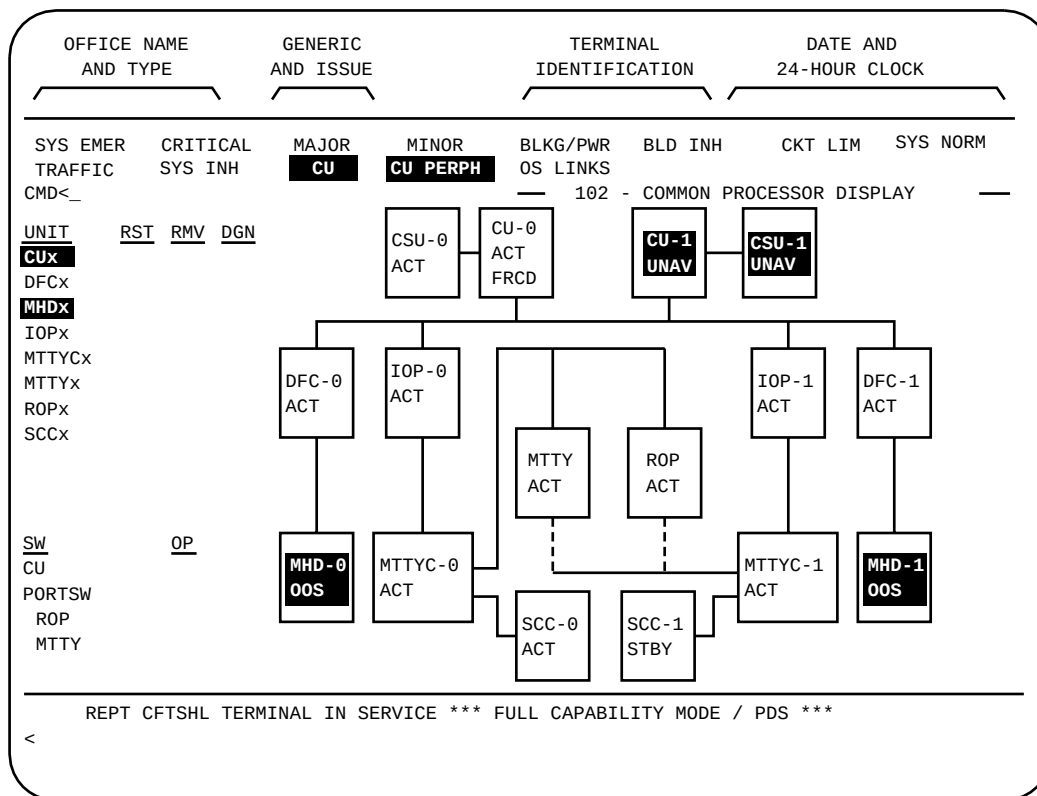


Figure 12-15. Common Processor Display Page in DIOP

NOTE:

During full Disk Independent Operation (DIOP), all the menu options on the Common Processor Display Page are disabled. Because of this, the menu options on the screen on Page 102 are cleared. The CPDP in DIOP appears as shown in Figure 12-15.

Figure 12-16 indicates the various states of the MTTY and ROP on the CPDP.

- The solid line from the MTTY and ROP to MTTYC 0 indicates an active connection.
- The disjointed line from the MTTY to MTTYC 1 indicates a standby connection.
- The highlighted broken line (red on the CPDP) from the ROP to MTTYC 1 indicates an OOS connection. Moreover, the menu line ROPx (which is highlighted in red on the page) indicates the OOS state of the device. The CU PERPH (CU peripheral) alarm also starts flashing.

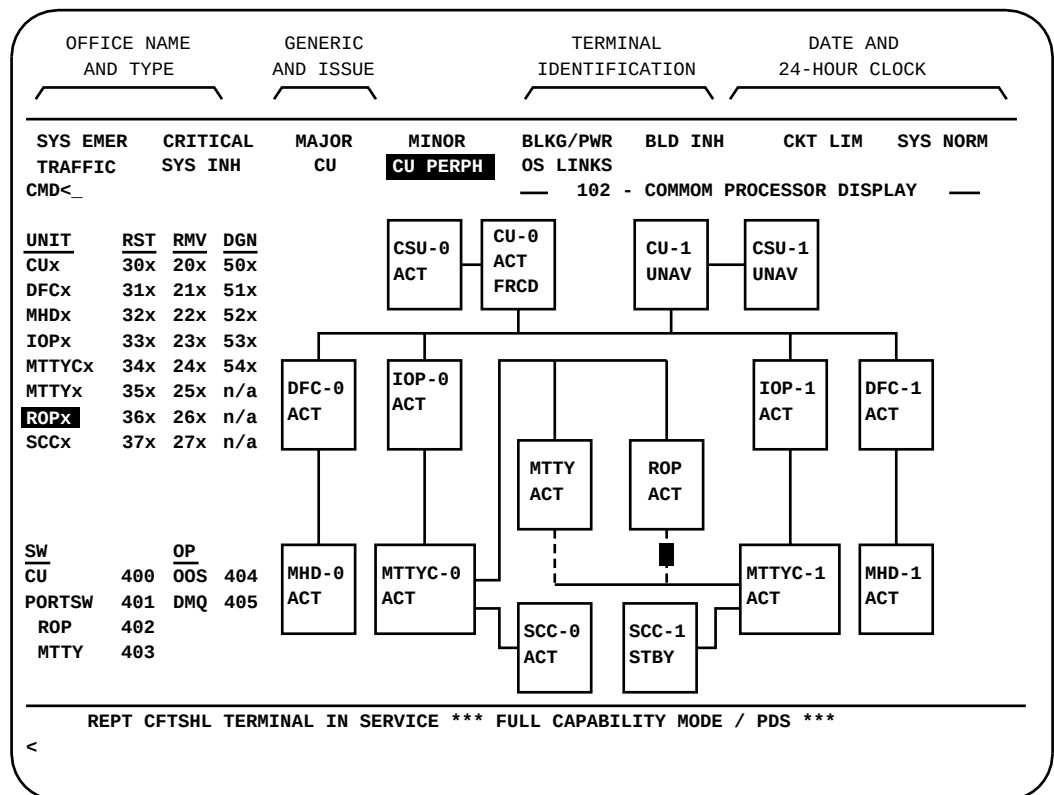


Figure 12-16. Common Processor Display Page — MTTY and ROP States

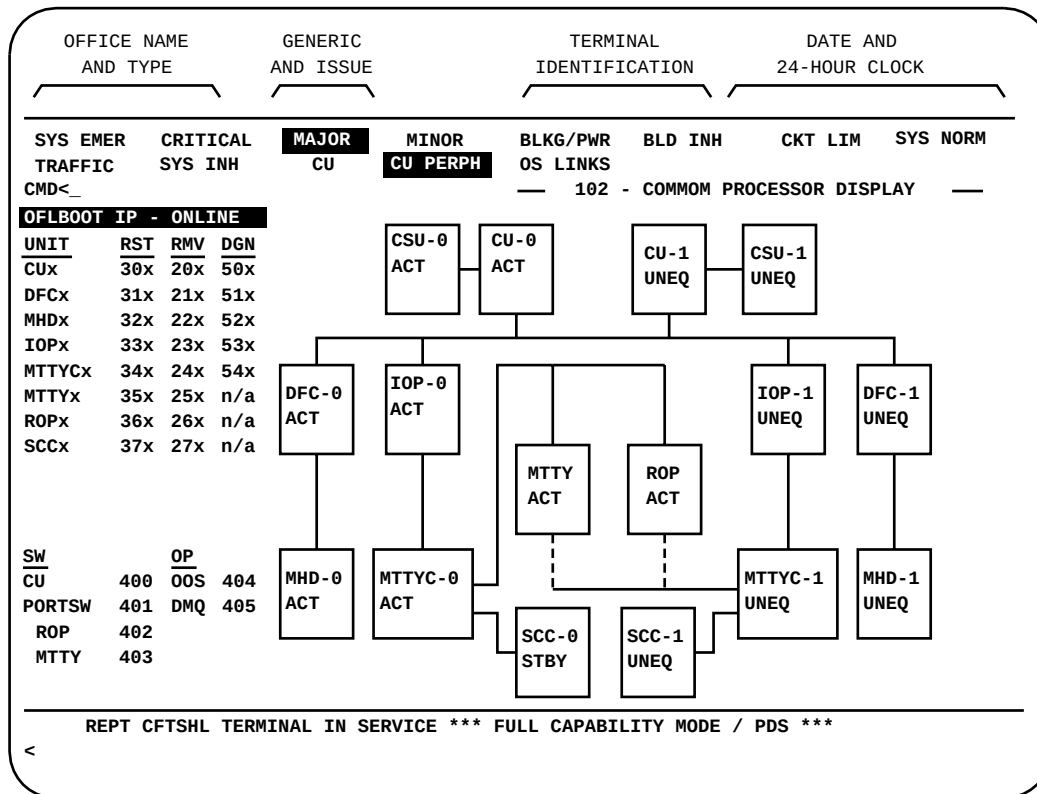


Figure 12-17. Common Processor Display Page — On-Line

⇒ **NOTE:**

When the off-line boot is in progress and the side is on-line, the indicator OFLBOOT IP-ONLINE will be displayed. This indicator will clear when the off-line boot stops.

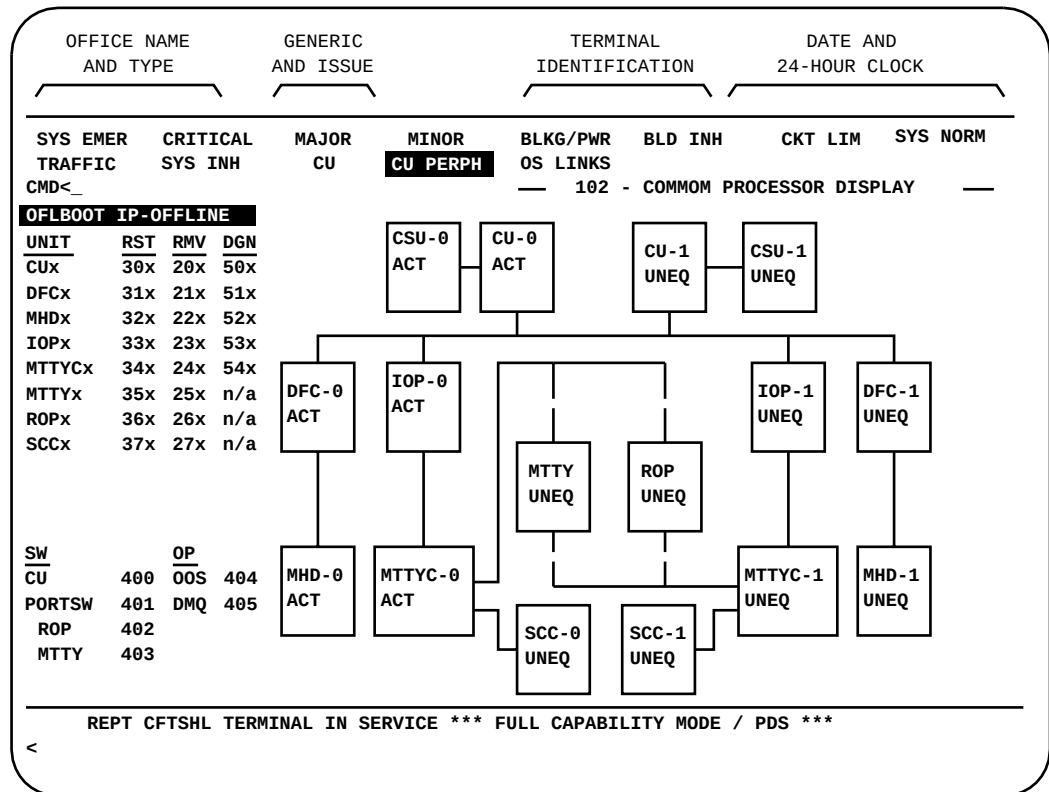


Figure 12-18. Common Processor Display Page — Off-Line

⇒ **NOTE:**

When the off-line boot is in progress and the side is off-line, the indicator OFLB00T IP - OFFLINE will be displayed. This indicator will clear when the off-line boot stops.

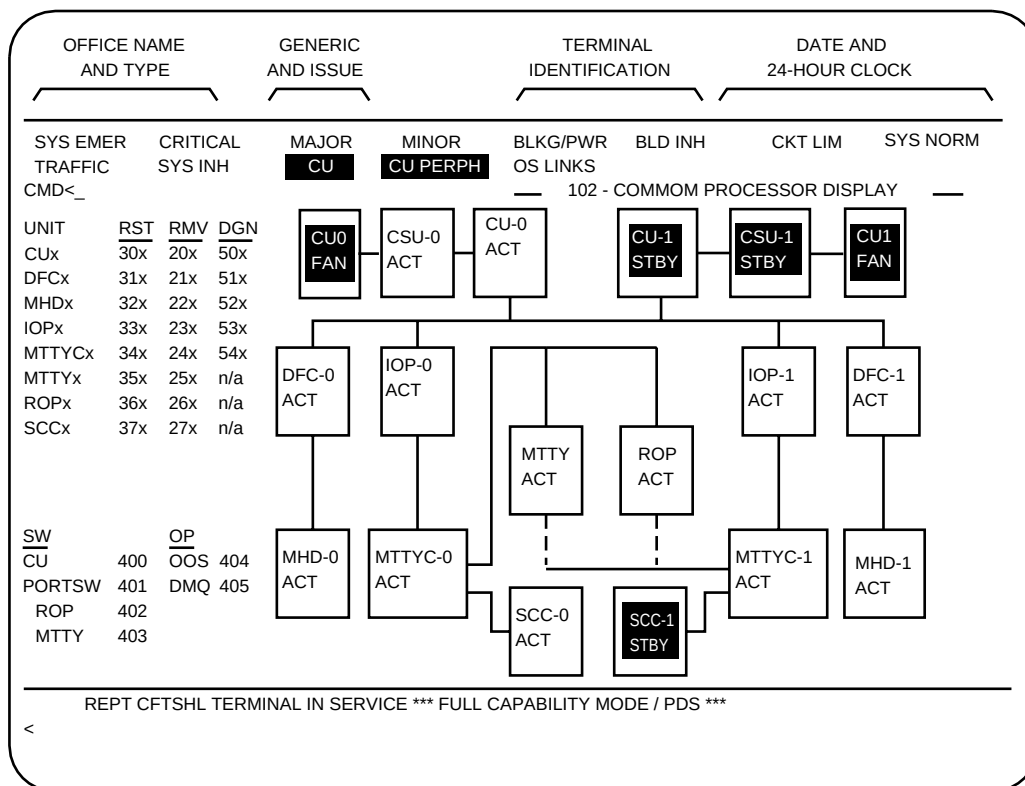


Figure 12-19. 3B20D Computer Common Processor Display Page with CU FAN Alarms

For RTR Release 21.13 and later, the 3B20D computer CU fans now have visual indicators associated with each CU. If a CU fan failure occurs, the corresponding CU FAN indicator will be highlighted.

Also, the text "INH" will appear in the CU FAN indicator(s) when the CU fan scan point(s) have been inhibited. If the CU FAN indicators are not desired or for more information, refer to Chapter 16, "SC Status and SD Point Operation."

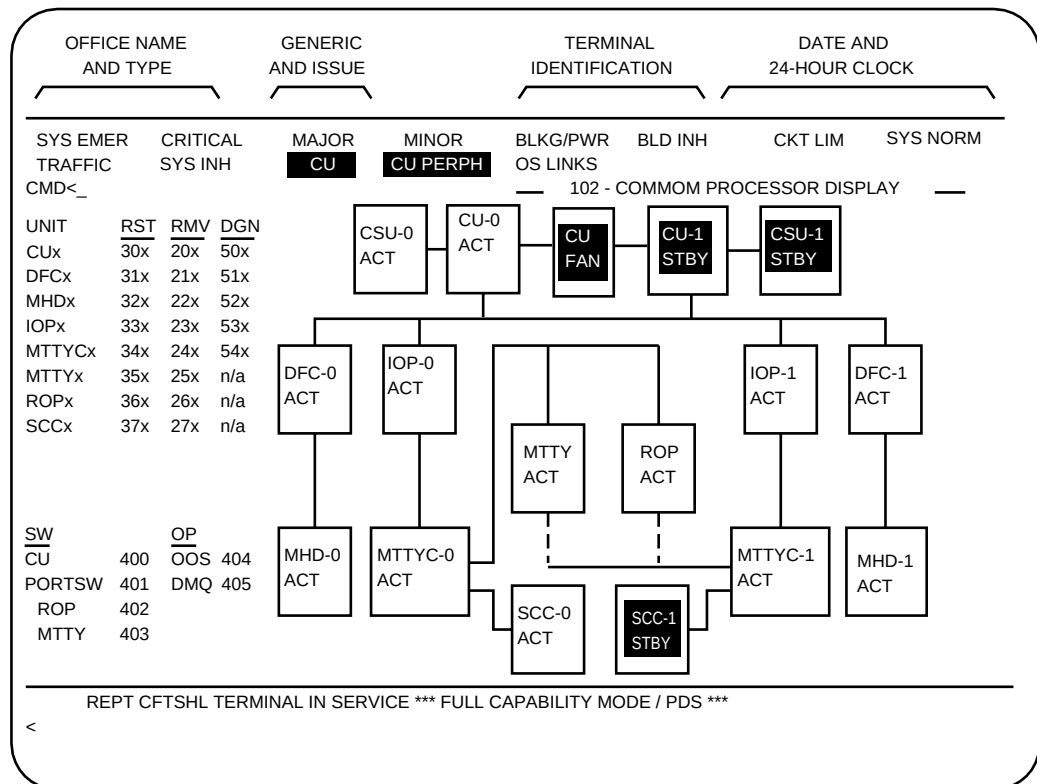


Figure 12-20. 3B21D Computer Common Processor Display Page with CU FAN Alarm

For RTR Release 21.13 and later, the 3B21D computer CU fan now has a visual indicator associated with both CUs. If a CU fan failure occurs, the CU FAN indicator will be highlighted.

Also, the text "INH" will appear in the CU FAN indicator when the CU fan scan point has been inhibited. If the CU FAN indicators are not desired or for more information, refer to Chapter 16, "SC Status and SD Point Operation."

Table 12-11 lists the CPDP commands without the ABD feature.

Table 12-11. Common Processor Display Page Commands Without ABD Feature

Menu Item*	Equivalent PDS Input Command*
30x	RST:CU x!
20x	RMV:CU x!
50x	DGN:CU x!
31x	RST:DFC x!
21x	RMV:DFC x!
51x	DGN:DFC x!
32x	RST:MHD x!
22x	RMV:MHD x!
52x	DGN:MHD x!
33x	RST:IOP x!
23x	RMV:IOP x!
53x	DGN:IOP x!
34x	RST:MTTYC x!
24x	RMV:MTTYC x!
54x	DGN:MTTYC x!
35x	RST:MTTY x!
25x	RMV:MTTY x!
36x	RST:ROP x!
26x	RMV:ROP x!
37x	RST:SCC x!
27x	RMV:SCC x!
400	SW:CU!
401	SW:PORTSW!
402	SW:PORTSW:ROP!
403	SW:PORTSW:MTTY!
404	OP:OOS!
405	OP:DMQ!

* x is equal to either 0 or 1.

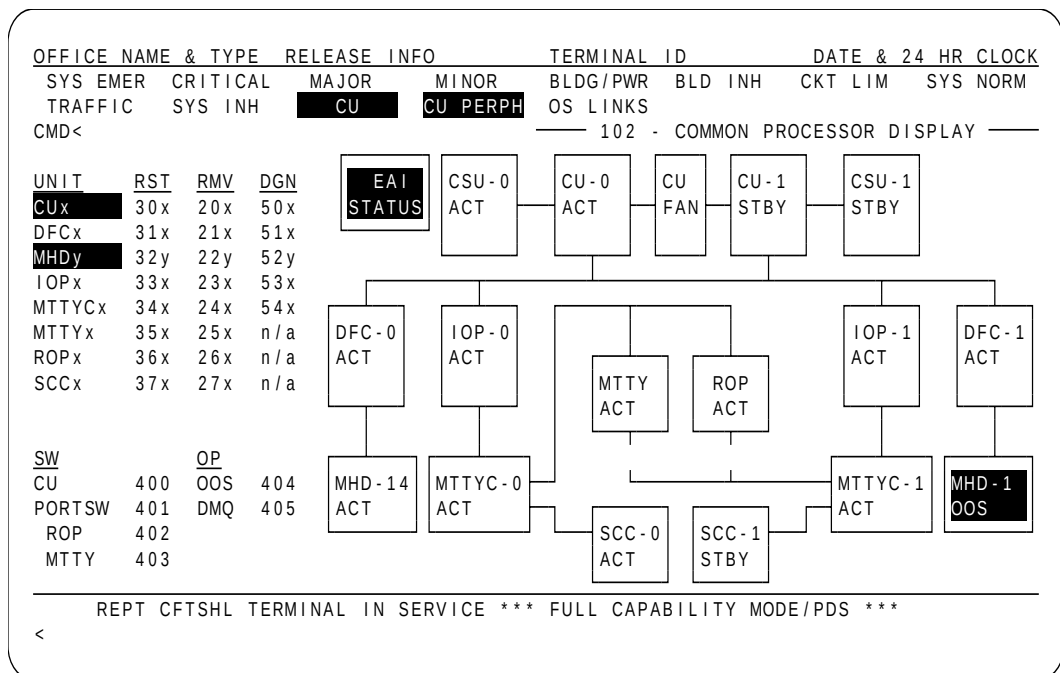


Figure 12-21. 3B21D Computer Common Processor Display Page with ABD Feature

For RTR Release 21.27 and later, the 3B21D computer's CPDP now has a visual indication that the ABD feature is available. If there is an indicator on the EAI Page that is considered off normal, the EAI STATUS indicator will be highlighted.

The MHD boxes contain the boot drive numbers which now can be numbers other than 0 or 1. To determine what is off normal, check the EAI Page.

Table 12-12 lists the CPDP commands with the ABD feature.

Table 12-12. Common Processor Display Page Commands with ABD Feature

Menu Item*	Equivalent PDS Input Command*
30x	RST:CU x!
20x	RMV:CU x!
50x	DGN:CU x!
31x	RST:DFC x!
21x	RMV:DFC x!
51x	DGN:DFC x!
32y	RST:MHD y!
22y	RMV:MHD y!
52y	DGN:MHD y!
33x	RST:IOP x!
23x	RMV:IOP x!
53x	DGN:IOP x!
34x	RST:MTTYC x!
24x	RMV:MTTYC x!
54x	DGN:MTTYC x!
35x	RST:MTTY x!
25x	RMV:MTTY x!
36x	RST:ROP x!
26x	RMV:ROP x!
37x	RST:SCC x!
27x	RMV:SCC x!
400	SW:CU!
401	SW:PORTSW!
402	SW:PORTSW:ROP!
403	SW:PORTSW:MTTY!
404	OP:OOS!
405	OP:DMQ!

* x is equal to either 0 or 1.

y is equal to MHD number under DFC 0 or DFC 1.

Maintenance procedures (removes, restores, and diagnostics) can be performed through the use of menu numbers. Menu numbers are also known as poke commands. (See Table 12-11 or Table 12-12 and Figure 12-14.) Beginning with *UNIX* RTR Operating System Release 1 and later, these commands will be executed conditionally unless options are specified by the user on the same command line. For each of the options on the commands under the **RST**, **RMV**, and **DGN** columns, the system allows valid options on the corresponding input messages to be entered following the command number and a comma (,).

For example, to restore an out-of-service computer control unit unconditionally, enter the following command at the `CMD<` prompt:

CMD< 300, UCL!

Where **300** is the menu number entered to restore CU-0 and Unconditional (UCL) is the optional text entered to restore the specified unit unconditionally. The commands under the **RST** column will restore units conditionally (which is the default) if the UCL option is not entered.

As another example, to diagnose Disk File Controller (DFC) 1 and invoke the trouble-locating procedure, enter the following command at the `CMD<` prompt:

CMD< 511, TLP!

The total length of the menu number and optional text must not exceed 30 characters.

Major and Minor State Display Field Values

Information such as the major state and the minor state (if available) are also provided for units on the CPDP. The CPDP is composed of a set of indicators that have the following attributes.

- Major State:

ACT	Active
GROW	In growth
INIT	Initialization in progress
OFL	Off-line
OOS	Out of service
STBY	Standby
UNAV	Unavailable
UNEQ	Unequipped.

- Optional Minor State:

DGN	Diagnostic in progress
ERR	Removed because of errors exceeding threshold
FALT	Removed because of diagnostics failure
MAN	Removed because of manual request
RST	Restore in progress.

- Other Information Displayed in the Minor State Field:

BYP	Inhibit cache
FDIO	Device that caused the system to go into DIOP
FRCD	Forced on-line
INH	Error interrupts inhibited.

How to Access the Control/Display Update Page

The menu commands on the C/D Update Page, shown in Figure 12-22, allow update of the internal craft interface tables without initializing all craft interface related processes via menu 15 on the Emergency Action Page. These commands, normally used when a field update is applied, should be used when:

- New system pages are installed.
- Video states change (color combination, blinking, and so forth).
- Craft daemon-process configurations change.
- System pages change, except Operating System Status Page, Page 104.
- A new Acknowledgment Database (ACKDB) is installed.
- A new Input Message Catalogue (IMCTLG) is installed.
- The user must wait 2 minutes between each consecutive poke on the 103 C/D Update Page. An “RL” (Retry Later) acknowledgment will be issued if a poke is entered before the two minute period expires.



CAUTION:

Use caution when entering input from this page. Misuse may lead to user terminal lockout. Each menu input must complete before the next menu input is entered.

The following procedure should be used to access the page on the MTTY terminal:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 103<CR> .

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				103 - C/D UPDATE			
CMD	FUNCTION			CMD	-	FUNCTION	
800	- UPDATE C/D GLOBAL MENU			806	-	RESTART CIA	
801	- UPDATE C/D STATE TRANSLATION			807	-	RESTART POKER(S)	
802	- RESTART ULARP			808	-	RESTART CFTSHL(S)	
803	- RESTART CSOP						
804	- RESTART RTS			810	-	SCC COLOR TERMINAL	
805	- RESTART DAP			811	-	SCC B/W TERMINAL	
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							

Figure 12-22. Control/Display Update Page

⇒ NOTE:

Menu commands 810 and 811 are executable only on remote maintenance consoles for SCC links.

How to Access the Operating System Status Page

To access the Operating System Status Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 104<CR> .

The menu commands on the Operating System Status Page, shown in Figure 12-23, are normally used to access resource usage information or to stop the display. These menu commands automatically return the screen to the normal FREEZE position when the screen has not been refreshed after various intervals.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD< 505 OK				— 104 - OPERATING SYSTEM STATUS PAGE —			
DISPLAY RATE		RESOURCE		PERCENTAGE OF RESOURCE USED			
500 -	FREEZE	QTY	TYPE	0	10	20	30
501 -	1 SECOND			40	50	60	70
505 -	5 SECONDS	17	MSG	80	90	100	
530 -	30 SECONDS			++++	++++	++++	++++
		50	DCT	=====	=====	=====	=====
SPN	00000000	387	SDT	=====	=====	=====	=====
SWAPI	0	2607	PDT	=====	=====	=====	=====
SWAP0	0	0	PDT1	=====	=====	=====	=====
		387	PGT	=====	=====	=====	=====
STIME	19			=====	=====	=====	=====
KTIME	81	4832	SSZ	=====	=====	=====	=====
KPTIME	2			=====	=====	=====	=====
		4625	SDK	=====	=====	=====	=====
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-23. Operating System Status Page

How to Interpret the Operating System Status Page

The display shown in Figure 12-23 represents the Operating System Status Page with a request for data every 5 seconds. In the upper left portion of the display is a box labeled `DISPLAY RATE`. This box represents the options available to the terminal user. Each subfield in this box represents a selectable display refresh rate.

500 - FREEZE	This initial display rate indicates that the screen is not currently being refreshed. No overhead is incurred when in this state. All other display rates will automatically switch to this state after various intervals.
501 - 1 SECOND	If selected, this option refreshes the display once each second and runs a maximum of 20 seconds. It is the most real-time display rate offered, but it incurs the greatest short-term overhead. This causes the short refresh period.
505 - 5 SECONDS	The display refreshes once every 5 seconds and runs a maximum of 10 minutes.
530 - 30 SECONDS	The display refreshes once every 30 seconds and runs a maximum of 2 hours.

To select a display refresh rate, enter the appropriate menu number and a carriage return. The selected option switches to reverse video display, and the various fields contained on the page reflect legitimate data. All data fields continue to be refreshed until the terminal user selects another option or until a fixed number of display refresh cycles have occurred (20 at the one-second rate, 120 at the other rates) at which time the `FREEZE` option is automatically invoked.

The Operating System Status Page, shown in Figure 12-23, displays the following fields:

SPN	The hexadecimal process number of the last supervisor process dispatched.
SWAPI	The number of segments swapped in since the last refresh of the display. Neither this nor the next field shows activity unless the page or page table utilization goes to 100 percent.
SWAPO	The number of segments swapped out since the last refresh of the display.
STIME	The number of 10-millisecond timer intervals accrued to supervisor processes since the last refresh of the display. This value added with the next two fields should be approximately equal to 100 times the selected display rate.

KTIME	The number of 10-millisecond timer intervals accrued to kernel and special process activity since the last refresh of the display.
KPTIME	The number of 10-millisecond timer intervals accrued to all kernel processes since the last refresh of the display.

The following fields represent system resources and the total amount of that resource being used. A bar graph indicates what percentage of the total available resource that amount in use represents:

MSG	Message buffers in use.
DCT	The current number of processes active in the system.
SDT	Segments in use. Every process must have a minimum of three segments. This value should always numerically exceed the active process count by at least threefold.
PDT0	Physical pages in use. Swapping will occur when all physical pages are in use.
PDT1	Physical pages in use in memory module 1. These pages are never swapped out and are freed only when the process which owns the pages terminates. This field is only displayed on computers equipped with the Extended Main Memory (EMM) feature. For non-EMM equipped computers, the value will always be 0.
PGT	Page tables in use. As long as the number of segments in use is less than the maximum number of page tables (<i>sgen</i> parameter), both counts should be equal. As soon as there are more segments than page tables, the counts will diverge and swapping will commence.
SSZ	Swappable memory utilization. At boot time, a calculation is made of the maximum available swappable memory. Later, as segments are locked or made nonswappable, the amount of swappable memory will decrease. This display represents the amount of initially swappable memory that is no longer available for swapping with 100 percent utilization representing the <i>sgen</i> specified minimum swap size.
SDK	Disk swap blocks in use. As segments are created, disk swap space is assigned to correspond to the segments.

If the craft display terminal has color supported then the operating system status display page bar graphs will also be in color with three distinct color levels:

NORMAL	All resources being utilized well within normal limits are displayed in green.
WARNING	Heavily used but not overloaded resources will be displayed in magenta. In particular, the usage of physical pages and page tables (PDT and PGT) will only go to the warning level at 100 percent utilization as a visual highlight that swapping is occurring.
DANGER	Any resource approaching overload will be displayed in red. The danger thresholds will occur at a slightly lower level of usage than the System Integrity Monitor (SIM) overload thresholds.

How to Access the Craft Field Maintenance Command Display Pages

The Craft Field Maintenance Commands (CFTMTCs) [Program Documentation Standard (PDS) or Man-Machine Language (MML)] to be displayed on the MTTY terminal are divided into two pages: 105 CRAFT FM 01 and 106 CRAFT FM 01 (Figures 12-24 and 12-26, respectively).

To access a desired CFTMTC display page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 105<CR> or 106<CR> .

How to Execute CFTMTC Commands via Menu Numbers

Each command is assigned a unique menu number. Once the desired page is displayed, each command is executed by entering the number adjacent to the command name on the CMD< prompt in the upper left portion of the display. After entering the menu number, a second page will be displayed which contains the parameters associated with that command and the menu number will be displayed in reverse video.

Figure 12-25 shows a page which contains required and optional fields. For example, on Page 105, when 912 is entered, the second page will contain SRC=, OPL=, BKG=, ARGS=, and EXEC. A message will be displayed on the lower part of the screen explaining which parameters are required and which parameters are optional. To enter data, use the "arrow" function keys found to the left of the key pad on the MTTY terminal. Position the cursor at the appropriate field, enter the data, and follow it with a <CR>. If the <CR> is not entered, the data will not be sent to the CFTMTC commands process. After the <CR>, the cursor will be repositioned to the CMD< prompt. Repeat these instructions to enter data for the next desired field. After entering the required data (for example, SRC is a required field), any of the optional parameters may be entered or omitted. After entering all of the appropriate data, use the "arrow" keys to position the cursor at the EXEC field. The cursor is at the EXEC field when a ">" character is placed at the beginning of the field (that is, >EXEC). Entering a <CR> at this point will execute the command using the craft shell.

For menu input commands not requiring parameters, a second page will not be created. Those commands are menu numbers 965, 966, and 971 on Page 106 and menu number 999 (ABORT) on both pages.

OK IP is displayed next to the menu number entered at the CMD< prompt, and the command is executed. If a menu number is not recognized, an NG is displayed.

The **ABORT** command is provided to allow users to erase a parameter page if one has been created and the user decides not to execute that command. However, **ABORT** cannot stop a command from executing after EXEC and <CR> have been entered.

Help messages are provided by entering a question mark (?) in the fields provided for entering parameters or by entering <CR>. Additional information, if needed, may be obtained by referring to the IO message manuals. The menu commands for executing **PDS/MML** commands on Pages 105 and 106 are shown in Figures 12-24 and 12-26.

Parameters and options associated with the CFTMTC commands are explained in Table 12-13. Commands, parameters, and options associated with the menu numbers on Pages 105 and 106 are shown in Tables 12-14 and 12-15, respectively.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<				105 -		CRAFT FM 01	
<u>CMR DISK</u>		<u>EXC EMVIR</u>		<u>FILESYS</u>		<u>IN FILE</u>	
901 CORE		911 PROC		921 ACCESS		932 DELT	
		912 UPROC		922 INDIR			
				923 INH			
<u>DUMP FILE</u>		<u>STOP EXC</u>		924 OWNER		999 ABORT	
902 ALL		913 ANY		925 RDIR			
903 DATA		914 USER		926 RFILE			
904 PARTL							
REPT.CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-24. CFTMTC Display Page 105

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				_____ 105 CRAFT FM 01 _____			
<u>CMPR DISK</u>		<u>EXC ENVIR</u>		<u>FILESYS</u>		<u>IN FILE</u>	
901 CORE		911 PROC		921 ACCESS		932 DEL	
<u>DUMP FILE</u>		912 UPROC		922 INDIR			
902 ALL		<u>STOP EXC</u>		923 INH			
903 DATA		913 ANY		924 OWNER		999 ABORT	
904 PARTL		914 USER		925 RDIR			
SRC= /bin/banner				926 RFILE			
OPL=							
BKG=							
ARGS= "HELLO"							
> EXEC							
912:_(REQUIRED): SRC, EXEC; (OPTIONAL): ARGS, BKG, OPL							

Figure 12-25. CFTMTC Display Page 105 with Menu 912 Data

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				106		CRAFT FM 01	
<u>COPY</u>		<u>COPY TAPE</u>		<u>OP STATUS</u>		<u>CLK</u>	
941 CFILE		951 IN		961 FREEDISK		971 OP	
942 FILE		952 LISTFILE		962 LISTDIR		972 SET	
943 PIN		953 OUT		963 PROCESS			
		954 TEST		964 SUM		<u>DISKUSE</u>	
<u>FILESYS</u>		955 EMERDMP		965 FILESYS		973 OP STATUS	
944 MOUNT				966 PORTS		999 ABORT	
REPT.CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-26. CFTMTC Display Page 106

Table 12-13. Parameters and Options for CFTMTC Commands

Fields	CFTMTC Command Meaning
1. FN =	Filename.
2. SRC =	Pathname of the source file to be copied by a COPY command.
3. DEST =	Pathname of the destination file in a COPY command.
4. OPL =	Stands for "output segment limit" specified in decimal. The default = 10 segments; maximum = 999. Each segment has a maximum size of 1000 bytes or 20 lines per segment.
5. FORMAT =	Specifies the output format of dump:file:data (default = octal). The formats are as follows: C = Interprets bytes in ASCII notation. D = Interprets words in decimal. O = Interprets words in octal. X = Interprets words in hexadecimal.
6. TOC =*	Table of Contents only.
7. PID =	Process identification of the process to be terminated.
8. SIG =	When terminating a process with a signal, SIG specifies which signal to use.

* Only "y" or "n" needs to be entered.

Table 12-13. Parameters and Options for CFTMTC Commands (Contd)

Fields	CFTMTC Command Meaning
9. LINE =	The line to delete when editing a file.
10. RO =*	A file system is to be mounted read-only.
11. AUD =*	The internal File Manager (FMGR) mount audit will run on the file system prior to mounting, but not File-System Block (FSBLK) or File-System Link (FSLINK) audits. The file manager will audit the file system prior to mounting.
12. BKG =*	A process is to be run in the background.
13. ARGS =	List of arguments for the user process when executing exc:envir:uproc (912). The arguments should be enclosed in double quotes and separated by commas when listing more than one argument. If only one argument is given, it should also be enclosed in double quotes. The maximum number of arguments is six.
14. ALLTERMS =*	Report status of all processes except group leaders.
15. ALLKERNS =*	Report status of all kernel processes.
16. ALL =*	Same as ALLTERMS, plus process group leaders and processes not associated with a terminal.
17. PATH =*	PATH is an option that can be used with op:status:process (963). It is equivalent to ps -axp . It gives the terminal identification and the full pathname of the process. It also gives the parent process identification of a process.

* Only "y" or "n" needs to be entered.

Table 12-13. Parameters and Options for CFTMTC Commands (Contd)

Fields	CFTMTC Command Meaning
18. FLIST =	File list is a file containing a list of full pathnames or a list of relative pathnames if BSDIR is used.
19. BSDIR =	BSDIR (base directory) is the pathname of a directory other than the current directory where files are to be written from or copied to.
20. TOTAL =*	The total number of blocks for a pathname is printed.
21. EACH =*	An entry for each file is printed.
22. ID =	When changing the owner or group of a file, ID specifies the new owner(s).
23. MODE =	MODE allows the user to change the access permissions of a file. The permissions allow the owner, group, and others to read, write, and execute a file.
24. TD =	TD specifies the tape special device filename; for example, TD /dev/mt08.
25. PARMS =	PARMS is an option that allows users to enter the parameters when executing set:clk (972).
26. ERASE =*	When executing menu number (942), if “y” is entered, there will not be a copy of file 1 left. It is equivalent to the <i>UNIX</i> system mv command.

* Only “y” or “n” needs to be entered.

Table 12-14. Display Page 105 Commands, Parameters, and Options for Menu Numbers

Menu Numbers	Commands, Parameters, and Options
901	CMPR:DISK:CORE, FN "a"!
902	DUMP:FILE:ALL, FN "a" [,OPL b]!
903	DUMP:FILE:DATA, FN "a" [,b] [,OPL c]!
904	DUMP:FILE:PARTL, FN "a", LINE b!
911	EXC:ENVIR:PROC, FN "a"!
912	EXC:ENVIR:UPROC, FN "a" [,ARGS (b)] [,BKG] [,OPL c]!
913	STOP:EXC:ANY,{ PID a FN "b" }, UCL!
914	STOP:EXC:USER, { PID a FN "b" } [,SIG c]!
921	ALW:FILESYS:ACCESS "a", FN "b"!
922	IN:FILESYS:DIR, FN "a"!
923	INH:FILESYS:UMOUNT, FN "a"!
924	ALW:FILESYS:OWNER "a", FN "b"!
925	CLR:FILESYS:DIR, FN "a"!
926	CLR:FILESYS:FILE, FN "a"!
932	IN:FILE:DEL, FN "a", LINE b!
999	ABORT!

Table 12-15. Display Page 106 Commands, Parameters, and Options for Menu Numbers

Menu Numbers	Commands, Parameters, and Options
941	COPY:FILESYS:CFILE, FN "a"!
942	COPY:FILESYS:FILE, SCR "a", DEST "b", \[ERASE NOERASE]!
943	COPY:PTN:ALL, SCR "a", DEST "b"!
944	ALW:FILESYS:MOUNT, FN "a", BSDIR "b" [,RO] [,AUD]!
951	COPY:TAPE:IN, TD "a" [,TOC] [,BSDIR "b"]!
952	COPY:TAPE:OUT, TD "a" {,FN "b" ,FLIST "c"}!
953	COPY:TAPE:OUT, TD "a" {,FN "d" ,FLIST "e"} ,BSDIR "f"!
954	COPY:TAPE:TEST, TD "a"!
955	COPY:TAPE:EMERDMP, TD "a"!
961	OP:STATUS:FREEDISK [,FN "a"]!
962	OP:STATUS:LISTDIR, FN "a" [,OPL b]!
963	OP:STATUS:PROCESS, {ALLTERMS ALLKERNS ALL PATH}\[,OPL a]!
964	OP:STATUS:SUM, FN "a"!
965	OP:STATUS:FILESYS!
966	OP:STATUS:PORTS!
971	OP:CLK!
972	SET:CLK "abcde"!
973	OP:STATUS:DISKUSE, FN "a", { TOTAL EACH } [,OPL b]!
999	ABORT!

How to Access the System Update Menu Page

The System Update Menu Page (Figure 12-27) is composed of three screens that are displayed on the maintenance terminal to allow the user to execute system update and application commands with their appropriate arguments. The main screen is called the "System Update Menu Page." The other two screens are called "Enter Argument Page 1" and "Enter Argument Page 2." All actions are performed on the System Update Menu Page screen except specifying the enter command and line argument which are performed on the two enter argument screens. For more information, refer to Chapter 11, "System Update."

To access the System Update Menu Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.	
3	At the CMD< prompt, enter 107<CR> .

On each of the three screens, a prompt (**CMD<**) is located in the upper left corner of the screen. A blinking underscore cursor is placed after the prompt to designate where to type the desired menu numbers. On the right-hand side of the **CMD<** line, the title of each screen appears. Below the **CMD<** line, a "Messages Line" exists to report four types of messages: input errors made by the user, warning messages, instructions concerning system update procedures, and information indicating the status of the update. All messages that are displayed on the Messages Line are also written to the ROP. The last two lines of each screen are reserved for the scrolling line (used to display output) and the manual input line (used to executed PDS/MML input commands manually), respectively.

How to Access the DSCH Computer Interconnect (DCI) Status Page

The DCI Status Page (Figures 12-28 through 12-30) is currently composed of three possible screens displayed on the maintenance terminal to allow the user to monitor the hardware status of DCIs that are equipped. The DCI feature was implemented on RTR Release 21.3 and later. For more information, refer to Chapter 22, "DSCH Computer Interconnect Feature."

To access the DCI Status Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 108<CR> .

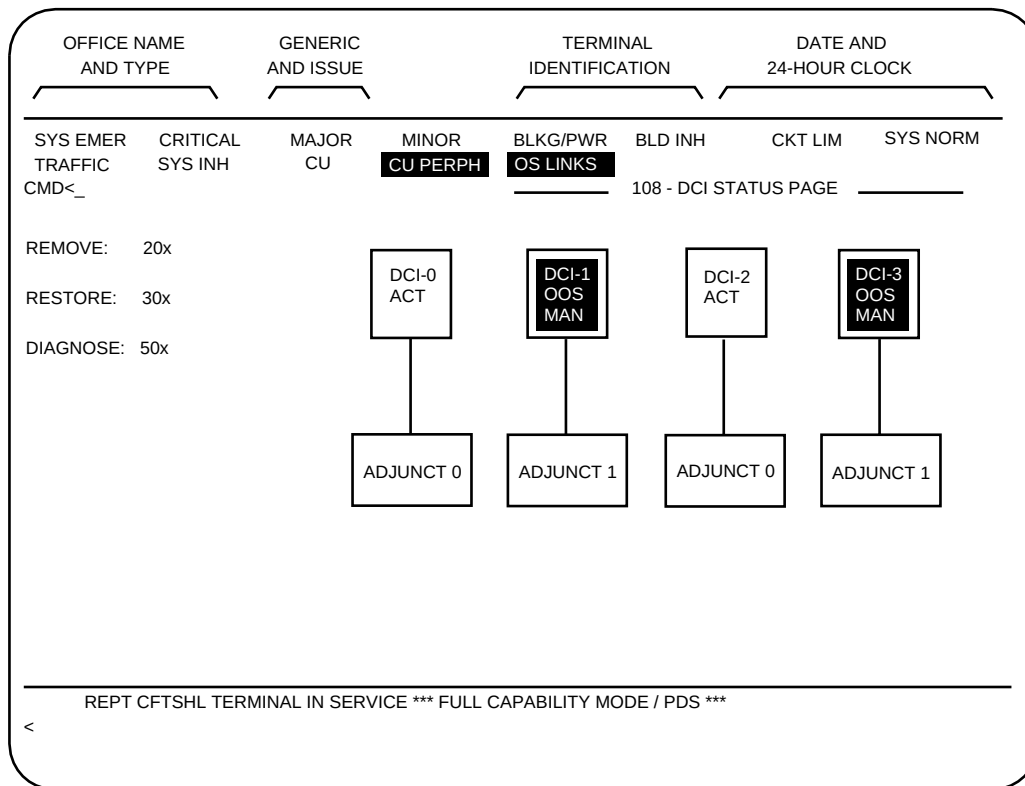


Figure 12-28. DCI Status Page

NOTE:

The DCI Status Page shown is defined for RTR Release 21.17 and later and displays the hardware status of the DCIs that are equipped. The DCI Status Page will not be accessible if no DCIs are equipped. If there is only one DCI configuration equipped, only one attached computer (ADJUNCT) will be displayed.

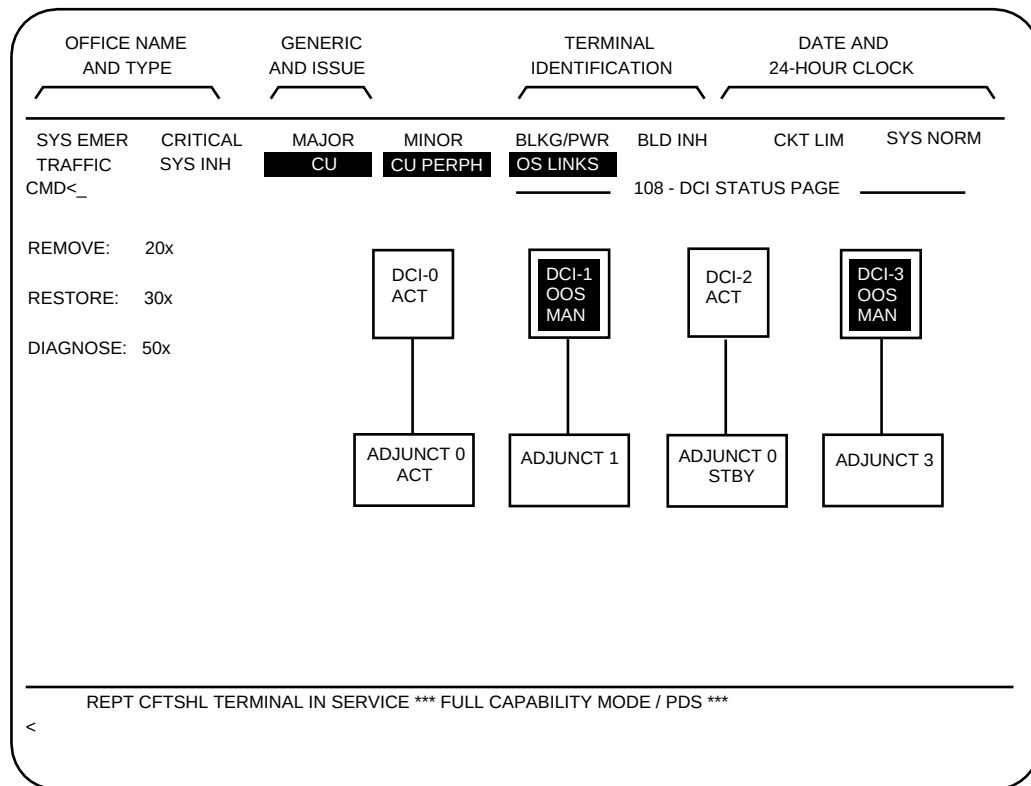


Figure 12-29. DCI Status Page for RCC

⇒ NOTE:

When two DCIs are connected to a Reliable Cluster Computer (RCC), the attached computer (ADJUNCT) will have a state similar to the DCI unit's state. The states for the ADJUNCTs are maintained only in an RCC configuration.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				108 - DCI STATUS PAGE			
REMOVE: 20x		DCI - 0 ACT ADJUNCT 0					
RESTORE: 30x							
DIAGNOSE: 50x							
<u>DSMLNK</u>							
REMOVE:: 210		TTY 14 ACT					
RESTORE: 310							
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-30. DCI Status Page with DSMLNK

⇒ NOTE:

The DSMLNK information will appear on this page if TTY was installed with the DSMLNK device type; otherwise, this indicator will not be visible. The DSMLNK TTY device is connected to a serial port on an ADJUNCT processor.

Table 12-16 lists the DCI Status Page commands.

Table 12-16. DCI Status Page Commands

Menu Item*#	Equivalent PDS Input Command *
20x	RMV:DCI x!
30x	RST:DCI x!
50x	DGN:DCI x!
210	RMV:TTY m!
310	RST:TTY m!

* x is equal to either 0, 1, 2, or 3.

m is the number of the TTY displayed on the page
210 and 310 only appear on DCI Status Page when DSMLMK is defined on the page.

Maintenance procedures (removes, restores, and diagnostics) can be performed through the use of menu numbers. Menu numbers are also known as poke commands. (See Table 12-16 and Figures 12-28 and 12-29.) Beginning with *UNIX RTR Operating System Release 21.17* and later, these commands will be executed conditionally unless options are specified by the user on the same command line. For each of the options on the commands under the **RST**, **RMV**, and **DGN** columns, the system allows valid options on the corresponding input messages to be entered following the command number and a comma (.).

For example, to restore an out-of-service DCI unconditionally, enter the following command at the **CMD<** prompt:

CMD< 300, UCL!

Where **300** is the menu number entered to restore DCI-0 and Unconditional (UCL) is the optional text entered to restore the specified unit unconditionally. The commands under the **RST** column will restore units conditionally (which is the default) if the UCL option is not entered.

The total length of the menu number and optional text must not exceed 30 characters.

Major and Minor State Display Field Values for DCI

Information such as the major state and the minor state (if available) are also provided for units on the DCI Status Page. The DCI Status Page is composed of a set of indicators that have the following attributes:

Major State:

ACT	Active
GROW	In growth
INIT	Initialization in progress
OFL	Off-line
OOS	Out of service
STBY	Standby
UNAV	Unavailable
UNEQ	Unequipped

How to Access the Field Update Page

To access the Field Update Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 109<CR> . This will cause BWMINT to be created and the Field Update Page to be displayed.

Page Description

Figure 12-31 is a sample of the Field Update Page which comprises the following areas:

- **Indicators:** The indicator portion of the page identifies the possible *BWMINT* requests and associated menu numbers. Each indicator is activated by entering its menu number at the `CMD<` prompt. Some indicators are text, and some are nontext. Text indicators require additional input upon activation; nontext indicators do not. Text indicators are followed by an equal sign identifying where the text is entered. There are 3 categories of indicators: configuration, scope, and action.

- The configuration indicators request global actions associated with an update session and describe the MSGS file to *BWMINT* as follows: (The MSGS file contains input messages to install the update package.)

BWM	Text	Allows input of BWM name for update session. This must be a directory in <i>/etc/bwm</i> that contains an MSGS file.
TERM	Nontext	Terminates current <i>BWMINT</i> update session.
DIR	Text	Allows input of alternate directory for the MSGS file.

- The scope indicators restrict the action of *BWMINT* to certain portions of the MSGS file as follows:

APPLY	Nontext	Makes the APPLY section the current working portion.
SOAK	Nontext	Makes the SOAK section the current working portion.
BKOUT	Nontext	Makes the BKOUT section the current working portion.
PERM	Nontext	Makes the PERM section the current working portion.
FILE	Nontext	Makes the entire MSGS file the current working portion.
LINE	Text	Allows input of the line number of a message to be executed on reset.

— The action indicators are as follows:

DISPLAY	Nontext	Displays the current working portion of the MSGS file.
NEXT	Nontext	Displays next page of the current working portion.
PREVIOUS	Nontext	Displays previous page of the current working portion.
PRINT	Nontext	Prints the current working portion to ROP.
RESET	Nontext	Executes a reset for a message that failed.
EXEC	Nontext	Executes messages in the current working portion.

- **Communication Line:** The communication line is reserved for output by *BWMINT*. The two types of output information are errors and status.

Error output information includes:

- Invalid combination of indicators
- Invalid menu number
- Cannot execute PERM before APPLY is successfully executed.

Status output information includes:

- Execution COMPLETE
- Execution ABORTED.

Output on the communication line will remain until the next menu number is input. All information output to the communication line will also be formatted and printed on the ROP.

- **MSGS Display:** The MSGS file display area will contain portions of the current section of the MSGS file for the current BWM. Any display of the MSGS file includes all comments and input messages in the portion displayed. The working area for the MSGS file to be displayed will consist of 12 lines for data. The displayed portion will contain as much of the requested section as the working area will allow. Input messages do not split across display portions. If an input message does not completely fit in the working area, it is reserved to be displayed with the remaining section. In order to display the remainder of that section, the user requests to go forward in the MSGS file. To display a portion of the section previously displayed, a request would be made to go backward and display that portion of the MSGS file. When an execution request occurs, if the portion of the MSGS file being executed is not currently displayed, the display will change to include that portion.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				FIELD UPDATE PAGE (109)			
400 BWM=CFT83-0101		403 APPLY	406 PERM		409 DISPLAY	412 PRINT	
401 TERM		404 SOAK	407 FILE		410 NEXT	413 RESET	
		405 BKOUT	408 LINE=		411 PREVIOUS	475 EXEC	
402 DIR=							
COMMUNICATION							
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-31. Field Update Page

Functional Description

The field update visual control display package will be a menu driven CMD< line procedure. *BWMINT* is invoked via menu input at the CMD< prompt of the MTTY. DAP receives the menu input and creates *BWMINT*. Once in service, *BWMINT* will establish a client relationship with DAP. When the Field Update Page menu number is specified at the CMD< prompt, DAP will bring up the field update page on the terminal. (Refer to Figure 12-31.) The methods to make requests to *BWMINT* are available as follows:

- Enter the menu number for a configuration indicator. If the selection is for a text indicator, position the cursor after the indicator name and enter a text string. Another way is to enter the menu number, a comma, and the text string on the CMD< line. In any case, the indicator name is placed in reverse video to indicate that the indicator is active and *BWMINT* registers the request. If an action is associated with the request, *BWMINT* performs the action.
- Enter the menu number of a scope state followed by the menu number for an action. If text input is required, position the cursor at the indicator to allow input. Another way is to enter the menu number, a comma, and the text string on the CMD< line. In any case, the indicator names are placed in reverse video. Any scope or action indicators in reverse video from previous requests are returned to normal state. The *BWMINT* performs the requested action with the requested scope.

Once a *BWMINT* request has been made via the menu, no further *BWMINT* requests will be processed until the current request is completed. For example, after requesting execution of line 4, any subsequent request will be processed after line 4 has completed execution.

The MSGS file display area will contain the portion of the MSGS file specified by the user via menu input for the scope indicators. All lines of the MSGS file will be displayed with input message lines consecutively numbered for use with the line indicator. Only the first 76 characters of each line in the MSGS file will be displayed. Successfully completed messages (including the message number) are displayed in reverse video, currently executing messages are blinking, messages which have failed are blinking, and unexecuted messages are in normal video state.

During an update session with the MSGS file, the user can escape to the scrolling area at any time for manual input. The escape mechanism will be via the **CMD MSG** key on the terminal. The user may also go to a different page and then return to the Field Update Page by reentering the Field Update Page number or by using the **DELETE** key on the terminal. *BWMINT* will not terminate in either of these cases. *BWMINT* will continue its requested operations while the user views other pages. Upon return, the Field Update Page will reflect its current state: executing, error, or completed.

Refer to Table 12-17 for the allowable scope/action combinations and the effects of each combination.

Table 12-17. Field Update Scope/Action Combinations

Scope	Action	Effect
APPLY	DISPLAY	The APPLY section or the first 12 lines of the APPLY section are displayed on the MTTY screen.
SOAK	DISPLAY	The SOAK section or the first 12 lines of the SOAK section are displayed on the MTTY screen.
BKOUT	DISPLAY	The BKOUT section or the first 12 lines of the BKOUT section are displayed on the MTTY screen.
PERM	DISPLAY	The PERM section or the first 12 lines of the PERM section are displayed on the MTTY screen.
FILE	DISPLAY	The entire MSGS file or the first 12 lines of the MSGS file are displayed on the MTTY screen.
APPLY	EXEC	The APPLY section messages are sequentially executed. If the entire APPLY section does not fit in the MSGS area, the portion of the APPLY section displayed will shift to include the messages being executed.
SOAK	EXEC	The SOAK section messages are sequentially executed. If the entire SOAK section does not fit in the MSGS area, the portion of the SOAK section displayed will shift to include the messages being executed.
BKOUT	EXEC	The BKOUT section messages are sequentially executed. If the entire BKOUT section does not fit in the MSGS file area, the portion of the BKOUT section displayed will shift to include the messages being executed. Executing the entire BKOUT section will cause the APPLY section to return to normal video (unexecuted).

Table 12-17. Field Update Scope/Action Combinations (Contd)

Scope	Action	Effect
PERM	EXEC	The PERM section messages are sequentially executed. If the entire PERM section does not fit in the MSGS file area, the portion of the PERM section displayed will shift to include the messages being executed. The PERM section can only be executed after the APPLY section has been completely executed.
LINE	EXEC	The message number input to the LINE indicator message is executed. If that line is not in the MSGS display area, the display will shift to include the message. Lines in a section must be executed sequentially. The third line cannot be executed before the first or second line in a section is executed.
LINE	RESET	A RESET operation is initiated for the incomplete APPLY or PERM message identified by the message number input to the LINE indicator. A reset can only be performed if the line is flashing to indicate that the message failed.
APPLY	PRINT	The APPLY section is printed on the ROP.
SOAK	PRINT	The SOAK section is printed on the ROP.
BKOUT	PRINT	The BKOUT section is printed on the ROP.
PERM	PRINT	The PERM section is printed on the ROP.
FILE	PRINT	The entire MSGS file is printed on the ROP.
APPLY	NEXT	If the remainder of the APPLY section is not in the MSGS display area following an APPLY DISPLAY or APPLY NEXT, the next portion of the APPLY section is displayed on the MTTY screen.
SOAK	NEXT	If the remainder of the SOAK section is not in the MSGS display area following a SOAK DISPLAY or SOAK NEXT, the next portion of the SOAK section is displayed on the MTTY screen.

Table 12-17. Field Update Scope/Action Combinations (Contd)

Scope	Action	Effect
BKOUT	NEXT	If the remainder of the BKOUT section is not in the MSGS display area following a BKOUT DISPLAY or BKOUT NEXT, the next portion of the BKOUT section is displayed on the MTTY screen.
PERM	NEXT	If the remainder of the PERM section is not in the MSGS display area following a PERM DISPLAY or PERM NEXT, the next portion of the PERM section is displayed on the MTTY screen.
FILE	NEXT	If the remainder of the MSGS file is not in the MSGS display area following a FILE DISPLAY or FILE NEXT, the next portion of the MSGS file is displayed on the MTTY screen.
APPLY	PREVIOUS	If the initial portion of the APPLY section is not in the MSGS display area following an APPLY NEXT or an APPLY PREVIOUS, the previous portion of the APPLY section is displayed on the MTTY screen.
SOAK	PREVIOUS	If the initial portion of the SOAK section is not in the MSGS display area following a SOAK NEXT or a SOAK PREVIOUS, the previous portion of the SOAK section is displayed on the MTTY screen.
BKOUT	PREVIOUS	If the initial portion of the BKOUT section is not in the MSGS display area following a BKOUT NEXT or a BKOUT PREVIOUS, the previous portion of the BKOUT section is displayed on the MTTY screen.
PERM	PREVIOUS	If the initial portion of the PERM section is not in the MSGS display area following a PERM NEXT or a PERM PREVIOUS, the previous portion of the PERM section is displayed on the MTTY screen.
FILE	PREVIOUS	If the initial portion of the MSGS file is not in the MSGS display area following a FILE NEXT or a FILE PREVIOUS, the previous portion of the file is displayed on the MTTY screen.

How to Access and Interpret the DFC Index Page

To access the DFC Index Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, enter 110<CR> .

The display area indicates the DFC menu pages that are available and the DFCs supported by each menu page. Figure 12-32 shows a typical DFC Index Page.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				110 - DFC INDEX PAGE			
<u>CMD</u>	<u>PAGE</u>	<u>TITLE</u>		<u>CMD</u>	<u>PAGE</u>	<u>TITLE</u>	
120	DFC	DFC0-1 STATUS		122	DFC	DFC2-3	STATUS
121	DFC	DFC0-1	PERFORMANCE	123	DFC	DFC2-3	PERFORMANCE
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-32. Typical DFC Index Page

How to Access the DFC Status Page

To access the DFC Status Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	Access the DFC Index Page. (Refer to the “How to Access and Interpret the DFC Index Page” section.)
4	From the DFC Index Page, determine the DFC Status Page menu number for the desired DFC. At the CMD< prompt, enter the menu number followed by a carriage return.

The number of DFC Status Pages and the number of DFCs displayed on the pages will vary based on the *UNIX* RTR operating system release and the hardware equipped. Figures 12-33 through 12-37 show examples of typical DFC Status Pages.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<				120 - DFC STATUS PAGE			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX			
DFC-0	ACT	Y		DFC-1	ACT	Y	
MHD-0	ACT	Y E		MHD-1	OOS	N E	
MHD-2	ACT	Y		MHD-3	OOS	N	
MHD-4	ACT	Y					
CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION
600	INIT MHD	603	DUMP DEFECT			677	SET MSG LVL
601	VFY MHD					688	CLEAR BUFF
602	RST DFC	605	RST MHD				
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

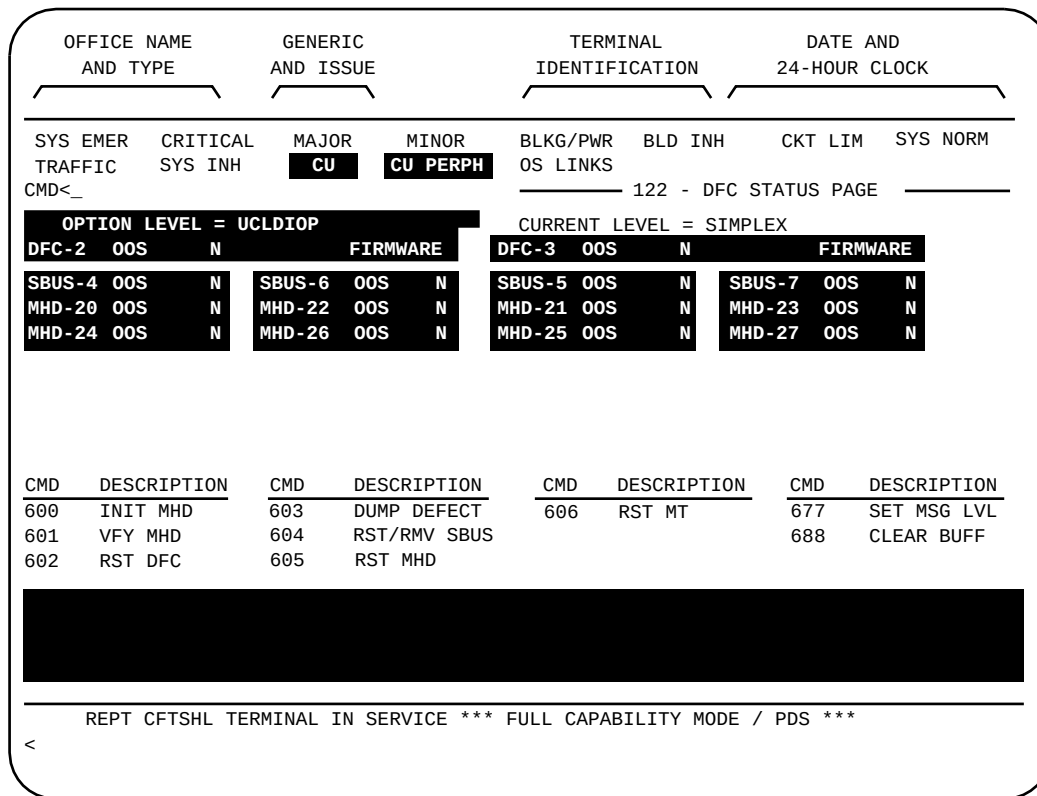
Figure 12-33. Typical DFC Status Page for a System Equipped with Storage Module Drive (SMD) DFCs 0 and 1

OFFICE NAME AND TYPE			GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM	
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS				
CMD<_				122 - DFC STATUS PAGE				
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX				
DFC-2	ACT	Y		DFC-3	ACT	Y		
MHD-16	ACT	Y E		MHD-17	OOS	N		
MHD-18	ACT	Y		MHD-19	OOS	N		
MHD-20	ACT	Y						
CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	
600	INIT MHD	603	DUMP DEFECT			677	SET MSG LVL	
601	VFY MHD					688	CLEAR BUFF	
602	RST DFC	605	RST MHD					
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <								

Figure 12-34. Typical DFC Status Page for a System Equipped with SMD DFCs 2 and 3

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				120 - DFC STATUS PAGE			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = DUPLEX			
DFC-0	ACT Y	PUMPCODE		DFC-1	ACT Y	PUMPCODE	
SBUS-0	ACT Y	SBUS-2	ACT Y	SBUS-1	ACT Y	SBUS-3	ACT Y
MHD-0	ACT Y E	MHD-2	ACT Y	MHD-1	ACT Y E	MHD-3	ACT Y
MHD-4	ACT Y	MHD-6	ACT Y	MHD-5	ACT Y	MHD-7	ACT Y
MHD-8	ACT Y	MHD-10	ACT Y	MHD-9	ACT Y	MHD-11	ACT Y
MHD-12	ACT Y	MHD-14	ACT Y	MHD-13	ACT Y	MHD-15	ACT Y
				MT-1	ACT Y		
CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION
600	INIT MHD	603	DUMP DEFECT	606	RST MT	677	SET MSG LVL
601	VFY MHD	604	RST/RMV SBUS			688	CLEAR BUFF
602	RST DFC	605	RST MHD				
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-35. Typical DFC Status Page for a System Equipped with SCSI DFCs 0 and 1



OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				120 - DFC STATUS PAGE			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = FULL DIOP			
DFC-0	ACT	Y	FIRMWARE	DFC-1	ACT	Y	FIRMWARE
SBUS-0	ACT	Y	SBUS-2	ACT	Y	SBUS-3	ACT
MHD-0	OOS	FDIO	Y E	MHD-1	OOS	N E	MHD-3
MHD-4	ACT	Y	MHD-6	ACT	Y	MHD-7	ACT
MHD-8	ACT	Y	MHD-10	ACT	Y	MHD-11	ACT
MHD-12	ACT	Y	MHD-14	ACT	Y	MHD-15	ACT
MT-1	ACT	Y					
CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION	CMD	DESCRIPTION
600	INIT MHD	603	DUMP DEFECT	606	RST MT	677	SET MSG LVL
601	VFY MHD	604	RST/RMV SBUS	607	DUMP BKTape	688	CLEAR BUFF
602	RST DFC	605	RST MHD	620	LOAD MHD	699	ABORT CMD IP
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-37. Typical DFC Status Page for a System, in Full DIOP, Equipped with SCSI DFCs 0 and 1

How to Interpret the DFC Status Page

The `OPTION LEVEL` display field indicates the DIOP feature level that is selected in the *ecdorg* form of the ECD. It will be one of the following:

<code>NODIOP</code>	DIOP capabilities are not provided.
<code>UCLDIOP</code>	DIOP capabilities are unconditionally provided.

The `CURRENT LEVEL` display field indicates the *UNIX* RTR operating system DIOP status. It will be one of the following:

<code>NORMAL</code>	DIOP is disabled.
<code>INITIALIZING</code>	The on-line CU is initializing. If <code>UCLDIOP</code> is selected in the ECD, <code>NODIOP</code> will be displayed as the <code>OPTION LEVEL</code> during the CU initialization interval. The DIOP feature will be enabled when the CU initialization completes.
<code>SIMPLEX</code>	DIOP is enabled and at least one essential Moving Head Disk (MHD) of a duplex pair is out of service.
<code>DUPLEX</code>	DIOP is enabled and all essential MHDs are active.
<code>CONDITIONAL DIOP</code>	DIOP is enabled and pending further errors, the last active essential MHD of a duplex pair is about to be removed from service.
<code>FULL DIOP</code>	DIOP is enabled and the last active essential MHD of a duplex pair has been removed from service.

⇒ NOTE:

In later *UNIX* RTR operating system releases, only essential MHDs which contain the `ROOT` and `SWAP` partitions affect the DIOP status displayed in the `CURRENT LEVEL` field.

The DFC Status Page displays the status of the equipped DFC, Small Computer System Interface Buses (SBUSs), and peripheral devices. The following information is provided for each unit displayed on the DFC Status Page.

- Major state of the unit. (Refer to the “Major and Minor State Display Field Values” section.)
- Minor state of the unit (if available). (Refer to the “Major and Minor State Display Field Values” section.)
- Usability of the unit (Y = usable, N = unusable).
- Essential status of the unit (E = essential, M = manually nonremovable, Blank = nonessential).
- Operational microcode for the DFC (FIRMWARE or PUMPCODE).
- Overload status for the DFC (OVLD = overload, Blank = normal).

The DFC Status Page commands provide maintenance functions that can be used when the operating system is in disk independent operation mode. Most of these commands have parameter fields associated with them. The parameter fields are entered following the command number. The maximum number of characters for the command line is 32.

The command parameters vary based on the *UNIX* RTR operating system release. Explanations of the parameters can be obtained using the DFC Status Page “help” facility or by entering a command, that normally requires parameters, without parameters.

To invoke the DFC Status Page “help” facility, enter:

command-number,help<CR>

or

command-number,h<CR>

or

command<CR> (for commands that normally require parameters).

This will cause an explanation of the parameters fields to appear in the three-line communication area at the bottom of the DFC Status Page.

Table 12-18 provides a function description of the DFC Status Page commands.

⇒ **NOTE:**

Some of the commands are functionally equivalent to similar *UNIX* RTR operating system input messages. For additional information on the command functionality, refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*.

Table 12-18. DFC Status Page Commands

Command (Description)	Function Description
600 (INIT MHD)	Format an MHD. This command is functionally equivalent to the INIT:MHD input message.
601 (VFY MHD)	Verify an MHD. This command is functionally equivalent to the VFY:MHD input message.
602 (RST DFC)	Restore a DFC, controller only. This command is functionally equivalent to the RST:DFC input message executed with the CONT option.
603 (DUMP DEFECT)	Dump an MHD defect table. This command is functionally equivalent to the DUMP:MHD,DEF input message.
604 (RST/RMV SBUS)	Restore (controller only) or remove an SBUS. This command is functionally equivalent to the RST:SBUS input message executed with the CONT option and to the RMV:SBUS input message.
605 (RST MHD)	Restore an MHD. When the system is not in FULL DIOP, this command is functionally equivalent to the RST:MHD input message. When the system is in FULL DIOP, a restore will be allowed only if the MHD does not have an active mate and if the MHD is not marked essential.
606 (RST MT)	Restore a Small Computer System Interface (SCSI) Magnetic Tape (MT). This command is displayed only if a SCSI DFC (supporting the MT) is equipped. It is functionally equivalent to the RST:MT input message.

Table 12-18. DFC Status Page Commands (Contd)

Command (Description)	Function Description
607 (DUMP BKTAPE)	Dump header information from a multivolume Load Disk from Tape (LDFT) formatted Digital Audio Tape (DAT) unit. This command is displayed and functional only if a DAT unit is equipped and the system is in FULL DIOP. The following information is displayed for each logical volume on the DAT: ▪ Session number. ▪ Volume number. ▪ Volume name. ▪ Number of disk blocks contained in the logical volume. ▪ Source disk(s) used to write the logical volume. ▪ Date the logical volume was written to tape. The total number of disk blocks on the multivolume tape is also displayed.
620 (LOAD MHD)	Load an MHD from tape. This command is only displayed and functional when the system is in FULL DIOP. It is similar to the EAI LDFT function. The following operations can only be performed with a single volume tape sequence: ▪ Load both the generic and database tapes (BOTH). ▪ Load the generic tape only (GEN). ▪ Load the database tapes only (DBONLY). ▪ Continue the load disk from tape process after a new tape has been mounted (CONT). For a multivolume DAT, a logical volume identified by a session and volume number can be loaded.

Table 12-18. DFC Status Page Commands (Contd)

Command (Description)	Function Description
677 (SET MSG LVL)	Select the DFC driver Enhanced Information Report (EIR) format for the specified DFC. One of the following formats can be selected: ▪ Turn off EIR (n). ▪ Long EIR format (l). ▪ Short EIR format (s).
688 (CLEAR BUFF)	Clear the three-line communication area at the bottom of the DFC Status Page.
699 (ABORT CMD IP)	Terminate the currently running DFC Status Page menu command. This command is displayed and functional only when the system is in FULL DIOP.

How to Access the DFC Performance Page

To access the DFC Performance Pages (Figures 12-38 and 12-39) on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	At the CMD< prompt, access the DFC Index Page by entering 110<CR> . (Refer to the “How to Access and Interpret the DFC Index Page” section.)
4	From the DFC Index Page, determine the DFC Performance Page menu number for the desired DFC. At the CMD< prompt, enter the menu number followed by a carriage return.

The number of DFC Performance Pages and the number of DFCs displayed on the pages will vary based on the *UNIX* RTR operating system release and the hardware equipped. Figures 12-38 and 12-39 show examples of typical DFC Performance Pages.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK							
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM						
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS									
CMD<_622,1 1000 OK				121 - DFC PERFORMANCE PAGE									
	CMP	AVG	MAX		CMP	AVG	MAX						
DFC-0	3	252	256	DFC-1	0	0	0						
MHD-0	3	252	256	MHD-1	0	0	0						
MHD-2	0	0	0	MHD-3	0	0	0						
MHD-4	0	0	0										
<table border="1"> <thead> <tr> <th>CMD</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>622</td> <td>PERFORMANCE</td> </tr> <tr> <td>688</td> <td>CLEAR BUFF</td> </tr> </tbody> </table>								CMD	DESCRIPTION	622	PERFORMANCE	688	CLEAR BUFF
CMD	DESCRIPTION												
622	PERFORMANCE												
688	CLEAR BUFF												
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <													

Figure 12-38. Typical DFC Performance Page for a System Equipped with SMD DFCs 0 and 1

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK					
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM				
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS							
CMD<622,	1 1000 OK				123 - DFC	PERFORMANCE PAGE					
	CMP AVG MAX		CMP AVG MAX		CMP AVG MAX		CMP AVG MAX				
DFC-2	0 0 0			DFC-3	0 0 0						
SBUS-4	0 0 0	SBUS-6	0 0 0	SBUS-5	0 0 0	SBUS-7	0 0 0				
MHD-20	0 0 0	MHD-22	0 0 0	MHD-21	0 0 0	MHD-23	0 0 0				
MHD-24	0 0 0	MHD-26	0 0 0	MHD-25	0 0 0	MHD-27	0 0 0				

CMD	DESCRIPTION
622	PERFORMANCE
688	CLEAR BUFF

REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***

Figure 12-39. Typical DFC Performance Page for a System Equipped with SCSI DFCs 2 and 3

How to Interpret the DFC Performance Page

The DFC Performance Page provides the optional DFC performance data for the 3B20D/3B21D computers. The DFC performance data is not available when the operating system is in **disk independent operation mode**. Table 12-19 provides the function descriptions of the DFC Performance Page menu commands.

Table 12-19. DFC Performance Page Menu Commands

Command (Description)	Function Description
622 (PERFORMANCE)	Display statistics on jobs done for units in the DFC community. The DFC Performance Page “help” facility can be used to obtain the menu command parameters. The parameters are provided to: ▪ Specify the number of seconds (0-60) between statistic updates. ▪ Stop updates in progress. ▪ Specify the number of times to repeat the updates (default = 20).
688 (CLEAR BUFF)	Clear the three-line communication area at the bottom of the DFC Performance Page.

The following information (Name/Meaning) is provided for each unit displayed on the DFC Performance Page:

CMP	Jobs completed by the DFC for the unit. This field is normalized to show the number of jobs that were completed per second.
AVG	The average size of the jobs completed for a unit (number of 512-byte blocks).
MAX	The maximum size of a job completed in a time interval (number of 512-byte blocks).

⇒ NOTE:

The DFC fields are the sum of the individual unit fields associated with the DFC. For the SCSI DFC, the SBUS fields are the sum of all the unit fields associated with this SBUS.

How to Access and Interpret the Disk File System Access Index Page

To access the Disk File System Access (DFSA) Index Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
 NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.	
3	At the CMD< prompt, enter 110<CR> .

The display area indicates the DFSA menu pages that are available and the DFCs supported by each menu page. Figure 12-40 shows a typical DFSA Index Page.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				110	DISK SYSTEM ACCESS INDEX		
<u>CMD</u>	<u>PAGE</u>	<u>TITLE</u>		<u>CMD</u>	<u>PAGE</u>	<u>TITLE</u>	
120	DFSA	DFC0-1	STATUS	122	DFSA	DFC2-3	STATUS
121	DFSA	DFC0-1	PERFORMANCE	123	DFSA	DFC2-3	PERFORMANCE
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-40. Typical DFSA Index Page

How to Access the DFSA Status Page

To access the DFSA Status Page on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	 NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	Access the DFSA Index Page. (Refer to the "How to Access and Interpret the Disk File System Access Index Page" section.)
4	From the DFSA Index Page, determine the DFSA Status Page menu number for the desired DFC. At the CMD< prompt, enter the menu number followed by a carriage return.

The number of DFSA Status Pages and the number of DFCs displayed on the pages will vary based on the *UNIX* RTR operating system release and the hardware equipped. Figures 12-41 through 12-45 show examples of typical DFSA Status Pages.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				120 - DISK FILE SYSTEM ACCESS			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX			
DFC-0	ACT	Y		DFC-1	ACT	Y	
MHD-0	ACT	Y E		MHD-1	OOS	N E	
MHD-2	ACT	Y		MHD-3	OOS	N	
MHD-4	ACT	Y					
MENU		DESCRIPTION	MENU		DESCRIPTION	MENU	
600,n [v][s]		INIT MHD	603,n [t;fn]		DUMP DEFECT	677,n A	
601,n [s]		VFY MHD				688	
602,n [u]		RST DFC				CLEAR BUFF	
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-41. Typical DFSA Status Page for a System Equipped with SMD DFCs 0 and 1

OFFICE NAME AND TYPE			GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM	
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS				
CMD<_				122 - DISK FILE SYSTEM ACCESS				
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX				
DFC-2	ACT	Y		DFC-3	ACT	Y		
MHD-16	ACT	Y E		MHD-17	OOS	N		
MHD-18	ACT	Y		MHD-19	OOS	N		
MHD-20	ACT	Y						
MENU		DESCRIPTION	MENU		DESCRIPTION	MENU		DESCRIPTION
600,n [v][s]		INIT MHD	603,n [t:fn]		DUMP DEFECT	677,n A		SET MSG LVL
601,n [s]		VFY MHD				688		CLEAR BUFF
602,n [u]		RST DFC						
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***								
<								

Figure 12-42. Typical DFSA Status Page for a System Equipped with SMD DFCs 2 and 3

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLKG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD<_				120 - DISK FILE SYSTEM ACCESS			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX			
DFC-0	ACT Y	PUMPCODE		DFC-1	ACT Y	PUMPCODE	
SBUS-0	ACT Y	SBUS-2	ACT Y	SBUS-1	ACT Y	SBUS-3	ACT Y
MHD-0	ACT Y E	MHD-2	ACT Y	MHD-1	ACT Y E	MHD-3	ACT Y
MHD-4	ACT Y	MHD-6	ACT Y	MHD-5	ACT Y	MHD-7	ACT Y
MHD-8	ACT Y	MHD-10	ACT Y	MHD-9	ACT Y	MHD-11	ACT Y
MHD-12	ACT Y	MHD-14	ACT Y	MHD-13	ACT Y	MHD-15	ACT Y
MENU		DESCRIPTION		MENU		DESCRIPTION	
600,n [v][s]		INIT MHD		603,n [t:fn]		DUMP DEFECT	
601,n [s]		VFY MHD		604,n a [u]		RST/RMV SBUS	
602,n [u]		RST DFC		688		SET MSG LVL	
						CLEAR BUFF	
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-43. Typical DFSA Status Page for a System Equipped with SCSI DFCs 0 and 1

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				122 - DFC STATUS PAGE			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = SIMPLEX			
DFC-2	OOS	N	FIRMWARE	DFC-3	OOS	N	FIRMWARE
SBUS-4	OOS	N	SBUS-6	OOS	N	SBUS-5	OOS
MHD-20	OOS	N	MHD-22	OOS	N	MHD-21	OOS
MHD-24	OOS	N	MHD-26	OOS	N	MHD-25	OOS
						MHD-27	OOS
MENU		DESCRIPTION		MENU		DESCRIPTION	
600,n [v][s d]		INIT MHD		603,n [t;fn]		DUMP DEFECT	
601,n [s]		VFY MHD		604,n a [u]		RST/RMV SBUS	
602,n [u]		RST DFC				688	
						SET MSG LVL	
						CLEAR BUFF	
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS ***							
<							

Figure 12-44. Typical DFSA Status Page for a System Equipped with SCSI DFCs 2 and 3

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK	
SYS EMER	CRITICAL	MAJOR	MINOR	BLKG/PWR	BLD INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS			
CMD<_				120 - DISK FILE SYSTEM ACCESS			
OPTION LEVEL = UCLDIOP				CURRENT LEVEL = FULL DIOP			
DFC-0	ACT	Y	FIRMWARE	DFC-1	ACT	Y	FIRMWARE
SBUS-0	ACT	Y	SBUS-2	ACT	Y	SBUS-3	ACT
MHD-0	OOS	FDIO Y E	MHD-2	ACT	Y	MHD-3	OOS
MHD-4	ACT	Y	MHD-6	ACT	Y	MHD-7	ACT
MHD-8	ACT	Y	MHD-10	ACT	Y	MHD-11	ACT
MHD-12	ACT	Y	MHD-14	ACT	Y	MHD-15	ACT
MENU		DESCRIPTION		MENU		DESCRIPTION	
600,n [v][s]	INIT MHD	603,n [t:fn]	DUMP DEFECT	677,n A	SET MSG LVL		
601,n [s]	VFY MHD	604,n a [u]	RST/RMV SBUS	688	CLEAR BUFF		
602,n [u]	RST DFC	620,n td a	LOAD MHD	699	ABORT CMD IP		
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <							

Figure 12-45. Typical DFSA Status Page for a System, in FULL DIOP, Equipped with SCSI DFCs 0 and 1

How to Interpret the DFSA Status Page

The `OPTION LEVEL` display field indicates the DIOP feature level that is selected in the *ecdorg* form of the ECD. It will be one of the following:

<code>NODIOP</code>	DIOP capabilities are not provided.
<code>DIOP</code>	DIOP capabilities are conditionally provided. This option level is not supported in some versions of the <i>UNIX</i> RTR operating system.
<code>UCLDIOP</code>	DIOP capabilities are unconditionally provided.

The `CURRENT LEVEL` display field indicates the *UNIX* RTR operating system DIOP status. It will be one of the following:

<code>NORMAL</code>	DIOP is disabled.
<code>INITIALIZING</code>	The on-line CU is initializing. If <code>UCLDIOP</code> is selected in the ECD, <code>NODIOP</code> will be displayed as the <code>OPTION LEVEL</code> during the CU initialization interval. The DIOP feature will be enabled when the CU initialization completes.
<code>SIMPLEX</code>	DIOP is enabled and at least one essential MHD of a duplex pair is out of service.
<code>DUPLEX</code>	DIOP is enabled and all essential MHDs are active.
<code>CONDITIONAL DIOP</code>	DIOP is enabled and pending further errors, the last active essential MHD of a duplex pair is about to be removed from service.
<code>FULL DIOP</code>	DIOP is enabled and the last active essential MHD of a duplex pair has been removed from service.

NOTE:

In later *UNIX* RTR operating system releases, only essential MHDs which contain the `ROOT` and `SWAP` partitions affect the DIOP status displayed in the `CURRENT LEVEL` field.

The DFSA Status Page displays the status of the equipped DFC, SBUSs, and MHDs. The following information is provided for each unit displayed on the DFSA Status Page.

- Major state of the unit. (Refer to the “Major and Minor State Display Field Values” section.)
- Minor state of the unit (if available). (Refer to the “Major and Minor State Display Field Values” section.)
- Usability of the unit (Y = usable, N = unusable).
- Essential status of the unit (E = essential, M = manually nonremovable, Blank = nonessential).
- Operational microcode for the DFC (FIRMWARE or PUMPCODE).
- Overload status for the DFC (OVLD = overload, Blank = normal).

The DFSA Status Page menu commands provide disk maintenance functions that can be used when the operating system is in disk independent operation mode. Most of these menu commands have parameter fields associated with them. The parameter fields are entered following the menu number. The maximum number of characters for the command line is 32.

The menu command parameters shown on the DFSA Status Page vary based on the *UNIX* RTR operating system release. Explanations of the parameters can be obtained using the DFSA Status Page “help” facility.

To invoke the DFSA Status Page “help” facility, enter:

menu-number,help<CR>

or

menu-number,h<CR>

This will cause an explanation of the parameters fields to appear in the three-line communication area at the bottom of the DFSA Status Page.

Table 12-20 provides a function description of the DFSA Status Page menu commands.



NOTE:

Some of the menu commands are functionally equivalent to similar *UNIX* RTR operating system input messages. For additional information on the command functionality, refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*.

Table 12-20. Disk File System Access Status Page Menu Commands

Menu Description	Function Description
600 (INIT MHD)	Format an MHD. This command is functionally equivalent to the INIT:MHD input message.
601 (VFY MHD)	Verify an MHD. This command is functionally equivalent to the VFY:MHD input message.
602 (RST DFC)	Restore a DFC, controller only. This command is functionally equivalent to the RST:DFC input message executed with the CONT option.
603 (DUMP DEFECT)	Dump an MHD defect table. This command is functionally equivalent to the DUMP:MHD,DEF input message.
604 (RST/RMV SBUS)	Restore (controller only) or remove an SBUS. This command is functionally equivalent to the RST:SBUS input message executed with the CONT option and to the RMV:SBUS input message.

Table 12-20. Disk File System Access Status Page Menu Commands (Contd)

Menu Description	Function Description
620 (LOAD MHD)	Load an MHD from tape. This command is only displayed and functional when the system is in FULL DIOP. It is similar to the Emergency Action Interface (EAI) Load Disk from Tape (LDFT) function. The following operations can be performed: ▪ Load both the generic and database tapes (BOTH). ▪ Load the generic tape only (GEN). ▪ Load the database tapes only (DBONLY). ▪ Continue the load disk from the tape process after a new tape has been mounted (CONT).
677 (SET MSG LVL)	Select the disk driver Enhanced Information Report (EIR) format for the specified DFC. One of the following formats can be selected: ▪ Turn off EIR (n). ▪ Long EIR format (l). ▪ Short EIR format (s).
688 (CLEAR BUFF)	Clear the three-line communication area at the bottom of the DFSA Status Page.
699 (ABORT CMD IP)	Terminate the currently running DFSA Status Page menu command. This command is displayed and functional only when the system is in FULL DIOP.

How to Access the DFSA Performance Page

To access the DFSA Performance Pages on the MTTY terminal, use the following procedure:

Step	Procedure
1	Select a normal system display page by pressing the NORM DISP function key.
2	If the terminal is not in the command mode, place it in this mode by pressing the CMD MSG function key.
	⇒ NOTE: When in the command mode, the CMD< prompt is displayed and the cursor is positioned in the command input area in the upper left portion of the display.
3	Access the DFSA Index Page. (Refer to the “How to Access and Interpret the Disk File System Access Index Page” section.)
4	From the DSFA Index Page, determine the DFSA Performance Page menu number for the desired DFC. At the CMD< prompt, enter the menu number followed by a carriage return.

The number of DFSA Performance Pages and the number of DFCs displayed on the pages will vary based on the *UNIX* RTR operating system release and the hardware equipped. Figures 12-46 and 12-47 show examples of typical DFSA Performance Pages.

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK							
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM						
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS									
CMD<_622,1 1000 OK				121 - DFSA PERFORMANCE									
	CMP	AVG	MAX		CMP	AVG	MAX						
DFC-0	3	252	256	DFC-1	0	0	0						
MHD-0	3	252	256	MHD-1	0	0	0						
MHD-2	0	0	0	MHD-3	0	0	0						
MHD-4	0	0	0										
<table border="0"> <tr> <td>MENU</td> <td>DESCRIPTION</td> </tr> <tr> <td>622 [,s [t]]</td> <td>PERFORMANCE</td> </tr> <tr> <td>688</td> <td>CLEAR BUFF</td> </tr> </table>								MENU	DESCRIPTION	622 [,s [t]]	PERFORMANCE	688	CLEAR BUFF
MENU	DESCRIPTION												
622 [,s [t]]	PERFORMANCE												
688	CLEAR BUFF												
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE /PDS *** <													

Figure 12-46. Typical DFSA Performance Page for a System Equipped with SMD DFCs 0 and 1

OFFICE NAME AND TYPE		GENERIC AND ISSUE		TERMINAL IDENTIFICATION		DATE AND 24-HOUR CLOCK							
SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLD INH	CKT LIM	SYS NORM						
TRAFFIC	SYS INH	CU	CU PERPH	OS LINKS									
CMD<622, 1 1000 OK				123 - DFC PERFORMANCE PAGE									
	CMP	AVG	MAX		CMP	AVG	MAX						
DFC-2	0	0	0	DFC-3	0	0	0						
SBUS-4	0	0	0	SBUS-6	0	0	0						
MHD-20	0	0	0	MHD-22	0	0	0						
MHD-24	0	0	0	MHD-26	0	0	0						
				SBUS-5	0	0	0						
				MHD-21	0	0	0						
				MHD-25	0	0	0						
				SBUS-7	0	0	0						
				MHD-23	0	0	0						
				MHD-27	0	0	0						
<table border="1"> <thead> <tr> <th>MENU</th> <th>DESCRIPTION</th> </tr> </thead> <tbody> <tr> <td>622 [,s [t]]</td> <td>PERFORMANCE</td> </tr> <tr> <td>688</td> <td>CLEAR BUFF</td> </tr> </tbody> </table>								MENU	DESCRIPTION	622 [,s [t]]	PERFORMANCE	688	CLEAR BUFF
MENU	DESCRIPTION												
622 [,s [t]]	PERFORMANCE												
688	CLEAR BUFF												
REPT CFTSHL TERMINAL IN SERVICE *** FULL CAPABILITY MODE / PDS *** <													

Figure 12-47. Typical DFSA Performance Page for a System Equipped with SCSI DFCs 2 and 3

How to Interpret the DFSA Performance Page

The DFSA Performance Page provides the optional disk access performance data for the 3B20D/3B21D computers. The DFSA performance data is not available when the operating system is in **disk independent operation mode**. Table 12-21 provides the DFSA Performance Page menu command function descriptions.

Table 12-21. DFSA Performance Page Menu Commands

Menu Command	Function Description
622 (PERFORMANCE)	Display statistics on jobs done for units in the disk community. The DFSA Performance Page “help” facility can be used to obtain the menu command parameters. The parameters are provided to: ▪ Specify the number of seconds (0-60) between statistic updates. ▪ Stop updates in progress. ▪ Specify the number of times to repeat the updates (default = 20).
688 (CLEAR BUFF)	Clear the three-line communication area at the bottom of the DFSA Performance Page.

The following information (name/meaning) is provided for each unit displayed on the DFSA Performance Page.

CMP	Jobs completed by the DFC for the unit. This field is normalized to show the number of jobs that were completed per second.
AVG	The average size of the jobs completed for a unit (number of disk blocks).
MAX	The maximum size of a job completed in a time interval (number of disk blocks).

⇒ NOTE:

The DFC fields are the sum of the individual MHD fields associated with the DFC. For SCSI DFC, the SBUS fields are the sum of all the MHD fields associated with this SBUS.

How to Use the Craft Shell and Dialogue Shell with the Input Message Catalog

The Craft Shell (CFTSHL) and Dialogue Shell (DLGSHL), hereafter referred to as the shell unless otherwise noted, are the portions of the *UNIX* RTR operating system that handle input messages from a maintenance terminal in either PDS or MML syntax.

The shell provides the interface between a maintenance terminal and the *UNIX* RTR operating system by supporting the maintenance terminal input language.

A shell process is created for each maintenance terminal in the system. Client programs stored in the file system under various directories are invoked by the shell in response to a command from a maintenance terminal.

After a DLGSHL process has been created, a special login - type procedure must be followed to begin a dialogue session. A dialogue request character, defined in the *eaiopt* record in the ECD (default : control - G), must be entered, possibly followed by a password. The need for a password is determined by the value of the *auth_chk* field in the device's *getty* record in the ECD. If "no authority" is specified, then no password is required.

The passwords may be manipulated using the terminal and personal authority, including the following input commands: **ADD:PAUTH**, **ADD:TAUTH**, **CHG:PAUTH**, **DEL:PAUTH**, **DEL:TAUTH**, **RNM:PSSWD**, and **ADD:PROFL**.

A CFTSHL may also be created by another process. For example, DAP creates a CFTSHL to execute a menu command. For some menu commands, DAP generates an input message character string and passes it to the CFTSHL. If DAP passes an erroneous input message, the CFTSHL will echo the invalid input message string to the originating terminal along with an appropriate error acknowledgment.

The DAP communicates with the DLGSHL via a message interface. For those menu commands that result in an input message character string, DAP will send that string to the existing DLGSHL to be parsed instead of creating a new DLGSHL.

Input Message Catalog Database

For Release 1 and later, CFTSHL accesses an input message catalog database called *imcatlg* to validate the contents of input messages. The input message catalog contains information about input message formats, keywords, valid keyword argument values, and full pathnames of the client programs that execute the specified commands. The input message catalog also contains information that indicates whether the CFTSHL or a client program is to display the appropriate acknowledgment for each input message. The DLGSHL performs the same functions but in Release 1 only.

Capability Modes

The shell operates in three different capability modes: full mode, partial mode, and limp mode. The full capability mode assumes that all *UNIX* RTR operating system and application input messages are defined in the input message catalog database. The partial capability mode assumes that all *UNIX* RTR operating system input messages are defined in the catalog but not all application input messages are defined in the catalog. The shell will work in the limp mode if the input message catalog database cannot be accessed (that is, accidentally deleted).

The shell capability mode is stored in the data field of the spooler information (*splinfo*) record of the ECD. It can be set to either "FULL" or "PART" using the RC/V system. The *UNIX* RTR operating system is delivered with the capability mode set to FULL. Therefore, applications that do not want to add their messages into the input message catalog must change the mode from FULL to PART using the RC/V system.

If the shell operates in full capability mode, users have the following capabilities:

- Early error detection requiring less central processing unit time for incorrect and/or incomplete input messages
- Faster acknowledgment response time — three to five times faster compared to previous releases (approximately 0.2 to 1 second depending upon workload)
- Improved performance with elimination of the directory search for the command execution process
- Use of the help facility (CFTSHL only)
- Use of flexible input message formats.

If the shell cannot access the input message catalog for some reason, the shell will operate in minimum capability mode (called limp mode), and those capabilities described previously will not be available. However, the CFTSHL will continue to function in the same way as implemented in previous releases.

When operating in full capability mode, the shell will reject, with an error acknowledgment, any input messages that are not represented in the input message catalog database. Consequently, a missing or erroneous entry in the input message catalog will cause the shell to reject the associated input message even if it is typed correctly. If this should occur, the user can force the shell to accept the message and attempt to execute it by entering the special command immediately following the error acknowledgment:

In PDS, **EXC:UCL!**
In MML, **EXC:UCL;**

The command affects only the immediately preceding rejected input messages. It will not cause the shell to reexecute a successful message, and it has no effect on any subsequent message. If it is used to force the shell to execute an invalid message, the invalid message will be rejected by the client program.

Shell Configuration

The shell configuration information is stored in the ECD and in the *.pname* file for the terminal on which it is running. When started, each shell establishes its environment by obtaining the first line from the *.pname* file and by reading configuration information from the ECD.

splinfo Record in the ECD

The shell sets its input message syntax and capability mode by reading the miscellaneous spooler information (*splinfo*) record from the ECD. The *splinfo* record defines miscellaneous information used by the output spooler.

- Input message and output message syntax. This field can be set to MML or PDS. The shell and spooler read this field to determine the proper IO syntax.
- FULL/PART capability mode. This field can be set to FULL or PART. The shell reads this field to determine the proper capability mode.

When it is initialized, the shell reports the capability mode and input message syntax it is using in the REPT xxx SHL TERMINAL IN SERVICE output message, where xxx is either CFT or DLG.

Also included in the *splinfo* record are three fields used to define the layout of the output report header and trailer and the solicited report header (output resulting from an input command seen only under a DLGSHL environment). Any printable character can be entered into these fields as well as variables that will be replaced by the spooler as follows:

Variable	Description
% s	system name
% d	date stamp (in format specified in <i>eaiopt</i> record)
% t	time
% b	basic time stamp (minutes after the hour)
% i	identity (person or terminal)
% n	dialogue number
% c	command number
% g	global report sequence number
% m	report sequence number per message class
% r	report destination (message class)
% p	process ID of sending process
% u	utility ID of sending process

ECD Changes for the Dialogue Shell

Fields in the *eaiopt*, *getty*, *device*, and *logdev* forms must be modified for the DLGSHL.

Pathname File

During initialization, the shell assumes that a *.pname* file is in the current directory. Each line of the *.pname* file describes an environment for the shell as follows:

a:b:c:d:e:f:g

Where:

- a = Numeric labels (0 – 255)
- b = Shell directory search list — up to eight directories, minimum of one required
- c = Current directory
- d = User identification
- e = Group identification
- g = Optional user-defined string(s).

How to Field Update the Input Message Catalog

Whenever the format of an input message is changed by a field update, the input message catalog must also be updated.

The following procedure is used to field update the input message catalog:

Step	Procedure
1	Replace the <i>imcatlg</i> file using the UPD:APPLY:FILER command.
2	Invoke the CLR:IMCAT command to determine the segment index(es) used by the old catalog and to invalidate them.
3	Kill all processes connected to the old input message catalog using the STOP:EXC:ANY command. These processes are automatically restarted, and the new input message catalog is loaded into main memory by the first process restarted.

How to Invoke the CFTSHL Help Facility

The **help** facility of the CFTSHL provides assistance to users entering *UNIX RTR* operating system PDS/MML input messages and can be used to complete or correct errors in input messages. If an input message is entered incorrectly, the CFTSHL will reject it with an appropriate error acknowledgment. If the user does not understand or know what to do with the error acknowledgment, "help" can be obtained from the CFTSHL by simply entering a question mark (?). The user can also get help by entering a ? at the end of a partial input message.

The CFTSHL interprets the "?" at the end of a partial input message or following an error acknowledgment as a request for information about the format of the message. A second "?" is interpreted as a request to enter the prompting mode. In the prompting mode, the CFTSHL will prompt the user for each keyword and argument in the message and provide information about the values that can be entered. When a complete input message has been constructed, the user may either execute it or cancel it. The help session is then completed; that is, help is provided for only one input message at a time.

When help is requested on a C/D terminal with 25 lines or less, the CFTSHL will cause the currently-displayed pages to be erased and the status summary area page to be displayed instead. This will provide enough space for the message format and prompting information to be printed without scrolling off the screen. When the help session is concluded by either executing or canceling the input message, the CFTSHL will cause the previously displayed pages to be redisplayed.

How to Use the Flexible Input Message Formats

To improve the effectiveness and productivity of the user, the shell will accept input message formats that are easier to use and remember. This feature allows input messages to be entered with commas as field separators instead of colons and semicolons after the verb and first colon. More specifically, all options may be included in a single, comma-separated, parameter block. A few commands that have positionally defined data fields are exceptions to this. The colon before the positionally defined data field must still be entered as a colon.

The shell will convert a flexibly formatted input message to the format specified in the input message catalog before it passes the message to the client process. Consequently, if the client process echoes the input message to an output device, it will appear with standard punctuation (and in MML with the DATA keyword at the beginning of the data field). Input messages constructed by the **help** facility (CFTSHL only) will always be in standard (input message catalog) format.

How to Specify the Incore Size of the Message Spooler's Output Message Database in the ECD

The output message spooler is the *UNIX* RTR operating system user-level program that handles formatting and spooling of output messages to devices. The spooler uses an Output Message Database (OMDB) to first format output messages defined and then spool them. Nearly all *UNIX* RTR operating system messages use this interface. Messages may also be passed already formatted in spool-only request; for these messages, the OMDB is not accessed.

The ECD *splinfo* record contains a field called "*omdb_mode*" which is issued by the output message spooler to determine how many segments of the OMDB should be kept incore during normal system operation. The valid range for the "*omdb_mode*" field is 1 to 15 inclusive. The total number of OMDB segments and the number of segments that are active (incore) are displayed in both the REPT CSOP IN SERVICE and ACTV OMDB COMPLETED output messages.

If the disk copy of the OMDB (*/cft/spl/omdb*) contains three segments and the "*omdb_mode*" field is set to 1, the output message spooler will keep one of the three segments incore at any given time. The output message spooler will interchange the segments (remove the segment that is currently in memory and bring in a new segment) whenever the need arises in formatting messages.

The advantage of making the "*omdb_mode*" field a small value is that less memory will be consumed by the incore OMDB. However, the disadvantage of making the "*omdb_mode*" less than the total number of OMDB segments is that additional processing is necessary to bring in OMDB segment(s) which are not currently in memory. This increases the time needed to format a message.

If the "*omdb_mode*" value is greater than or equal to the size of the OMDB, all OMDB segments will be incore all the time. The memory consumed will never exceed the actual number of OMDB segments.

For applications using disk independent operations, the output message spooler will be notified whenever the essential disks change states (that is, from simplex to duplex, duplex to simplex). When the disks go into the simplex mode, the output message spooler will bring all OMDB segments into memory regardless of the value of the "*omdb_mode*" field. These segments will be locked in memory, thus decreasing the amount of swappable main memory. When the disks return to the duplex state, the segments will be unlocked and the extra incore OMDB segments will be released according to the "*omdb_mode*" setting.

How to Field Update the Output Message Database

Whenever output messages are added, changed, or deleted, the OMDB must also be field updated.

The following procedure is used during a field update of the OMDB:

Step	Procedure
1	Replace the output message database file <i>/cft/spl/omdb</i> using the UPD:APPLY:FILTER command.
2	Invoke the ACTV:OMDB command to force the output message spooler to disconnect from the old OMDB and to begin using the updated one.
3	Invoke the APPLY:OMDB command during the PERM phase of the field update. This will reapply any craft changes that had been previously made to the output message alarm levels or message classes using the UPD:OMDB command.
4	Invoke another ACTV:OMDB command to force the output message spooler to refresh its incore copy of the OMDB with these reapplied alarm level/message class changes.

How to Abort Commands or Messages

To abort input messages or commands, use the kill line character or abort input character. The kill line character is \$ for PDS syntax and @ for MML syntax. The abort input character for PDS syntax is & (ampersand) and DEL for MML syntax. Input is also aborted prior to completion if the intercharacter time expires.

To delete the last character typed (except the execute character), use the _ (underscore) character or backspace.

How to Retire Alarms

To retire an alarm, press the **ALM RLS** function key. When this key is pressed, it retires both audible and visual alarms in the following manner:

- For software sanity, the flashing *type* indication (for example, CU or BLDG/PWR) changes to reverse video and the reverse video *severity* indication (CRITICAL, MAJOR, or MINOR) is retired along with audible alarms.
- For software insanity, the visible system emergency indication changes to reverse video and audible alarms are retired.

How to Reinitialize the Craft Interface

On the MTTY terminal, use the following procedure to reinitialize the craft interface:

Step	Procedure
1	Press the EA DISP function key to access the emergency action page.
2	At the CMD< prompt, enter 42! Application Parm: <i>n</i> . <i>n</i> equals 1 if level 1 craft initialization is requested <i>n</i> equals 2 if level 2 craft initialization is requested <i>n</i> equals 3 if level 3 craft initialization is requested.
3	At the CMD< prompt, enter 15! .
	 NOTE: A level 1 craft initialization is the default request when the Application Parm is not set.

This procedure is used when:

- A field update to a craft interface process requires a craft interface initialization.
- All system functions except the MTTY terminal appear to be operating correctly.
- A craft lockout is encountered.

When command 15 is entered with application parameter 1, all the craft processes are terminated and restarted. PRMs are displayed on the EAI Page as the processes are restarted. Use this procedure only when the craft interface is not responding normally to craft input messages. Use the **INIT:ULARP** command if only one craft interface process has terminated. This also restarts the MTTY controllers.

When command 15 is entered with application parameter 2, all the killable non-essential user and supervisor processes are terminated and restarted. Also, all the MTTY and TTY controllers get restarted. Use this procedure only when a total craft lockout happens and level 1 craft initialization cannot recover it.

When command 15 is entered with application parameter 3, all killable essential and nonessential user and supervisor processes are terminated and restarted. This command also causes reboot of all the IOPs. It should be used only when a total craft lockout happens and levels 1 or 2 craft initialization cannot recover it.

How to Switch Devices to Alternate MTTYC

One of the following methods should be used to switch one or both switchable devices (that is, the ROP or MTTY terminal) to an alternate MTTYC:

- **SW:PORTSW** input messages. Refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*.



NOTE:

The port switch must be in the **AUTO** position.

- Menu input command (401, 402, or 403) shown on the CPDP 102.



NOTE:

The port switch must be in the **AUTO** position.

- Manually flip the port switch associated with the switchable device to either 0 or 1 (0 = MTTYC 0, 1 = MTTYC 1).



NOTE:

In either of these two positions, the port switch cannot be reconfigured using the previous two methods.

Contents

Introduction	13-1
Update Methods and Processes	13-1
■ Update Methods	13-1
File Replacement	13-2
Function and Byte Replacement	13-2
■ Field Updatable Processes	13-3
Nonkillable Processes	13-3
Boot Image Component Processes	13-3
Quasi-Killable Processes	13-4
Killable Processes	13-4
Killable Kernel Processes	13-5
Diagnostic Processes	13-5
Peripheral Microcode	13-5
Broadcast Warning Messages	13-6
■ BWM File Structure	13-7
■ BWM Components	13-8
SCANS File	13-9
HEADER File	13-11
MSGs File	13-12
Update Files	13-15
■ Update Package Rules	13-15
■ Update Package States	13-18
Typical Update Package States	13-19
Other Update Package States	13-20
Messages Files Execution and Display	13-22
■ Field Update Page	13-22

Contents

■ Menu Indicators	13-24
Configuration Indicators	13-24
Scope Indicators	13-24
Action Indicators	13-25
Key Points About Menu Indicators	13-25
Allowable Scope/Action Combinations	13-27
Field Update Command Execution Locations	13-29
Scrolling Area Only Commands	13-29
Scrolling Area/Poke Messages	13-30
Message File Commands	13-30
■ Update Display Command	13-31
■ Troubleshooting Hints	13-31
■ Verification of Proper Installation	13-32
Field Update Session Example	13-34
■ Call Up the Field Update Page	13-36
■ Display the APPLY Section	13-37
■ Enter the BWM Number	13-38
■ Display the APPLY Section	13-39
■ Display Page 2 of the APPLY Section	13-40
■ Display the PERM Section	13-41
■ Execute the PERM Section	13-42
■ Execute the APPLY Section	13-43
Message 1 Continues to Execute	13-44
Message 2 Continues to Execute	13-45
Message 3 Continues to Execute	13-46
Message 4 of the APPLY Section Fails	13-47
Reset the Failed Message	13-48
■ Reexecute the APPLY Section	13-49
APPLY Section Completes Execution	13-50
■ Execute the BKOUT Section	13-51
BKOUT Section Completes Execution	13-52
■ Redisplay the APPLY Section	13-53
■ Reexecute the APPLY Section	13-54
Message 1 Completes Execution	13-55
Message 2 Completes Execution	13-56
Message 3 Completes Execution	13-57
APPLY Section Completes Execution	13-58
■ Display the SOAK Section	13-59
■ Execute the SOAK Section by Line Number	13-60
Message 5 Completes Execution	13-61

Contents

■ Execute Message 6 of the SOAK Section	13-62
Message 6 Completes Execution	13-63
■ Redisplay the APPLY Section	13-64
■ Display the PERM Section	13-65
■ Print the PERM Section	13-66
■ Execute the PERM Section	13-67
Message 8 Completes Execution	13-68
Message 9 Completes Execution	13-69
Message 10 Completes Execution	13-70
Messages 11 and 12 Complete Execution	13-71
■ Terminate the Field Update Session	13-72
Input Messages	13-73
System Generation Database	13-74
■ File-System Body Form	13-75
■ System Generation Form 2	13-77
■ Process Form	13-78
■ ULARP Form	13-79

Introduction

This chapter describes field update concepts and defines procedures for implementing updates. Field update is the software subsystem that installs changes (updates) to files and their main memory images on the 3B20D/3B21D computers.

An update affects only one file and/or its main memory image or images, although that update can effect several changes to that file at once.

Update Methods and Processes

Update Methods

There are three methods of updates:

- *File Replacement* — installs a new file on secondary storage (disk), possibly replacing an old file
- *Function Replacement* — may change global data for, adds a new source to, or replaces an old function in an executable process
- *Byte Replacement* — loads specified data values into locations in an executable process.

⇒ NOTE:

The field update mechanism does **not** apply to databases, special device files, or hardware components.

File Replacement

File replacements, as the name implies, affect only disk images. Any file replacement must be known to the system. That is, the file must have an entry in the System Generation (SG) database; and if the file is always contiguous, it must have a preallocated shadow file. (A shadow file is another “always contiguous” file that exists to preserve contiguous file space for future updates. The update file is copied into the shadow file and is then designated as the true process file.) File replacement also includes replacing an entire disk partition.

Function and Byte Replacement

Function and byte replacements affect “updatable” object files, those files that can have executable main memory images. This file type represents processes, shared libraries, or the kernel itself, and may be either killable, quasi-killable, or nonkillable. A killable process or shared library may die and leave main memory without disrupting service. A nonkillable process may not, and thus has permanent residence in memory. In general, nonkillable processes are members of the boot image file and are brought up once at boot time. Quasi-killables are monitored killables, that is, a watchdog process restarts them if they die. Function and byte replacements to nonkillables affect main memory and disk images. Function and byte replacements to killables affect only the disk images.

The Kernel Process UPDATE (KPUPDATE) will allow a killable, kernel process to have its memory and disk images updated, using function and byte replacements. This feature will allow killable kernel processes to be field updated without being terminated.

Any effect of an update is seen only when main memory is changed. Thus, a function or byte replacement to a nonkillable becomes effective once the update has been applied. A file replacement to an updatable object file becomes effective only when the new file has an image in main memory. Thus, making the file replacement changes evident requires a boot for nonkillables, killing the current invocation of a quasi-killable and reexecuting a killable. The maximum number of transfer vector overwrites (functions updated) is 50 for nonkillable processes and 25 for shared libraries.

Field Updatable Processes

Field update affects processes differently, depending on the mortality of the process. The following field update processes will be discussed:

- Nonkillable processes
- Boot image component processes
- Quasi-killable processes
- Killable processes
- Killable kernel processes
- Diagnostic processes
- Peripheral microcode.

Nonkillable Processes

Nonkillable processes are considered so vital to the system's operation that they cannot be killed without service interruption. Boot image and kernel processes fall into this category. As an example, the Memory Manager (MMGR) is a nonkillable process.

Field update handles nonkillable processes as follows:

- Update Method
 - Function replacement to the incore memory image of the process
 - Byte replacement to the incore memory image of the process.
- Multiple temporary updates may exist in a nonkillable process simultaneously.

Boot Image Component Processes

These processes are nonkillable processes which are incorporated into the boot images */appdmert* and */dmert*. Processes which comprise the boot images reside on the system in the */bootfiles* directory.

Field update handles boot image processes exactly the same as nonkillable processes. The only difference being — when updates to boot image processes are made official, the application boot image */appdmert* is regenerated incrementally by the process **isgen** – the Incremental System Generator process.

The minimum configuration boot image */dmert* is not automatically regenerated by field update. To rebuild */dmert*, use the **UPD:BLDBOOT** message.

⇒ **NOTE 1:**

Boot image should NOT be done until updates to the application boot image (*appdmert*) have been proven bootable or if there has been a dependent update between a boot product and a nonboot product.

⇒ **NOTE 2:**

Refer to the “System Generation Database” section for information regarding the System Generation (SG) database forms used by field update and system generation to build the boot image.

Quasi-Killable Processes

Quasi-killable processes are killable processes which are monitored to ensure that they will always be automatically restarted by the monitor process User-Level Automatic Restart Process (ULARP). As an example, the Display Administration Process (DAP) which handles craft terminal displays is in this category.

Field updates to these processes are applied to a temporary disk image. The process is killed by field update. ULARP restarts the quasi-killable process which loads the new temporary overwrite's disk image into main memory for field testing.

⇒ **NOTE:**

In *UNIX*® Real-Time Reliable (RTR) operating system, there is a ULARP form in the Equipment Configuration Database (ECD) for each process to be started, at boot time, and monitored by process ULARP. Refer to the “System Generation Database” section in this IP for additional information on ULARP.

Killable Processes

Killable processes are loaded into main memory only when they are requested. On completion, they give up their incore main memory space to other processes. All *UNIX* system level processes are in this category. Examples of processes in this category are diagnostics and commands, such as */bin/ps*.

Field updates to these processes:

- Usually use file replacement.
- Will not make the changes (update) until the process is terminated.

Killable Kernel Processes

The KPUPDATE processes are killable kernel processes. There are two field update methods that handle KPUPDATE processes as follows:

- Function replacement to the incore memory image of the process (if it is incore at the time of the update) as well as its disk image.
- Byte replacement to the incore memory image of the process (if it is incore at the time of the update) as well as its disk image.

Field update allows only one killable kernel update to be in the temporary state at one time.

Functionally, the update to memory and disk is handled with the same approach as if the process were nonkillable.

Diagnostic Processes

Diagnostic processes are a special case. They do not run continuously but run for a long period of time before completion. Since a diagnostic process does eventually stop, field update does not kill it, but installs an overwrite after it completes. Field updates to these processes usually use file replacement.

Peripheral Microcode

This category does not represent a process, but rather a file that is downloaded into a peripheral controller when the controller is placed in service.

Updates to these files replace the disk image. The peripheral controller must be removed and restored to service to download the new version of microcode into the controller.

Broadcast Warning Messages

The update information that is applied by the field update is contained in an update package known as a Broadcast Warning Message (BWM). The BWM is identified by an identification number of the following form:

xxxxyy-nnnn

Where: xxx = three-character prefix:
BWM — for general updates
CFT — for craft-supplied updates
DMT — for operating system only updates

yy = last two digits of the current year

nnnn = four-digit sequence number.

Thus BWM88-0015 is the fifteenth general update package for 1988.

⇒ NOTE:

The abbreviation BWM is used to identify both an update package in general and a category of update packages.

Each update is applied or installed as temporary. After a testing period, a decision is made to either “back out” the update or to make it permanent. The update must be backed out if it does not work or if it adversely affects the system. The update is made permanent when field update overwrites a temporary update disk image onto the official disk image.

The command to make an update package permanent makes each of its applied updates permanent in the order in which they were applied. When an update package is made permanent, it becomes a permanent part of the system software. The command to back out an update package backs out each of its applied updates in the reverse order in which they were applied. When an update package is backed out, its sequence number may be reused. If an update package has not been made permanent, its individual updates may be applied in accordance with the update package rules to be described.

BWM File Structure

The BWM packages are typically placed within the BWM file system */dev/bwm*. The device */dev/bwm* is mounted at boot time on a mount point named *bwm* within the */etc* directory.

Figure 13-1 represents the contents of the */etc/bwm* directory as follows:

- FILES UNDER */etc/bwm*
 - *fudb<>* — Contains the Software Demand Paging (SDP) information for accessing the field update database.
 - *fudb<>0* — Maintains the current state of the BWM as a whole and any historical information (regarding application of BWMs) required throughout the life of the generic.
 - *bwmdb<>* — Contains the SDP information for accessing the BWM database.
 - *bwmdb<>0* — Maintains, for each BWM, a record of sections and individual message lines completed.
 - *Fu_IlaErr* — Contains the latest error codes generated as a result of a fatal field update database error.
 - *PrseErr* — Contains the error codes generated as a result of parsing errors found interpreting field update input messages.
- DIRECTORIES UNDER */etc/bwm*

The BWM directories attached to *etc/bwm* may range in quantity from none to many. In the following example, the BWM directories are labeled BWMyy-aaaa through BWMyy-zzzz, with any additional BWM directories represented by bullets.

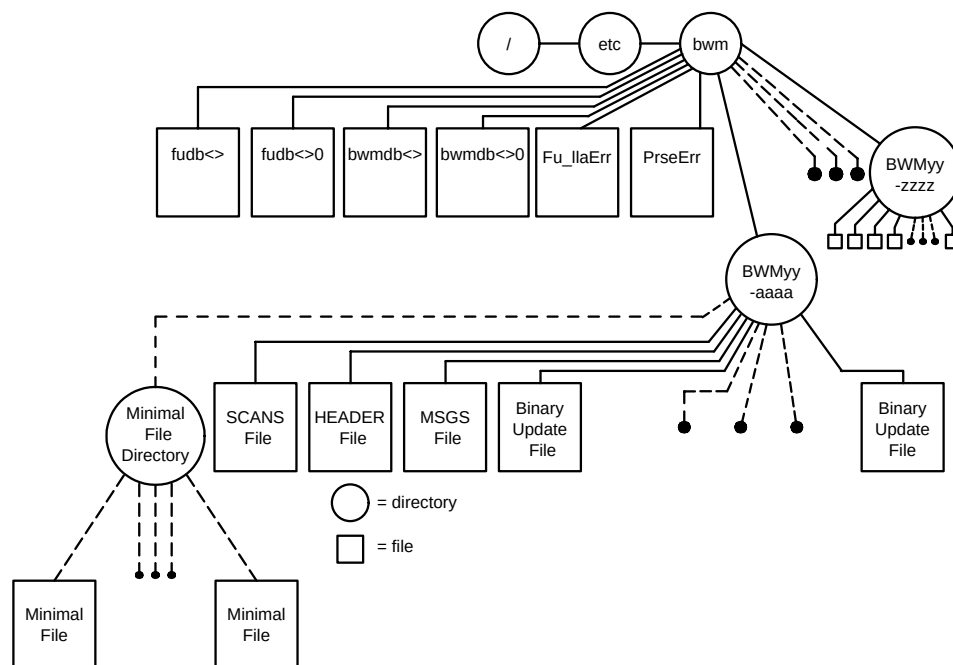


Figure 13-1. BWM File Structure

BWM Components

Each BWM package (identified by its category, year, and sequence number) consists of the following files and directory:

- The Software Change Administration and Notification System (SCANS) file contains information about the number of disk blocks required for the update package. A description of each problem addressed by the package, and its solution is also contained.
- The HDR file (BWM HEADER FILE) contains administrative and verification information about the other files within the BWM package.
- The MSGS file contains input messages to install the update package.
- Binary update files contain the actual update information to be applied to the system. They can be file replacement products as well as minimal files. If the BWM package is “informational” only, no update files will be present.
- The minimal file directory is an optional directory that contains minimal files.

SCANS File

The SCANS file contains a description of the problem being fixed and administrative information used for SCANS2 queries and distribution. The fields within the SCANS file have the following meanings (Figure 13-2):

- The BWM NUMBER field identifies this field update package.
- The SYSTEM IDENTIFICATION field identifies the target machine.
- The BWM TYPE field indicates one of the following:

GEN	Generic change
APT or APW	Auxiliary software change
INF	Information only
PAR	Parameter
LIB	Library (non-ASCII text)
BIN	Binary update files (non-ASCII text).

- The PROFILE MATCH field indicates the target machine and generic.
- The SUBJECT field defines the BWM.
- The PERSON TO CONTACT field is always the DIAGNOSTIC CENTER.
- The KEYWORDS field is usually none.
- The SPECIAL INSTRUCTIONS field is usually none.
- The GENERAL INFORMATION field is divided into four subheadings as follows:

INTRODUCTION	Describes the problem addressed by the BWM.
PROBLEM DESCRIPTION	Describes each problem addressed by the BWM and tells how the system is affected.
FIX DESCRIPTION	Describes the solution to the problem, with references to each source file affected.
TEST PROCEDURES	Is used by system test personnel to test the BWM in the lab.

- The MNEMONIC CODE field is usually none.
- The HASH SUMS field is none.
- The SOFTWARE CHANGE SIZE field identifies the total size of this BWM and each of its files in bytes and disk blocks.
- The OVERWRITE DATA identifies the type of overwrite files.

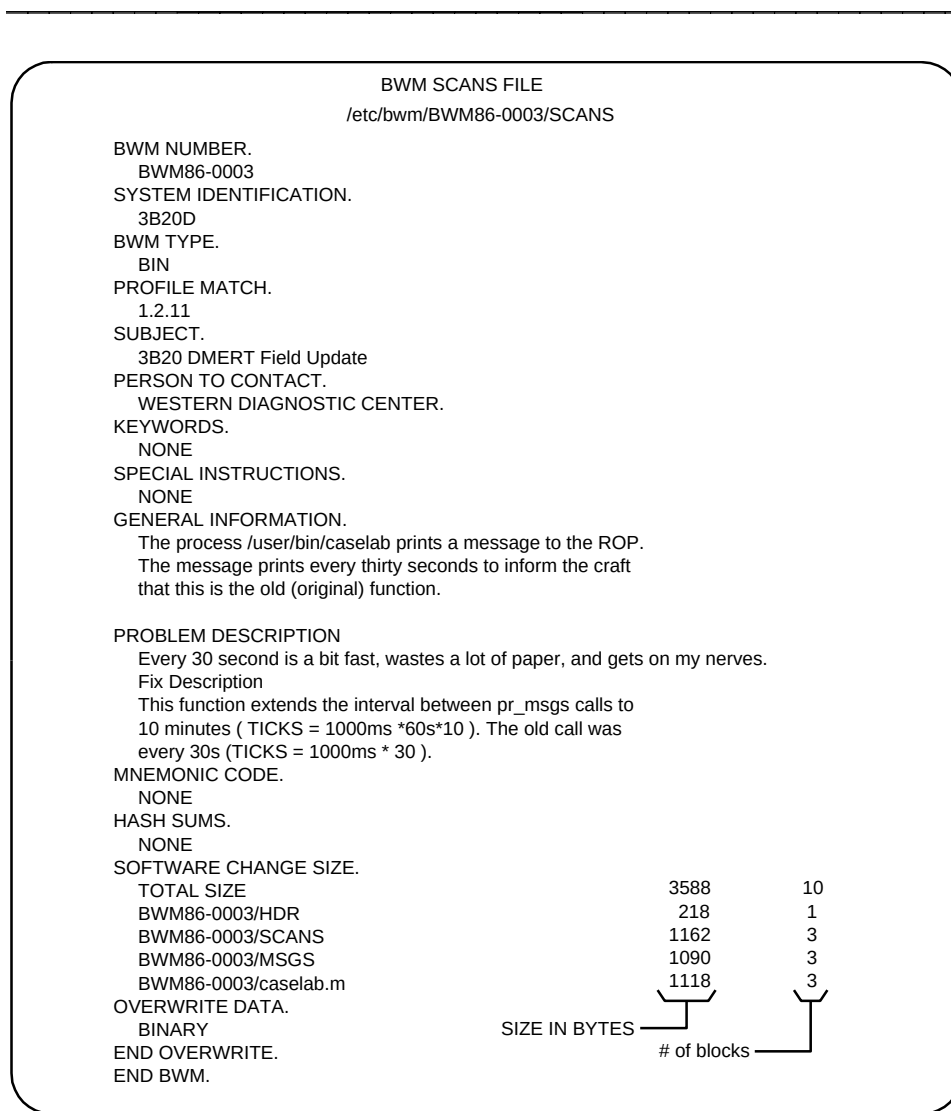


Figure 13-2. BWM SCANS File

HEADER File

The example BWM HEADER file named HDR as shown in Figure 13-3 contains information to verify the contents of the BWM package prior to its application. The fields within the HDR file have the following meanings:

- BWM identifies the BWM package.
- NOG identifies the Number of Generic (NOG) lines contained in the BWM package. The generic lines follow this line and are identified as GEN.
- GEN identifies the system and generic on the target machine for this update package.
- NOF indicates the number of update files contained in this BWM package. The MSGS and SCANS files are not included in the count.
- UFN identifies one entry for each update file in the BWM package. Each line contains the update filename followed by the file size in bytes and its hash sum, respectively.
- MSG identifies the MSGS message file and provides its size in bytes and its hash sum.
- SC2 identifies the SCANS information file and provides its size in bytes and its hash sum.

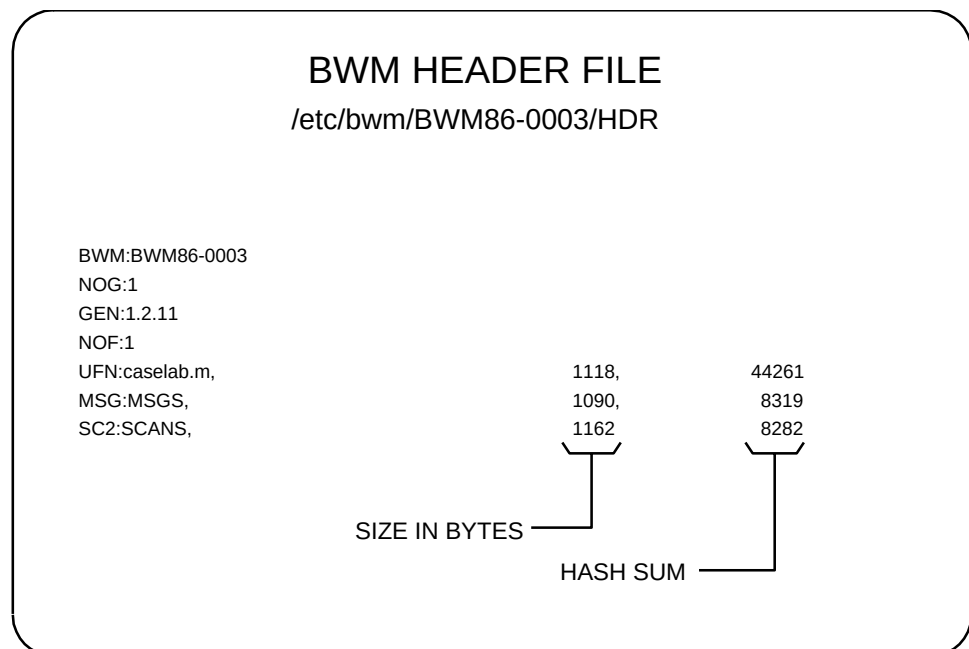


Figure 13-3. BWM HEADER File

MSGs File

The example BWM MESSAGE file named MSGS as shown in Figure 13-4 contains all the messages necessary to apply, soak, backout, and make permanent all the update files contained within the BWM package. Comment lines in the example begin with a single double quote ("). The messages are grouped into sections as follows:

- APPLY** The APPLY section contains messages to verify and apply the BWM's update files in the temporary state.
- **VERIFY MESSAGE.** The **UPD:VFY** message is always the first message in the APPLY section. Verify checks the integrity of the BWM package as loaded into the */etc/bwm* directory against the contents of the BWM's HDR file contents.
 - **APPLY MESSAGES.** The **UPD:APPLY** messages appear within the APPLY section of the message file following the **UPD:VFY** message. Only one apply message per update file within the BWM package is allowed.
- SOAK** The SOAK section contains messages and/or instructions for the testing of the BWM while in the temporary state.
- BKOUT** The BKOUT section contains messages to remove the BWM from the temporary state if the BWM should adversely affect the system. If a system boot occurred (level 52 or higher), all BWMs in the temporary state will have been automatically backed out.

NOTE:

The BWMs are backed out in the inverse order of application. That is, all BWMs in the temporary state – applied following the one requested by the backout message – will also be backed out regardless of category or sequence number. Refer to the "Update Package Rules" section in this chapter.

- PERM** The PERM section contains messages which make the BWM package permanent. Permanent BWMs are also referred to as OFFICIAL. BWMs that have been made official cannot be backed out. They are a permanent part of the system.
- **PERM MESSAGE.** The **UPD:PERM** message moves the specified BWM package from the temporary state to the permanent state. This message causes the OLD disk image of the updated files to be overwritten. The new contents are from the incore version – which was created when the update was applied in the “temporary” state.
 - **RECLAIM MESSAGE.** The **UPD:RECLAIM** message will be present in this section if any of the update files used **function replacement**. This message causes a rebuilding of the files reclaiming the address space of the replaced functions for patch space.

 NOTE:

An update to a file partition is automatically in the official state.

```

                                BWM MESSAGE FILE
                                /etc/bwm/BWM86-0003/MSGs

"~V3.1.1.1
"~VT002
"APPLY                      *****
"~C003
"
MR: 8600003
"~T004
UPD:VFY:DATA, BWM "BWM86-0003"!
"~C009
UPD:APPLY:DATA, FUNCR, UPNM= "BWM86-0003",
FN= "/usr/bin/caselab/" \
UF=("/etc/bwm/BWM86-0003/caselab.m");
"~T022
"SOAK                      *****
"~T023
"
This update should soak for 15 minutes. Verify the presents
of a change in the printout to the ROP by /usr/bin/caselab.
The frequency of the printout should change from every 30
seconds to every 10 minutes.
"
"BKOUT                      *****
"~T024
"
If the fix results in the need to reboot the system, the fix
will have been backed out automatically. If the fix does not
result in a reboot but otherwise does not work correctly. It
can be backed out by entering the command(s):
"
"~T028
UPD:BKOUT:DATA, UPNM= "BWM86-0003";
"~T039
"PERM                      *****
"~T040
UPD: PERM:DATA, UPNM= "BWM86-0003";
"~C044
UPD: RECLAIM: DATA, FN= "/usr/bin/caselab/";
```

Figure 13-4. BWM MSGS File (MML Format)

Update Files

Each update file is identified by its suffix. There are three types of BWM update files.

- A minimal object file that contains function replacement(s) for the target process. Minimal update files are identifiable by a **.m** (dot m) suffix.

This operation applies a function replacement contained in update file */etc/bwm/BWM86-0099/update.m* to process **/user/update**.

Example:

UPD:APPLY:FUNCRC,UPNM "BWM86-0099", FN

- A new object file that is a complete file replacement for the target process. File replacement update files are identifiable by the absence of any suffix. For example, **update** identifies an update file whose name is "update."

Example:

**UPD:APPLY:FILER,UPNM "BWM86-0100", FN **
"/user/update", UF "/etc/bwm/BWM86-0100/update";

This operation applies a file replacement contained in update file */etc/bwm/BWM86-0100/update* to process **/user/update**.

- A file in archive format that contains minimal files. For example, **update.a** identifies a file that contains the minimal files to be applied to a target process whose name is "update."

Example:

**"/user/update", UF "/user/update", **
UF "/etc/bwm/BWM86-0099/update.a"

Update Package Rules

The order in which BWMs are applied is important, and field update enforces the following rules:

1. Update package sequence number ordering is maintained within each prefix category (BWM, CFT, and DMT), but not across prefix category boundaries.

Sequence number ordering means that the field update software accepts only an update package with a numerically higher sequence number and year than the last active update package for that particular prefix category. (Refer to Rule 5 for an exception.) For example, suppose BWM88-0014 was the last active package and it has been made permanent. Only packages with higher sequence numbers and years within the BWM prefix category may be applied.

On the other hand, suppose that BWM88-0014 has not been made permanent (still active) and updates from the BWM88-0015 package have been applied. This action makes BWM88-0015 the LAST active update package. Therefore, field update will not accept any more updates from the BWM88-0014 package *unless* BWM88-0015 is first backed out.

This sequencing rule does not prevent the skipping of sequence numbers within a prefix category. For instance, BWM88-0017 may be applied immediately after BWM88-0015. (BWM88-0016 does not need to be applied.)

2. Update packages must be made permanent in the order they were applied.
3. The backout command backs out update packages in the reverse order they were applied. That is, the backout command (UPD=BKOUT) naming a particular update package will automatically back out any update packages applied after the one named. For example, suppose the update packages are applied in the following order.
 - BWM88-0015
 - CFT88-0002
 - CFT88-0004
 - BWM88-0016.

The command to back out CFT88-0002 would back out BWM88-0016, CFT88-0004, and CFT88-0002 in that order. An update package made permanent cannot be backed out.

4. The generic number in the HEADER file must match the generic number in the *eaiopt* form of the ECD database. Refer to 254-303-104, *Recent Change and Verify Manual*, for more information on the *eaiopt* form.
5. Field update provides a mechanism that allows “sequence number ordering” to be turned off for update packages that have not yet been made permanent.

For example, assume that an update from the BWM88-0015 package has been applied and BWM88-0015 has not been made permanent. Field update will allow an update from the BWM88-0014 package to be applied if sequence number ordering is turned off.

To turn off sequence ordering, the BWM must be applied unconditionally. (Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MLL Input Messages Manual*, for using the UCL option.)

Overriding ordering for one field update activity does not override ordering for the field update activities that follow. Ordering must be turned off for *each* field update activity where it is desired to override sequence number ordering. For example, suppose the update packages are applied in the following order:

- BWM88-0015
- CFT88-0004

- CFT88-0002
- BWM88-0016.

Applying CFT88-0002 after CFT88-0004 requires sequence number ordering override. If the update packages are applied in the order indicated in this example and it is necessary to back out CFT88-0004, then BWM88-0016, CFT88-0002, and CFT88-0004 must be backed out in that order.

Carrying this example further, assume that CFT88-0003 had been applied at some time in the past. This prevents reusing the CFT88-0002 sequence number even though it has been backed out, unless ordering override is again requested. Therefore, the next CFT sequence number available for use is CFT88-0004.

6. Changes to the BOOT IMAGE.

- If the APPLY section installed changes to boot files, executing the PERM will automatically rebuild the application boot image */appdmert* within the section, execute it manually following the completion of the PERM section.
- If the update affected the application boot image, the */dmert* boot image will also need to be rebuilt. To apply this change, construct and enter manually the appropriate **UPD:BLDBOOT** message. This message should be input ONLY after the changes to the application boot image have been verified as bootable or if there has been a dependent update between a boot product and a nonboot product.

⇒ NOTE:

The updating of the */dmert* boot image is the operator's responsibility and will never appear as part of the MSGS file.

7. Reclaiming PATCH SPACE.

If the update applied function replacements, then the PERM section of the MSGS file should contain the **UPD:RECLAIM** input message in the *p* file.

8. Following the completion of the PERM section in the MSGS file, the BWM package may be removed from the system via the **UPD:CLR** input message.
9. Each **UPD:APPLY** input message can contain a maximum of 10 minimal files for function replacement.
10. The maximum number of transfer vector overwrites (functions updated) is 50 for nonkillable processes and 25 for shared libraries. The maximum length of a **UPD:APPLY** input message is 256 characters.
11. There is no limit on the number of **UPD:APPLY** input messages per BWM, but disk resources may prevent specifying too many.

Update Package States

Figure 13-5 shows the states an update package moves through during a typical update session. The BWM database states are shown in parentheses.

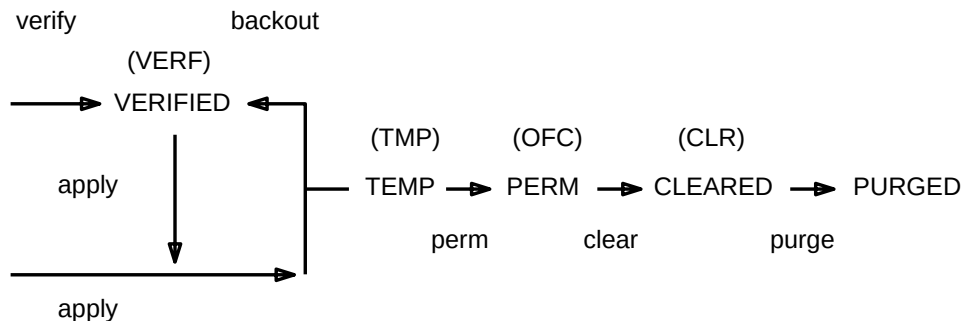


Figure 13-5. Update Package States

In general, the application of a BWM package is as follows:

- EXECUTE the APPLY section of the BWM MSGS file.
 - The VERIFY MESSAGE (from UPD:VFY) will be the first message in the section and must be completed successfully before additional execution of the APPLY section will continue.
 - Once the APPLY section is completed, the BWM package will be in the temporary state ready for on-site testing. Test procedures and the length of the soak interval are given in the SOAK section of the MSGS file. Additional information is contained in the SCANS file.
- EXECUTE the SOAK section of the BWM.



NOTE:

This section may not have any messages to execute depending on the fix and files affected.

- EXECUTE the BKOUT section of the BWM MSGS file if the BWM needs to be backed out.
- EXECUTE the PERM section of the BWM MSGS file. This will make the BWM a permanent part of the generic and cannot be backed out.
 - If a boot occurs during the SOAK period, the BWM package is automatically backed out.
 - If the BWM is suspected of causing a system problem prior to making it permanent, the BKOUT section may be implemented.

Typical Update Package States

The following list defines the states an update package moves through during a typical update session.

- **Verified** — The verify operation confirms the appropriate files exist in the update package directory, and that they are in the proper format. Ordinarily, verification of the update package is done before any other operation. This state applies only to the update package as a whole.
- **Temporary** — Changes to a running system are suspected until they are proven not to adversely affect service. Therefore, updates are applied in a temporary state (except as noted) so they may be removed if judged unsatisfactory. The one exception to an update being applied in a temporary state is an update to a file partition, for which there is no mechanism for a temporary state. Updates, except file partition updates, can be removed (backed out) by explicit command. All updates to main memory have an interval called the “wait time” during which any Phase 1 fault will cause the update to be removed. The default is 5 minutes, but the time interval can be changed by the WTIME data field in the update input message.
 - An update package is temporary if it contains only *temporary* or *temporary and omitted* updates. Such an update package may be backed out, made permanent, or have more updates added to it.
 - The temporary state can apply either to a update package as a whole or to individual updates.
- **Permanent** — Once the wait time for an update has expired and the changes are effective and not harmful, then the update may be made a permanent part of the system. Also, if you make a function replacement to a nonkillable process permanent, you permanently change the disk image of that process. If this process is part of the boot image, then the application boot image is automatically rebuilt. An update package is permanent if it contains only permanent or permanent and omitted updates. Such an update package is inactive.

- **Cleared** — The cleared state means that the update package files have been removed from the update package directory. An update package may be cleared only if it is permanent and all of its component function replacement updates have been reclaimed. The cleared state applies only to the update package as a whole.
- **Purged** — The purged state means that all history of the update package has been removed from the field update database. An update package may be purged only after it has been made permanent and its updates have been reclaimed. The purged state applies only to the update package as a whole.

Other Update Package States

The following list defines other states that may be encountered during field update. All of these states can apply to an update package as a whole or to individual updates.

- **Omitted** — An update package is omitted if all its updates are omitted. Such an update package is considered inactive, and updates may not be applied against it. The omitted state is not used typically unless a trouble condition occurs.
- **Reclaimed** — For function replacement, field update uses portions of the updated file or main memory image to effect the update. Once these portions and the replaced functions are no longer needed, the space they occupy may be reclaimed for later use. Only those files that have permanent function replacements may be reclaimed.
- **Pending** — If an update operation fails, that update is in a pending state. A pending update must be resolved before any other update operations can take place.

In the order they should be attempted, the following options are available for dealing with pending updates.

- CONTINUE the operation. Reinvoking the update command with the data keyword **CONT** will pick up the operation from the point of failure.
- RESET the operation. The command **UPD:RESET** will take the operation back to its sane state where the command can be reinvoked or backed out. If the APPLY operation fails, it should try to be continued. If the continue fails, it should be RESET and the apply command reexecuted. If a PERM operation fails, it should be continued. If the continue fails, it should be RESET back to a sane state and then reexecuted. If the reexecution fails, the update should again be RESET, backed out, and then reapplied.

- BACKOUT the update. The command **UPD:BKOUT:CONT** backs out all pending temporary and temporary updates for a specified update package or all pending temporary and temporary updates in the system.
- OMIT the update. The command **UPD:OMIT** will mark the update as omitted, removing the pending state from the database but not undoing any changes the interrupted update may have made. An OMIT changes the definition of the update package by removing an update from the update package. If an update package contains one omitted update and the rest are applied updates, the update package is still considered to be in the applied state.

Note that a system initialization (boot) is the equivalent of a backout in that temporary and pending temporary updates are removed from the system and the field update database.

Furthermore, pending permanent updates are omitted to prevent continuing and completing the making of the update as permanent. This eliminates any risk of the memory and disk images being mismatched should a boot occur during the **PERM** request.

Messages Files Execution and Display

The field update page provides, via poke commands, field update functions. The Broadcast Warning Message Initialization (BWMINT) process drives the field update page and provides automatic execution and display of a BWM message file. The BWMINT process is entered by inputting command **109** at the Maintenance Teletypewriter (MTTY). BWMINT attaches to the BWM database (*/etc/bwm/bwmdb*) or creates the *bwmdb* if one does not exist.

The user has a choice in determining where the field update database and BWM databases are to be created and accessed. To exercise this capability, the user needs to specify the directory pathname in which the databases will reside. The pathname is called */etc/fubddir*. The file may need to be created. The default path will be used if any of the following is true:

- There is no *fubddir* file under *etc*.
- The *fubddir* file is not readable or cannot be accessed.
- The pathname specified in the *fubddir* file is not a name of a directory, does not exist, or cannot be accessed for reading and writing.

Although it is possible to apply the updates by using input messages, the automatic action of the BWMINT process makes field updates user friendly and more efficient. Refer to the “Input Messages” and “Field Update Commands Execution Locations” sections for more information on field update input messages

Field Update Page

The field update page (Figure 13-6) is brought up on the operators terminal by entering command code **109**. The page contains the following parts.

- *Area A — Header*
The first two lines contain system status.
- *Area B — Command Entry*
The command entry line (labeled CMD<) is the line where menu numbers are entered.
- *Area C — Menu Indicators*
These four lines contain the menu numbers and names of the commands for the field update page.
- *Area D — Communication*
This line is reserved for error messages and informative messages that are output by the field update processes and menu page.

- **Area E — MSGS Display**
A 12-line area where portions of the MSGS file are displayed or the currently executing section is displayed.
- **Area F — Scrolling Area**
Where output messages are displayed.
- **Area G — Manual Input Line**
The bottom line where input messages may be entered. Note the “less than” sign is shown in this area. This line is also used as a scrolling line when input messages are not being entered.

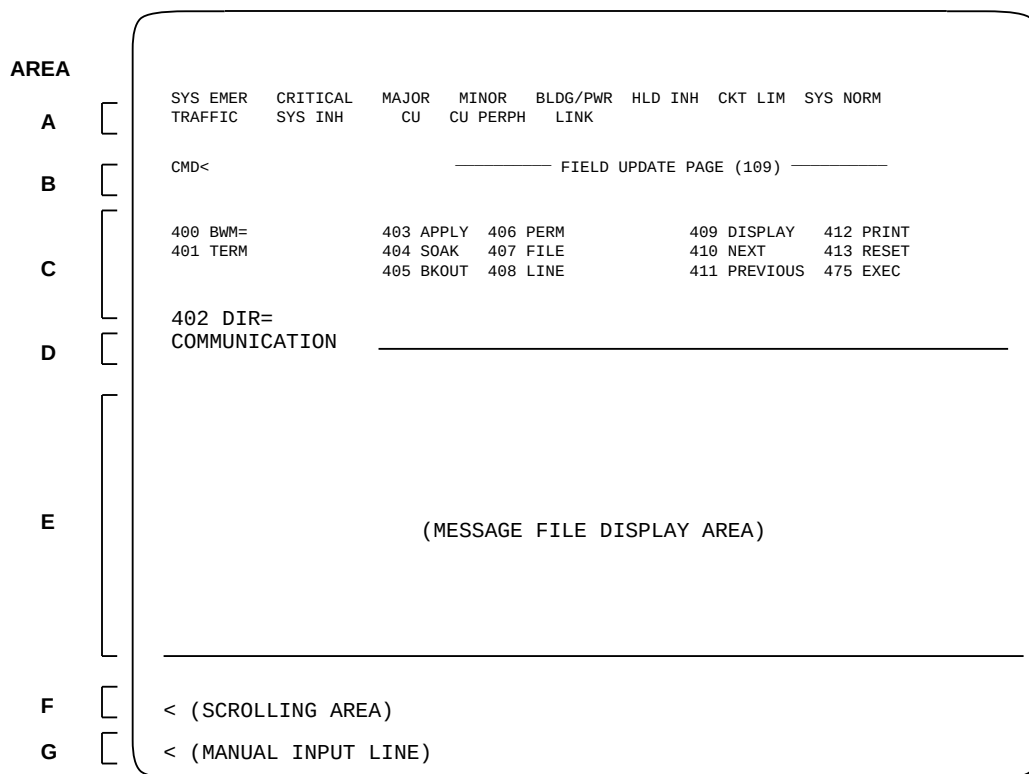


Figure 13-6. Field Update Page

Menu Indicators

The MSGS files may be displayed and input messages may be executed by the selection of menu numbers corresponding to the menu indicators. There are three classes of indicators: configuration, scope, and action.

Configuration Indicators

Configuration indicators request global actions associated with an update session. They also locate the MSGS file for the BWMINT process. The configuration indicators and a brief description of their uses are listed as follows:

400 BWM= Used to specify the directory name relative to the */etc/bwm* file partition containing the messages file, SCANS file, header file, minimal files for function replacements, and update files for file replacements.

401 TERM Used to terminate the update session.

⇒ NOTE:

Poking **401** before executing the correct BWM displayed leaves the BWM in the current state. Since the **401** poke command will terminate the BWMINT process, use **401** poke command to leave the field update page rather than poking in the page number at the CMD< line.

402 DIR= Used to specify an alternate directory, other than the one relative to */etc/bwm*, that contains the MSGS file and all other files. This must be an absolute pathname to the directory containing the BWM. The default directory is */etc/bwm*.

Scope Indicators

Scope indicators restrict the action of the BWMINT process to particular portions of the MSGS file or to the entire MSGS file. The scope indicators and a brief description of how they affect the BWMINT process is as follows:

403 APPLY Limits BWMINT to execute the APPLY section only
404 SOAK Limits BWMINT to execute the SOAK section only
405 BKOUT Limits BWMINT to execute the BKOUT section only
406 PERM Limits BWMINT to execute the PERM section only
407 FILE Executes the entire MSGS file
408 LINE Executes a particular input message (LINE) in the MSGS file.

Action Indicators

The action indicators name the actions the BWMINT process is to perform. These indicators and a brief description of the actions they represent are listed as follows:

409	DISPLAY	Displays the specified portion of the MSGS file
410	NEXT	Displays the next portion of a section in the MSGS file
411	PREVIOUS	Displays the previous portion of a section in the MSGS file
412	PRINT	Prints a portion of the MSGS file on the ROP
413	RESET	Performs the RESET operation on the failed input message
475	EXEC	Executes input messages in the MSGS file.

Key Points About Menu Indicators

Usually an update session consists of the following:

- A configuration indicator (either **400** or **402**)
- Pairs of scope-action indicators
- A final configuration indicator (**401**).

During a session, the operator may go to another display page by entering that page number at the CMD< line. The operator may then enter **109** at the CMD< line to return to the update session. Pressing the **CMD MSG** key takes the cursor to the manual input line where input messages may be entered. Pressing the **PF3** key returns the cursor to the CMD< line.

⇒ NOTE:

Update messages may themselves be entered outside the update session. However, except for the **UPD:DISPLAY** message, this practice is dangerous because the BWM database will be out of sync with the field update actions performed and is **not** recommended. The BWMINT process keeps track of the executed input messages in a MSGS file through the BWM database and the field update database. This process is unaware of any messages executed in the manual input line.

Update commands initiated by menu number 475 will continue to execute no matter where the cursor is positioned. The following menu indicators require text input:

400	The name of the BWM
402	The full pathname of the directory
408	The message file line number.

The text may be entered by moving the cursor, via the arrow keys, to the menu indicator and typing the text. Alternately, the text may be entered on the CMD< line by entering the menu indicator, a comma, and then the text; for example, **400, BWM85-0401**.

The 409 menu indicator displays a portion of the MSGS file without executing it. The 475 indicator executes a section of the MSGS file, displaying the portion that is executing. The input messages of the MSGS file are numbered sequentially by the BWMINT process. The BWMINT process indicates status as follows:

- Unexecuted messages are in normal video.
- Successfully executed messages are in reverse video.
- Presently executing or failed messages blink.

The notice of successful completion of a section or the termination notice of an unsuccessful message is displayed on the communication line. All output messages appear in the scrolling area and on the ROP.

The following rules apply to the scope indicators when executing the MSGS file sections. Note that executing a section means executing that section's input messages in the order they appear.

1. The PERM section cannot be executed (**406 475**) until the entire APPLY section has been executed.
2. Once the PERM section has been executed (**406 475**), neither the APPLY section nor the BKOUT section can be executed.
3. Once a BKOUT section has been executed or a boot has occurred, the APPLY section can be reexecuted.

Line execution (**408 475**) allows the operator to execute messages one at a time in ascending numerical order. Messages may not be skipped. Once a message in the APPLY section has been executed by line number, it may not be reexecuted until the BKOUT section has been executed.

A message that failed may be reset (**408 413**). If the message cannot be reset, it must be backed out or omitted via manual execution; for example, **UPD:BKOUT** or **UPD:OMIT** on the CMD< line.

A MSGS file cannot be executed as a file. It must be executed one section or line at a time. The RESET operation can only be performed on one line number; for example, **408 413**. It cannot be performed on an entire section or the entire file.

Allowable Scope/Action Combinations

Table 13-1 shows the allowable scope/action combinations for the field update page.

Table 13-1. Allowed Scope/Action Combinations

Scope	Action	Effect
APPLY	DISPLAY	The APPLY section, or the first 12 lines of the APPLY section, is displayed on the screen.
SOAK	DISPLAY	The SOAK section, or the first 12 lines of the SOAK section, is displayed on the screen.
BKOUT	DISPLAY	The BKOUT section, or the first 12 lines of the BKOUT section, is displayed on the screen.
PERM	DISPLAY	The PERM section, or the first 12 lines of the PERM section, is displayed on the screen.
FILE	DISPLAY	The entire MSGS file, or the first 12 lines of the MSGS file, is displayed on the screen.
APPLY	EXEC	The APPLY section messages are executed. If the entire APPLY section does not fit in the MSGS area, the portion of the APPLY section displayed will shift to include the message being executed.
SOAK	EXEC	The SOAK section messages are executed. If the entire SOAK section does not fit in the MSGS area, the portion of the SOAK section displayed will shift to include the message being executed.
BKOUT	EXEC	The BKOUT section messages are executed. If the entire BKOUT section does not fit in the MSGS file area, the portion of the BKOUT section displayed will shift to include the messages being executed.
PERM	EXEC	The PERM section messages are executed. If the entire PERM section does not fit in the MSGS file area, the portion of the PERM section displayed will shift to include the messages being executed.

Table 13-1. Allowed Scope/Action Combinations (Contd)

Scope	Action	Effect
LINE	EXEC	The message number input to the LINE indicator message is executed. If that line is not in the MSGS display area, the display will shift to include the message.
LINE	RESET	A reset operation is initiated for the incomplete APPLY or PERM message identified by the message number input to the LINE indicator.
APPLY	PRINT	The APPLY section is printed on the ROP.
SOAK	PRINT	The SOAK section is printed on the ROP.
BKOUT	PRINT	The BKOUT section is printed on the ROP.
PERM	PRINT	The PERM section is printed on the ROP.
FILE	PRINT	The entire MSGS section is printed on the ROP.
APPLY	NEXT	If the remainder of the APPLY section is not in the MSGS display area following an APPLY DISPLAY or an APPLY NEXT, the next portion of the APPLY section is displayed on the screen.
SOAK	NEXT	If the remainder of the SOAK section is not in the MSGS display area following a SOAK DISPLAY or a SOAK NEXT, the next portion of the SOAK section is displayed on the screen.
BKOUT	NEXT	If the remainder of the BKOUT section is not in the MSGS display area following a BKOUT DISPLAY or a BKOUT NEXT, the next portion of the BKOUT section is displayed on the screen.
PERM	NEXT	If the remainder of the PERM section is not in the MSGS display area following a PERM DISPLAY or a PERM NEXT, the next portion of the PERM section is displayed on the screen.
FILE	NEXT	If the remainder of the MSGS file is not in the MSGS display area following a FILE DISPLAY or a FILE NEXT, the next portion of the MSGS file is displayed on the screen.

Table 13-1. Allowed Scope/Action Combinations (Contd)

Scope	Action	Effect
APPLY	PREVIOUS	If the initial portion of the APPLY section is not in the MSGS display area following an APPLY NEXT or an APPLY PREVIOUS, the previous portion of the APPLY section is displayed on the screen.
SOAK	PREVIOUS	If the initial portion of the SOAK section is not in the MSGS display area following a SOAK NEXT or a SOAK PREVIOUS, the previous portion of the SOAK section is displayed on the screen.
BKOUT	PREVIOUS	If the initial portion of the BKOUT section is not in the MSGS display area following a BKOUT NEXT or a BKOUT PREVIOUS, the previous portion of the BKOUT section is displayed on the screen.
PERM	PREVIOUS	If the initial portion of the PERM section is not in the MSGS display area following a PERM NEXT or a PERM PREVIOUS, the previous portion of the PERM section is displayed on the screen.
FILE	PREVIOUS	If the initial portion of the MSGS file is not in the MSGS display area following a FILE NEXT or a FILE PREVIOUS, the previous portion of the FILE is displayed on the screen.

Field Update Command Execution Locations

Below is a summary of field update commands arranged by the location from where the command may be executed.

Scrolling Area Only Commands

The following is a list of field update commands that cannot be performed via poke commands. These messages will typically not appear as a command line within a MSGS file section. In such cases, these messages must be input manually.

UPD:DISPLAY	Displays information about the current state of BWMs as contained within the field update database.
UPD:FTRC	Lists function information from a <i>pfile</i> symbol table.

UPD:OMIT	Removes a pending update from the update package. Should be used only on updates that cannot be reset.
UPD:PURGE	Removes the specified BWM records from the field update database.

Scrolling Area/Poke Messages

The following message will NOT appear within a BWM MSGS file. This message may be input manually or poked in.

UPD:RESET	Resets an update's status to a previously acceptable state. This command may be executed via Poke 413 of Page 109.

⇒ NOTE:

Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for a complete listing of the input messages and their options.

Message File Commands

The MSGS file of input messages may contain many other messages not listed in the following sample. For more information, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

The following messages may appear within the MSGS file of a BWM.

UPD:BKOUT	Causes the specified BWM and all subsequent BWMs in the temporary state to be restored to their original state.
UPD:BLDBOOT	Rebuilds the boot image (either <i>appdmert</i> or <i>dmert</i> depending on input parameters).
UPD:APPLY:BYTER	Applies a byte replacement update to a process.
UPD:APPLY:FILER	Applies a file replacement update to a process.
UPD:APPLY:FUNCR	Applies a function replacement update to a process.
UPD:CLR	Clears the specified BWM update package from the field update directory <i>/etc/bwm</i> .
UPD:PERM	Makes the specified temporary update package permanent.
UPD:RECLAIM	Reclaims unused patch space in function replaced <i>pfiles</i> .
UPD:VFY	Verifies the consistency of the specified BWM package.

Update Display Command

The **UPD:DISPLAY** command is a tool for routine examination of the current state of BWMs as recorded in the *fudb* database. The following options are available to restrict the amount of information reported by the **UPD:DISPLAY** message:

- By TYPE of update – function replacement, byte overwrite, and so forth
- By CATEGORY – (BWM, CFT, and DMT)
- By STATUS – PERM, TMP, VERF, and so forth
- To LAST – last update with the given characteristics
- To TIME – earliest update time to be reported
- VLEV – provides additional information, input as a two-digit number separated by a comma; for example,
UPD:DISPLAY:VLEV(0,2);
Verbose level “2” provides the greatest amount of output for this input message.

For more information on verbosity levels, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Troubleshooting Hints

This section is a general guide for using field update commands in troubleshooting field update problems. Of course, technical support should be sought before any corrective measures are taken, if applicable. Usually, the information on the field update page will be sufficient to guide you through the installation of a BWM. However, provisions are available when a particular step fails.

- If a critical point has not been reached, a failed step may be corrected by resetting the operation to the previous state.
- If a critical point has been reached, the state will be Pending Temp (PTMP).

The possible steps in the order they should be attempted are as follows:

1. Examine the output messages associated with the failed operation to determine what problem the field update process is reporting. Resolve the source of the fault.

2. Determine the current state of the BWM by using the following:

UPD:DISPLAY:VLEV(0,2);

- If the current state is not PTMP, reenter Page 109 and execute the section again.
- If the current state is PTMP, then:
 - Continue the operation. Reinvoke the update command from the input message section with the key word **CONT**. This will pick up the operation from the point of failure.
 - If that fails, attempt to reset the operation using the **UPD:RESET** command. This takes the operation back to its starting state. Reset performs no update operation and may leave temporary files created as part of the update.
 - If that fails, back out the update using the **UPD:BKOUT:CONT** message. This will back out an update in the PTMP state.
 - Finally, and only with support approval, the update can be omitted, using the **UPD:OMIT** message. This will remove the pending state from the database, but not “undo” any changes that may have taken place. An OMIT changes the definition of the BWM by removing an update (one of the apply messages) from the BWM.

Verification of Proper Installation

Occasionally, it may be important to verify that a particular update has been applied to a file.

FILE REPLACEMENTS can be verified by checking time stamps and file size on the target process by inputting:

For MML, **OP:STATUS:DATA, LISTDIR, DN="path";**

For PDS, **OP:STATUS:LISTDIR, DN="path"!**

Then, compare the results to those of the update file in the */etc/bwm* directory for the BWM.

For FUNCTION REPLACEMENTS, the new address of the function is printed during execution of the **UPD:APPLY** input message by the field update processes as follows:

NEW FUNCTION NAME	ADDRESSES ADDRESS
<i>funcname</i>	<i>0xnewaddr</i>

Where *funcname* is the name of the function being replaced or added and *0xnewaddr* is the address of the function in hexadecimal.

The new address can be used to set up a GRASP breakpoint to determine if the function is accessed.

For more information on GRASP, refer to 254-303-107, *Software Troubleshooting Guide*.

The **UPD:FTRC** command can be used to identify the following about a function replacement:

- TV address of the replaced function
- Text address of the new function
- Size for the function
- Decision function address.

Field Update Session Example

This section describes a field update session example. This example tries to include many of the elements that may occur during “live site” field update sessions (including some errors). Therefore, all portions of this example probably would not be encountered during any one typical update session.

⇒ NOTE 1:

Field update error messages are contained in the UPD section of 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

⇒ NOTE 2:

The input messages used in this field update session example are Man-Machine Language (MML). For more information, refer to 254-303-112, *MML Input Messages Manual*.

Before getting into the example with its displays of the Field Update Page, it may be helpful to briefly review the following key steps of the example:

1. Call up the Field Update Page 109.
2. Display the APPLY section. (This fails because a specific BWM has not been named.)
3. Enter the BWM number.
4. Display the APPLY section.
5. Display Page 2 of the APPLY section.
6. Display the PERM section.
7. Execute the PERM section. (This fails because the APPLY section has not been executed.)
8. Execute the APPLY section. (For example, messages 1, 2, and 3 complete successfully while message 4 fails because of “built-in” error.) Reset the failed message (after correcting the error that caused message 4 to fail).
9. Reexecute the APPLY section.
10. Execute the BKOUT section.
11. Redisplay the APPLY section.
12. Rexecute the APPLY section.
13. Display the SOAK section.

14. Execute the SOAK section by line number.
15. Execute message 6 of the SOAK section.
16. Redisplay the APPLY section.
17. Display the PERM section.
18. Print the PERM section.
19. Execute the PERM section.
20. Terminate the field update session.

For clarity, the following conventions are used in the screen displays of the Field Update Page. Operator-entered data is shown in bold. [Note that when two command codes must be entered on the CMD< line, the **Enter** key (designated <CR>) must be operated after each code.] Reverse video is shown as white print on a black background. Blinking data is shown with a light gray overlay.

While messages are executing, the message and any corresponding warnings and/or errors will be displayed on the ROP.

Call Up the Field Update Page

To start our example field update session, call up the Field Update Page by entering command code **109** on the CMD< line. The system responds by displaying the the following Field Update Page:

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU        CU PERPH  LINK
CMD<                                FIELD UPDATE PAGE(109)
400 BWM=                403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION _____
```

(Message File Display Area)

< (Scrolling Area)

Display the APPLY Section

Display the APPLY section by entering **403**<CR> then **409**<CR> on the CMD< line. Note that this generates an error because the BWM number has not been entered. The error message appears on the COMMUNICATION line.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU      CU PERPH  LINK
CMD< 403 409 OK                                FIELD UPDATE PAGE(109)
400 BWM=                Y 406 PERM                Y 412 PRINT
401 TERM                404 SOAK 407 FILE            410 NEXT 413 RESET
                        405 BKOUT 408 LINE          411 PREVIOUS 475 EXEC
402 DIR=
ER
```

Enter the BWM Number

Enter the BWM number (directory name) by entering **400,CFT88-0001**<CR> on the CMD< line. The BWM number specifies the set of updates to be used.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU       CU PERPH  LINK
CMD< 400,CFT88-0001 OK          FIELD UPDATE PAGE(109)
01  403 APPLY 406 PERM          409 DISPLAY 412 PRINT
401 TERM    404 SOAK 407 FILE          410 NEXT  413 RESET
          405 BKOUT 408 LINE          411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
```

Display the APPLY Section

Display the APPLY section by entering **403<CR>** then **409<CR>** on the CMD< line. The first page of the APPLY section is displayed in the message file display area.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH      CU        CU PERPH  LINK
CMD< 403 409 OK                                FIELD UPDATE PAGE(109)

400 BWM=CFT88-0001      Y 406 PERM      Y 412 PRINT
401 TERM                404 SOAK  407 FILE      410 NEXT  413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR:d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/sknnt.p
"      with the minimal file nt21a.m.
"
1  UPD: VFY:DATA , BWM = "CFT88-0001";
"
2  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt11c.m";
```

➤ NOTE:

The **UPD:VFY** command is a part of the APPLY section.

Display Page 2 of the APPLY Section

Display the second page of the APPLY section by entering **403<CR>** then **410<CR>** on the CMD< line. The second page consists of the remainder of the APPLY section commands.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU       CU PERPH  LINK
CMD< 403 410 OK                                FIELD UPDATE PAGE(109)
400 BWM=CFT88-0001      Y 406 PERM              409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE              XT  413 RESET
                        405 BKOUT  408 LINE              411 PREVIOUS  475 EXEC
402 DIR=
COMMUNICATION
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt13c.m";
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt21a.m";
   "
```


Display the PERM Section

Display the PERM section by entering **406**<CR> then **409**<CR> on the CMD< line. The PERM section is displayed in the message file display area (messages 8, 9, 10, 11, and 12).

```
SYS EMER    CRITICAL    MAJOR    MINOR    BLDG/PWR    HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH      CU       CU PERPH   LINK
CMD< 406 409 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT-0001      403 APPLY  RM          Y  412 PRINT
401 TERM              404 SOAK   407 FILE    410 NEXT  413 RESET
                   405 BKOUT  408 LINE    411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"PERM. *****
"
8  UPD: PERM:DATA , UPNM = "CFT88-0001";
"
9  UPD: RECLAIM:DATA , FN = "/fldupd/sndnt.p";
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Execute the PERM Section

Execute the PERM section by entering **406<CR>** then **475<CR>** on the CMD< line. This represents an attempt to execute the PERM section out of order, which is illegal. Note the error message on the COMMUNICATION line. Since this is an illegal request, field update does not honor it.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE(109)
400 BWM=CFT88-0001      403 APPLY      RM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK      407 FILE      410 NEXT    413 RESET
                        405 BKOUT     408 LINE      411 PREVIOUS EC
402 DIR=
N
"PERM. *****
"
8  UPD: PERM:DATA , UPNM = "CFT88-0001";
"
9  UPD: RECLAIM:DATA , FN = "/fldupd/sndnt.p";
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Execute the APPLY Section

Execute the APPLY section by entering **403**<CR> then **475**<CR> on the CMD< line. Since this is a legal request, field update begins executing the first message. While field update is executing a message, the message is shown blinking.

```

SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH   CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE(109)
400 BWM=CFT88-0001      Y 406 PERM              409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE              410 NEXT    413 RESET
                        405 BKOUT  408 LINE              411 PREVIOUS  EC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR:d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/sknt.p
"      with the minimal file nt21a.m.
"
"
2  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt11c.m";

```

Message 1 Continues to Execute

When field update completes execution of a message, the message is displayed in reverse video. Page 1 of the APPLY section on the Field Update Page shows Message 1 in reverse video. When message 1 completes execution, field update begins executing message 2 automatically.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU        CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE(109)

400 BWM=CFT88-0001      [REDACTED] Y 406 PERM          409 DISPLAY    412 PRINT
401 TERM                404 SOAK    407 FILE          410 NEXT      413 RESET
                        405 BKOUT    408 LINE          411 PREVIOUS  [REDACTED] EC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR:d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/sknt.p
"      with the minimal file nt21a.m.
"
[REDACTED];
"
2  UPD: APPLY:DATA , FUNCN, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
"
```

Message 2 Continues to Execute

When message 2 completes execution, Page 2 of the APPLY section is displayed automatically on the Field Update Page. Then, the field update automatically begins to execute message 3. Message 3 is then shown blinking to indicate that message 3 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH      CU        CU PERPH   LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS  EC
402 DIR=
COMMUNICATION
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt21a.m";
   "
```

Message 3 Continues to Execute

When message 3 completes execution, message 3 is displayed in reverse video on the Field Update Page. Then, the field update automatically begins executing message 4.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH   CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT 408 LINE          411 PREVIOUS EC
402 DIR=
COMMUNICATION
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
   "
"
```

Message 4 of the APPLY Section Fails

A failure is encountered while field update is executing message 4. Note the error message on the COMMUNICATION line. Message 4 continues to be displayed blinking.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU        CU PERPH   LINK
CMD< 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS 475 EXEC
402 DIR=
ED
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
"
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/skntt.p"?
"
"
```

Reset the Failed Message

At this point in the example, the error that caused message 4 to fail is corrected (the missing minimal file *nt13c.m* was added to the CFT88-0001 directory). Reset the failed message by entering **408,4**<CR> then **413**<CR> on the CMD< line. A reset restores the *pfile*, database, and field update to the previous sane state. Message 4 continues to blink until the reset request has completed, then message 4 is displayed in normal video as shown.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU      CU PERPH  LINK
CMD< 408,4 413 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT      █
402 DIR=                405 BKOUT  █=4              411 PREVIOUS  475 EXEC
COMMUNICATION

3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt21a.m";
   "
```


Reexecute the APPLY Section

Reexecute the APPLY section by entering **403<CR>** then **475<CR>** on the CMD< line. Field update automatically displays the second page of the APPLY section. Messages 1, 2, and 3 (when displayed) are in reverse video, indicating that these messages have been executed. Message 4 is then shown blinking to indicate that message 4 is currently executing.

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR CU	MINOR CU PERPH	BLDG/PWR LINK	HLD INH	CKT LIM	SYS NORM
CMD< 403 475 OK				FIELD UPDATE PAGE (109)			
400 BWM=CFT88-0001		Y	406 PERM		409 DISPLAY	412 PRINT	
401 TERM		404 SOAK	407 FILE		410 NEXT	413 RESET	
		405 BKOUT	408 LINE		411 PREVIOUS	EC	
402 DIR=							
COMMUNICATION							
3 UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?							
"							
4 UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?							
"							

APPLY Section Completes Execution

When field update finishes executing message 4, message 4 is shown in reverse video. The SECTION COMPLETED message on the COMMUNICATION line means that all of the APPLY section messages have been executed.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS  EC
402 DIR=
      D
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
      "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
      "
      "
```

Execute the BKOUT Section

Execute the BKOUT section by entering **405<CR>** then **475<CR>** on the CMD< line. Message 7 is then shown blinking to indicate that message 7 is currently executing. Execution of the BKOUT section is shown as follows to demonstrate the capability. Normally, the BKOUT section is not executed routinely.

```
SYS EMER    CRITICAL    MAJOR    MINOR    BLDG/PWR    HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU        CU PERPH   LINK
CMD< 405 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        [REDACTED]T  408 LINE          411 PREVIOUS [REDACTED]EC
402 DIR=
COMMUNICATION
" BKOUT. *****
"      If the fix results in the need to reboot their system, the fix will
"      have been backed out automatically. If the fix does not result
"      in a reboot but otherwise does not work correctly, it can be backed
"      out by entering the command[s]:
"      [REDACTED] ;
"
```

BKOUT Section Completes Execution

When field update has completed executing the BKOUT section, message 7 is displayed in reverse video and the SECTION COMPLETED message appears on the COMMUNICATION line.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 405 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM      409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT    413 RESET
                        [REDACTED]T  408 LINE      411 PREVIOUS [REDACTED]EC
402 DIR=
[REDACTED]D
"BKOUT.*****
"      If the fix results in the need to reboot their system, the fix will
"      have been backed out automatically. If the fix does not result
"      in a reboot but otherwise does not work correctly, it can be backed
"      out by entering the command[s]:
[REDACTED];
"
```

Redisplay the APPLY Section

Display the first page of the APPLY section by entering **403<CR>** then **409<CR>** on the CMD< line. Note that the messages are no longer displayed in reverse video, indicating that the messages can be executed again.

```

SYS EMER   CRITICAL MAJOR   MINOR   BLDG/PWR HLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH   CU      CU PERPH  LINK

CMD< 403 409 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      Y 406 PERM      Y 412 PRINT
401 TERM                404 SOAK  407 FILE      410 NEXT  413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR: d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/skntnt.p
"      with the minimal file nt21a.m.
"
1 UPD: VFY:DATA , BWM = "CFT88-0001";
"
2 UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
  UF = "/etc/bwm/CFT88-0001/nt11c.m";

```

Reexecute the APPLY Section

Execute the APPLY section by entering **403<CR>** then **475<CR>** on the CMD< line. Message 1 is then shown blinking to indicate that message 1 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM              409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE              410 NEXT    413 RESET
                        405 BKOUT  408 LINE              411 PREVIOUS  EC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR: d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/sknt.p
"      with the minimal file nt21a.m.
"
"
"
2 UPD: APPLY:DATA , FUNCR, UPM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
UF = "/etc/bwm/CFT88-0001/nt11c.m";
```

Message 1 Completes Execution

When message 1 completes execution, message 1 appears in reverse video on the Field Update Page. Message 2 is then shown blinking to indicate that message 2 is currently executing.

```

SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS  EC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR: d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/sknt.p
"      with the minimal file nt21a.m.
"
"
"
2  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
"
```

Message 2 Completes Execution

When message 2 completes execution, the field update automatically displays Page 2 of the APPLY section on the Field Update Page. Message 3 is then shown blinking to indicate that message 3 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH      CU        CU PERPH   LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS  EC
402 DIR=
COMMUNICATION
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
   UF = "/etc/bwm/CFT88-0001/nt21a.m";
   "
```


Message 3 Completes Execution

When message 3 completes execution, message 3 is switched to reverse video. Message 4 is then shown blinking to indicate that message 4 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK

CMD< 403 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      Y 406 PERM              409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE              410 NEXT    413 RESET
                        405 BKOUT 408 LINE              411 PREVIOUS  EC
402 DIR=
COMMUNICATION
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
   "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/skntt.p"?
   "

```

APPLY Section Completes Execution

When field update completes the APPLY section, the SECTION COMPLETED message appears on the COMMUNICATION line in reverse video on the Field Update Page. Messages 3 and 4 also appear in reverse video.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 403 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK  407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS  EC
402 DIR=
      D
3  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
      "
4  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sknnt.p"?
      "
      "
```

Display the SOAK Section

Display the SOAK section by entering **404**<CR> then **409**<CR> on the CMD< line. The two soak messages, 5 and 6, are displayed in the message file display area.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 404 409 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM      █ Y 412 PRINT
401 TERM                █ AK 407 FILE      410 NEXT  413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"SOAK. *****
"
5  UPD:DISPLAY:DATA;
"
6  UPD:FTRC:DATA, FN="/fldupd/sndnt.p",FCN="control";
"
```

Execute the SOAK Section by Line Number

Execute the SOAK section by line number. In this example, message 5 is the first message in the SOAK section. To execute by line number, enter **408,5<CR>** then **475<CR>** on the CMD< line. (The SOAK section could have been executed automatically by entering **404** then **475**. The execution by line number is demonstrated only to show the capability.) Message 5 is then shown blinking to indicate that message 5 is currently executing.

```
SYS EMER   CRITICAL   MAJOR    MINOR    BLDG/PWR  HLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU       CU PERPH  LINK
CMD< 408,5 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        405 BKOUT  █=5             411 PREVIOUS █EC
402 DIR=
COMMUNICATION
"SOAK. *****
"
█;
"
6  UPD:FTRC:DATA, FN="/fldupd/sndnt.p",FCN="control";
"
```

When message 5 completes execution, message 5 is shown in reverse video. A message stating that message 5 has completed is shown on the COMMUNICATION line.

Execute Message 6 of the SOAK Section

Execute message 6 of the SOAK section by entering **408,6<CR>** then **475<CR>** on the CMD< line. Message 5 is still shown in reverse video, indicating that message 5 has completed execution. Message 6 is then shown blinking to indicate that execution is in progress.

```
SYS EMER   CRITICAL   MAJOR    MINOR    BLDG/PWR  HLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU       CU PERPH  LINK
CMD< 408,6 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        405 BKOUT  █=6             411 PREVIOUS █EC
402 DIR=
COMMUNICATION
"SOAK. *****
"
█;
"
█;
"
```

When message 6 completes execution, message 6 is shown in reverse video. A message stating that message 6 has completed execution is shown on the COMMUNICATION line. This completes the SOAK section.

Redisplay the APPLY Section

Display the APPLY section by entering **403<CR>** then **409<CR>** on the CMD< line. The first two messages of the APPLY section are shown in reverse video indicating that the messages have been executed.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU        CU PERPH  LINK
CMD< 403 409 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      Y 406 PERM      Y 412 PRINT
401 TERM                404 SOAK  407 FILE      410 NEXT  413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"APPLY. *****
"
"      MR: d88000001
"
"      this BWM updates /fldupd/sndnt.p with two minimal files,
"      nt11c.m and nt13c.m, and updates /fldupd/skntnt.p
"      with the minimal file nt21a.m.
"
"
"
2  UPD: APPLY:DATA , FUNCR, UPNM = "CFT88-0001", FN = "/fldupd/sndnt.p"?
"
```


Display the PERM Section

Display the PERM section by entering **406**<CR> then **409**<CR> on the CMD< line. The PERM section messages are displayed in the message file display area.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 406 409 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  RM          Y  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT  413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"PERM. *****
"
8  UPD: PERM:DATA , UPNM = "CFT88-0001";
"
9  UPD: RECLAIM:DATA , FN = "/fldupd/sndnt.p";
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Print the PERM Section

Print the PERM section on the ROP by entering **406<CR>** then **412<CR>** on the CMD< line. The PERM section messages continue to be shown in the message file display area.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH     CU       CU PERPH  LINK
CMD< 406 412 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  RM          409 DISPLAY  T
401 TERM                404 SOAK   407 FILE      410 NEXT     413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION
"PERM. *****
"
8  UPD: PERM:DATA , UPNM = "CFT88-0001";
"
9  UPD: RECLAIM:DATA , FN = "/fldupd/sndnt.p";
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Execute the PERM Section

Execute the PERM section by entering **406<CR>** then **475<CR>** on the CMD< line. The first message in the PERM section (message 8) is then shown blinking to indicate that message 8 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  RM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT     413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS EC
402 DIR=
COMMUNICATION
"PERM. *****
"
;
"
9  UPD: RECLAIM:DATA , FN = "/fldupd/sndnt.p";
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Message 8 Completes Execution

When message 8 completes execution, message 8 is shown in reverse video. Message 9 is then shown blinking to indicate that message 9 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH   CU       CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  RM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT    413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS EC
402 DIR=
COMMUNICATION
"PERM. *****
"
"
"
10 UPD:RECLAIM:DATA, FN="/fldupd/sknnt.p";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Message 9 Completes Execution

When message 9 completes execution, message 9 is shown in reverse video along with message 8. Message 10 is then shown blinking to indicate that message 10 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE (109)

400 BWM=CFT88-0001      403 APPLY  RM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE          410 NEXT    413 RESET
                        405 BKOUT  408 LINE          411 PREVIOUS EC
402 DIR=
COMMUNICATION
"PERM. *****
"
";
"
";
"
11 CMPR:DISK,CORE,FN="/fldupd/sndnt.p";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Message 10 Completes Execution

When message 10 completes, it is shown in reverse video along with messages 8 and 9. Message 11 is then shown blinking to indicate that message 11 is currently executing.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH    CU       CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  RM          409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT    413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS EC
402 DIR=
COMMUNICATION
"PERM. *****
"
";
"
";
"
";
"
12 CMPR:DISK,CORE,FN="/fldupd/sknnt.p";
```

Messages 11 and 12 Complete Execution

When messages 11 and 12 complete execution, they are shown in reverse video along with messages 8, 9, and 10.

```
SYS EMER    CRITICAL  MAJOR    MINOR    BLDG/PWR  HLD INH    CKT LIM    SYS NORM
TRAFFIC     SYS INH   CU        CU PERPH  LINK
CMD< 406 475 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY [REVERSE] 409 DISPLAY 412 PRINT
401 TERM                404 SOAK  407 FILE  410 NEXT  413 RESET
                        405 BKOUT  408 LINE  411 PREVIOUS [REVERSE] EC
```

```
402 DIR=
```

```
[REVERSE] D
```

```
"PERM. *****
```

```
"
```

```
[REVERSE] ;
```

```
"
```

```
[REVERSE] " ;
```

```
"
```

```
[REVERSE] ;
```

```
"
```

```
[REVERSE] " ;
```

```
"
```

```
[REVERSE] " ;
```

Terminate the Field Update Session

Terminate the update session by entering **401<CR>** on the CMD< line.

```
SYS EMER   CRITICAL   MAJOR    MINOR    BLDG/PWR  HLD INH   CKT LIM   SYS NORM
TRAFFIC    SYS INH    CU        CU PERPH  LINK
CMD< 401 OK                                FIELD UPDATE PAGE (109)
400 BWM=CFT88-0001      403 APPLY  406 PERM      409 DISPLAY  412 PRINT
401 TERM                404 SOAK   407 FILE      410 NEXT     413 RESET
                        405 BKOUT  408 LINE      411 PREVIOUS 475 EXEC
402 DIR=
COMMUNICATION _____
```

Once finished with a BWM, the BWM directory and its contents may be removed to conserve storage space. For the example, this is done by entering the following input message in the scrolling area:

```
In PDS,      UPD:CLR:BWM "CFT88-0001"!
In MML,      UPD:CLR:DATA, BWM="CFT88-0001";
```


Input Messages

Field update input messages (commands) function in a variety of ways depending on the options specified in the message. The following list of the field update input messages presents a summary of their functions. Refer to the appropriate Input Message Manual for details.

CMPR:DISK:CORE	Compares the disk image of a nonkillable process to its memory image.
IN:REMOTE	Allows the user to receive files from a remote site.
UPD:APPLY:BYTER	Applies a byte replacement to a process.
UPD:APPLY:FILER	Applies a file replacement by replacing an old file by a new one.
UPD:APPLY:FUNCR	Applies a function replacement.
UPD:BKOUT	Backs out one or more update packages.
UPD:BLDBOOT	Rebuilds the application-only boot image or the <i>UNIX</i> RTR operating system-only boot image.
UPD:CLR	Clears (removes) one or more update packages from the field update directory.
UPD:DISPLAY	Displays information from the field update database about field updates.
UPD:FTRC	Lists the function information from a <i>pfile</i> 's symbol table.
UPD:OMIT	Removes a pending update from the update package. Should be used only on updates that cannot be reset or continued.
UPD:PERM	Makes the updates of a temporary update package permanent.
UPD:PURGE	Purges records from the field update database.
UPD:RECLAIM	Reclaims the address space of old functions in function-replaced <i>pfiles</i> and shared libraries. This makes the reclaimed space available as patch space.
UPD:RESET	Resets an update's status to a previously "accepted" state.
UPD:VFY	Verifies update packages in the field update directory.

System Generation Database

The System Generation (SG) database is used by field update processes to determine what set of update actions are allowed for any particular file. These file characteristics are described within the File-System Body (*fsb*) forms of the SG database.

⇒ NOTE:

No update actions will be performed if the file to be updated does not exist within the SG database.

Refer to 254-303-104, *Recent Change and Verify Manual*, for information regarding additional field entries.

File-System Body Form

The *fsb* form specifies to the **3bmkdsk** program the location, owner, permissions, and so forth, for each file within a 3B20D/3B21D computer's file system. The following form is the first page of the *fsb* form for process **/bootfiles/eih**. The key fields of this form identify the disk, partition, and full pathname on the 3B20D/3B21D computers to the file associated with this form.

File System Body Information		fsb	(1/2) (Recent Change and Verify)
1.pack_id			
pack.name:rt		pack.number:0	
4.par_number:5		5.directory:/bootfiles	
6.fname:eih		7.ftype:x	
8.set_uid:-		9.set_gid:-	
10.usr:rw-	11.grp:r--	12.other:r--	
13.usr_id:0		14.grp_id:0	
15.crc:n		16.strip:f	

Enter Review, Validate, screen#, or Print:

FIELDS 1-6

- Fields 1-4 are key fields that identify the disk pack and disk partition.
- Fields 5-6 are key fields that identify the directory and file.

FIELDS 7-12

- Fields 7-12 specify the file type and file mode bits.

FIELDS 13-14

- Fields 13-14 specify the user ID and group ID of the file's owner.

FIELDS 15-16

- Fields 15-16 specify the crc information and stripping option.

Page 2 of the *fsb* form for **/bootfiles/eih** is as follows. Page 2 contains information used by field update processes.

fsb (2/2)

17.extra_blks:0
19.srcname:bootfiles/eih

18.fldupd_type:_____

Enter Review, Validate, screen#, or Print:

FIELD 17 – EXTRA_BLOCKS

- For contiguous files, field 17 contains the number of extra blocks needed by field update.

FIELD 18 – FLDUPD_TYPE

- Field 18 identifies the process type as indicated in the following table:

Field Entry	Meaning
noupd	No Update
boot	Boot Process
bootfiles	File which is part of boot image
dap	Display Administration Process
diag	Diagnostic
drpcp	Disk Resident Pumpcode Process
filereplace	File Replacement
gspac	Grasp process (generic access package)
inhrtimer	Inhibit Timer
kill15	Process to be killed with signal 15
mira	Maintenance Input Request Administrator
nonkillable	Nonkillable process
pfucode	Peripheral Microcode
qkill	Quasi-killable (nonself terminating)
qkill15	Quasi-killable with signal 15
fpart	File Partition
btfile	Bootfile replacement
nkfile	Nonkillable file replacement
kpupdate	Killable Kernel Process

FIELD 19 – SRCNAME

- Field 19 contains no pathname to the source file on the Product Engineering Control Center (PECC) or other support machines. This field has no meaning on the 3B20D/3B21D computers application dependent machines in the field.

System Generation Form 2

The System Generation Forms (*sgen2*) specify the pathnames of the most common processes needed at boot time. By convention, all files needed for the boot image reside in the */bootfiles* directory. If additional processes are required for the boot image, the additional processes will be specified on the process forms within the SG database. The key field for this form is always *sgen2*.

sgen2 (1/2)
Specifies Required Processes

- 1.ksgen2:SGEN2
- 2.kernel:bootfiles/3bsgen.kern
- 3.fmgr:bootfiles/fmprc
- 4.fcdrv:bootfiles/dkdrv
- 5.pmgr:bootfiles/3bpmgr
- 6.nub:bootfiles/3bnum
- 7.eih:bootfiles/eih

Enter Review, Validate, screen#, or Print:

sgen2 (2/2)

- 8.simprc:bootfiles/simprc
- 9.pcpaud:bootfiles/pcpaud.g
- 10.boot_process.pcreate
 - 1)/prc/bdf
 - 2)/prc/klmon
 - 3)/prc/pldmon
 - 4)/prc/suovprc
 - 5)/prc/fda
 - 6)/prc/scsd
 - 7) _____
 - 8) _____
 - 9) _____
 - 10) _____

FIELDS 2-9

- Fields 2-9 specify the names of processes to be located within the boot file directory.

FIELD 10

- Field 10 specifies the names of processes to be created at boot time.

Process Form

The *process* form describes processes, in addition to those on the *sgen2* form, that are needed as part of the boot image.

```
process (1/1)
Processes Other Than Basic UNIX RTR Processes (Recent Change and Verify)
1.boot_prc:bootfiles/inhadm
2.process_num:0
3.utility_id:0x86
```

Enter Review, Validate, screen#, or Print:

FIELD 1

- Field 1 is a key field that contains the pathname to the additional boot process. The most common boot processes are given in the *sgen2* form.

FIELD 2

- Field 2 provides the fixed process number to be assigned, if any.

FIELD 3

- Field 3 provides the utility ID number of the process.

ULARP Form

This ECD form contains information about a process or run command to be executed by the User-Level Automatic Restart Process (ULARP). The information contained in the forms is used by ULARP to:

- Specify the devices (file systems) to be mounted at boot time.
- Specify the “quasi-killable” processes to be restarted by ULARP if they should die or be killed.
- Specify commands that are to be executed at boot time.

The following form represents the *ularp* form for the Display Administration Process (DAP). DAP is a monitored (quasi-killable) process.

```
ULARP ECD Record                ularp      (1/1)
                                      (Recent Change and Verify)
  1.rec_name:DAP      2.exec_no:200
  3.child_priority:191 4.boot.prm:0xc1      5.sleep:0
  6.utild:0x203       7.runttype:PROCESS 8.run_config:BOTH
  9.,craft:y        10.cruc_craft:n 11.uniq_child:y 12.wait_for_run:n
13.proc_path/cft/dap/dap
14.proc_args:stype
15.shell_cmd:
```

FIELD 1

- Field 1 is a key field that contains the ULARP record name.

FIELD 7 runttype

- Field 7, when identified as PROCESS, indicates this form is for a monitored quasi-killable process.
- Field 7, when identified as COMMAND, indicates this form is for a command to be executed at boot time.

⇒ NOTE:

Refer to 254-303-104, *Recent Change and Verify Manual*, for additional information.

Contents

Introduction	14-1
System Integrity Subsystem	14-3
■ Audit-Control Subsystem	14-5
Software Integrity Records	14-6
Audit Identification	14-6
Audit Control and Scheduling	14-9
Audit-Control Subsystem Initialization	14-9
Audit Segmentation	14-10
Audit Dispatching	14-10
Audit Execution Modes	14-10
Blocking Audits	14-11
Audit Scheduling and Time Consumption	14-11
Scheduling Routine Audits	14-12
Inhibiting the Routine Execution of Audits	14-13
Frequency Groups	14-13
Timed Audits	14-14
Error Reporting and Recovery	14-14
Audit Execution Scenario	14-15
■ Audit User Input Commands	14-16
Allowing Audits	14-16
Requesting Audits	14-17
Inhibiting Audits	14-18
Reporting Status of Audits	14-19
Extracting Error Count Information for Audits	14-19
Analyzing Data for File-System Audits	14-20
Stopping Audits	14-21
■ Audit Output Messages and Error Reports	14-22

Contents

AUD Output Messages	14-22
OP AUD Output Message	14-23
OP AUDERR Output Message	14-23
OP FNAME Output Message	14-24
Processor Recovery Messages	14-24
Step B, Function Code 07	14-24
Step B, Function Code 08	14-25
REPT AUDSTAT Output Message	14-26
REPT SIMCHK Output Message	14-26
Corrective Action for Error Codes 101, 103, 104, 126, 127, 156, 158, and 183	14-35
Corrective Action for Error Code 141	14-36
Corrective Action for Error Code 146	14-36
Corrective Action for Error Codes 147, 152, and 154	14-36
Corrective Action for Error Code 159	14-37
Corrective Action for Error Code 190	14-37
Procedures for Modifying Audit Records	14-38
■ How to Modify Audit Records	14-38
■ How to Invoke RC/V	14-39
■ How to Review Audit Forms	14-42
■ How to Change Audit Forms	14-43
■ How to Insert Audit Forms	14-47
Operating-System Audits	14-50
■ Control Unit Memory Comparison Audit	14-51
Requesting the CUMEM Audit	14-51
Interpreting AUD CUMEM Output Messages and Error Reports	14-52
Responding to AUD CUMEM Error Reports	14-54
CUMEM Recent Change Forms	14-55
audrec Form for CUMEM 1	14-55
auderr Form for CUMEM 1, Category 0	14-56
auderr Form for CUMEM 1, Category 1	14-56
audinst Form for CUMEM 1	14-57
■ Control Unit Hardware Status Audit	14-58
Requesting the CUSTAT Audit	14-58
Interpreting AUD CUSTAT Output Message and Error Reports	14-59
Responding to AUD CUSTAT Error Reports	14-66
CUSTAT Recent Change Forms	14-68

Contents

<i>audrec</i> Form for CUSTAT 1	14-68
<i>auderr</i> Form for CUSTAT 1, Category 0	14-69
<i>auderr</i> Form for CUSTAT 1, Category 1	14-69
<i>auderr</i> Form for CUSTAT 1, Category 2	14-70
<i>auderr</i> Form for CUSTAT 1, Category 3	14-70
<i>audinst</i> Form for CUSTAT 1	14-71
■ ECD Audits	14-72
Requesting ECD Audits	14-72
Interpreting AUD ECD Output Messages and Error Reports	14-73
Responding to AUD ECD Error Reports	14-84
ECD Recent Change Forms	14-84
<i>audrec</i> Form for ECD 1	14-85
<i>audrec</i> Form for ECD 2	14-86
<i>audrec</i> Form for ECD 3	14-87
<i>auderr</i> Form for ECD 1, Category 3	14-88
<i>auderr</i> Form for ECD 1, Category 4	14-88
<i>auderr</i> Form for ECD 1, Category 5	14-89
<i>auderr</i> Form for ECD 2, Category 3	14-89
<i>auderr</i> Form for ECD 2, Category 4	14-90
<i>auderr</i> Form for ECD 2, Category 5	14-90
<i>auderr</i> Form for ECD 3, Category 3	14-91
<i>auderr</i> Form for ECD 3, Category 4	14-91
<i>auderr</i> Form for ECD 3, Category 5	14-92
<i>audinst</i> Form for ECD 1	14-92
<i>audinst</i> Form for ECD 2, ecd	14-93
<i>audinst</i> Form for ECD 2, appecd	14-93
<i>audinst</i> Form for ECD 3	14-94
■ ECD Owner Audit	14-95
Requesting the ECDOWN Audit	14-95
Interpreting AUD ECDOWN Output Messages and Error Reports	14-96
Responding to AUD ECDOWN Error Reports	14-97
ECDOWN Recent Change Forms	14-98
<i>audrec</i> Form for ECDOWN 1	14-98
<i>auderr</i> Form for ECDOWN 1, Category 1	14-99
<i>auderr</i> Form for ECDOWN 1, Category 2	14-99
<i>audinst</i> Form for ECDOWN 1	14-100
■ File-Manager Audits	14-100
Requesting FMGR Audits	14-102

Contents

Interpreting AUD FMGR Output Messages and Error Reports	14-103
Responding to AUD FMGR Error Reports	14-106
FMGR Audit Error Categories	14-106
FMGR Recent Change Forms	14-108
audrec Form for FMGR 1	14-108
audrec Form for FMGR 2	14-109
audrec Form for FMGR 3	14-110
audrec Form for FMGR 4	14-111
audrec Form for FMGR 5	14-112
audrec Form for FMGR 6	14-113
audrec Form for FMGR 7	14-114
audrec Form for FMGR 8	14-115
audrec Form for FMGR 9	14-116
auderr Form for FMGR 1, Category 0	14-117
auderr Form for FMGR 1, Category 5	14-117
auderr Form for FMGR 2, Category 0	14-118
auderr Form for FMGR 2, Category 2	14-118
auderr Form for FMGR 2, Category 3	14-119
auderr Form for FMGR 2, Category 5	14-119
auderr Form for FMGR 3, Category 2	14-120
auderr Form for FMGR 3, Category 4	14-120
auderr Form for FMGR 4, Category 0	14-121
auderr Form for FMGR 4, Category 2	14-121
auderr Form for FMGR 5, Category 1	14-122
auderr Form for FMGR 5, Category 2	14-122
auderr Form for FMGR 6, Category 0	14-123
auderr Form for FMGR 6, Category 1	14-123
auderr Form for FMGR 6, Category 2	14-124
auderr Form for FMGR 6, Category 4	14-124
auderr Form for FMGR 6, Category 5	14-125
auderr Form for FMGR 7, Category 0	14-125
auderr Form for FMGR 7, Category 1	14-126
auderr Form for FMGR 7, Category 2	14-126
auderr Form for FMGR 7, Category 3	14-127
auderr Form for FMGR 8, Category 2	14-127
auderr Form for FMGR 9, Category 5	14-128
audinst Form for FMGR 1	14-128
audinst Form for FMGR 2	14-129
audinst Form for FMGR 3	14-129
audinst Form for FMGR 4	14-130

Contents

<i>audinst</i> Form for FMGR 5	14-130
<i>audinst</i> Form for FMGR 6	14-131
<i>audinst</i> Form for FMGR 7	14-131
<i>audinst</i> Form for FMGR 8	14-132
<i>audinst</i> Form for FMGR 9	14-132
■ File-System Block Audit	14-133
Requesting the FSBLK Audit	14-134
Interpreting AUD FSBLK Output Messages and Error Reports	14-135
Responding to AUD FSBLK Error Reports	14-137
FSBLK Recent Change Forms	14-138
<i>audrec</i> Form for FSBLK 1	14-138
<i>auderr</i> Form for FSBLK 1, Category 1	14-139
<i>auderr</i> Form for FSBLK 1, Category 3	14-139
<i>audinst</i> Form for FSBLK 1, <i>/dev/db</i>	14-140
<i>audinst</i> Form for FSBLK 1, <i>/dev/etc</i>	14-140
<i>audinst</i> Form for FSBLK 1, <i>/dev/root</i>	14-141
<i>audinst</i> Form for FSBLK 1, <i>/dev/tmp</i>	14-141
<i>audinst</i> Form for FSBLK 1, <i>/dev/bdb</i>	14-142
<i>audinst</i> Form for FSBLK 1, <i>/dev/betc</i>	14-142
<i>audinst</i> Form for FSBLK 1, <i>/dev/broot</i>	14-143
■ File-System Link Audit	14-144
Requesting the FSLINK Audit	14-145
Interpreting AUD FSLINK Output Messages and Error Reports	14-146
Responding to AUD FSLINK Error Reports	14-148
FSLINK Recent Change Forms	14-149
<i>audrec</i> Form for FSLINK 1	14-149
<i>auderr</i> Form for FSLINK 1, Category 1	14-150
<i>auderr</i> Form for FSLINK 1, Category 2	14-150
<i>audinst</i> Form for FSLINK 1, <i>/dev/db</i>	14-151
<i>audinst</i> Form for FSLINK 1, <i>/dev/etc</i>	14-151
<i>audinst</i> Form for FSLINK 1, <i>/dev/root</i>	14-152
<i>audinst</i> Form for FSLINK 1, <i>/dev/tmp</i>	14-152
<i>audinst</i> Form for FSLINK 1, <i>/dev/bdb</i>	14-153
<i>audinst</i> Form for FSLINK 1, <i>/dev/betc</i>	14-153
<i>audinst</i> Form for FSLINK 1, <i>/dev/broot</i>	14-154
■ File-System Compaction Audit	14-155
When to Use the Compaction Audit	14-156
How to Use the Compaction Audit	14-158
Requesting the FSCMPT Audit	14-159

Contents

Interpreting AUD FSCMPT Output Messages and Error Reports	14-160
Responding to AUD FSCMPT Error Reports	14-162
FSCMPT Recent Change Forms	14-163
audrec Form for FSCMPT 1	14-163
auderr Form for FSCMPT 1, Category 1	14-164
auderr Form for FSCMPT 1, Category 3	14-164
auderr Form for FSCMPT 1, Category 5	14-165
audinst Form for FSCMPT 1, /dev/db	14-165
audinst Form for FSCMPT 1, /dev/etc	14-166
audinst Form for FSCMPT 1, /dev/root	14-166
■ Memory-Manager Audits	14-167
Requesting the MMGR Audits	14-168
Interpreting AUD MMGR Output Messages and Error Reports	14-169
Responding to MMGR Audit Reports	14-173
MMGR Recent Change Forms	14-174
audrec Form for MMGR 1	14-174
audrec Form for MMGR 2	14-175
audrec Form for MMGR 3	14-175
audrec Form for MMGR 4	14-176
audrec Form for MMGR 5	14-177
audrec Form for MMGR 9	14-177
auderr Form for MMGR 1, Category 1	14-178
auderr Form for MMGR 1, Category 2	14-179
auderr Form for MMGR 1, Category 3	14-179
auderr Form for MMGR 1, Category 4	14-180
auderr Form for MMGR 1, Category 5	14-180
auderr Form for MMGR 2, Category 1	14-181
auderr Form for MMGR 2, Category 2	14-181
auderr Form for MMGR 2, Category 3	14-182
auderr Form for MMGR 2, Category 4	14-182
auderr Form for MMGR 3, Category 1	14-183
auderr Form for MMGR 3, Category 2	14-183
auderr Form for MMGR 3, Category 3	14-184
auderr Form for MMGR 3, Category 4	14-184
auderr Form for MMGR 4, Category 3	14-185
auderr Form for MMGR 5, Category 3	14-185
auderr Form for MMGR 9, Category 0	14-186
audinst Form for MMGR 1	14-186
audinst Form for MMGR 2	14-187

Contents

<i>audinst</i> Form for MMGR 3	14-187
<i>audinst</i> Form for MMGR 4	14-188
<i>audinst</i> Form for MMGR 5	14-188
<i>audinst</i> Form for MMGR 9	14-189
■ Message-Buffer Audits	14-190
Requesting MSGBUF Audits	14-190
Interpreting AUD MSGBUF Output Messages and Error Reports	14-191
Responding to AUD MSGBUF Error Reports	14-195
MSGBUF Recent Change Forms	14-198
<i>audrec</i> Form for MSGBUF 1	14-198
<i>audrec</i> Form for MSGBUF 2	14-199
<i>auderr</i> Form for MSGBUF 1, Category 0	14-200
<i>auderr</i> Form for MSGBUF 1, Category 1	14-200
<i>auderr</i> Form for MSGBUF 1, Category 2	14-201
<i>auderr</i> Form for MSGBUF 1, Category 3	14-201
<i>auderr</i> Form for MSGBUF 1, Category 5	14-202
<i>auderr</i> Form for MSGBUF 2, Category 0	14-202
<i>auderr</i> Form for MSGBUF 2, Category 1	14-203
<i>auderr</i> Form for MSGBUF 2, Category 2	14-203
<i>auderr</i> Form for MSGBUF 2, Category 3	14-204
<i>audinst</i> Form for MSGBUF 1	14-204
<i>audinst</i> Form for MSGBUF 2	14-205
■ Plant Measurements System Audits	14-206
Requesting PMS Audits	14-206
Interpreting AUD PMS Output Messages and Error Reports	14-207
Responding to AUD PMS Error Reports	14-210
PMS Recent Change Forms	14-211
<i>audrec</i> Form for PMS 1	14-211
<i>audrec</i> Form for PMS 2	14-212
<i>auderr</i> Form for PMS 1, Category 1	14-213
<i>auderr</i> Form for PMS 1, Category 2	14-213
<i>auderr</i> Form for PMS 1, Category 3	14-214
<i>auderr</i> Form for PMS 1, Category 4	14-214
<i>auderr</i> Form for PMS 1, Category 5	14-215
<i>auderr</i> Form for PMS 2, Category 0	14-215
<i>auderr</i> Form for PMS 2, Category 3	14-216
<i>auderr</i> Form for PMS 2, Category 5	14-216
<i>audinst</i> Form for PMS 1	14-217
<i>audinst</i> Form for PMS 2	14-217

Contents

■ Process-Administration Audit	14-218
Requesting the PROAD Audit	14-218
Interpreting AUD PROAD Output Messages and Error Reports	14-218
Responding to AUD PROAD Error Reports	14-221
PROAD Recent Change Forms	14-222
<i>audrec</i> Form for PROAD 1	14-222
<i>auderr</i> Form for PROAD 1, Category 1	14-223
<i>auderr</i> Form for PROAD 1, Category 2	14-223
<i>audinst</i> Form for PROAD 1	14-224

Introduction

This chapter describes the system audits available to the user, as well as the software integrity subsystem — a system feature that monitors the operating system software reliability and stability by controlling and scheduling the execution of these audits. This chapter also provides procedural guidelines to follow when requesting the system audits, when interpreting the audit output messages and error reports, and when isolating and responding to system faults.

⇒ NOTE 1:

All *UNIX*® Real-Time Reliable (RTR) Operating System Release 1 and later audits are run in correcting mode. The File-System Link (FSLINK) and File-System Block (FSBLK) audits are run routinely depending on the application. FSLINK and FSBLK are automatically invoked by default in the noncorrecting mode, but they may be invoked manually in the correcting mode.

⇒ NOTE 2:

The control unit memory audit is not run automatically. It is designed to be run manually to isolate hardware faults in the control units.

⇒ NOTE 3:

The disk swap space compaction audit (MMGR 9) is not run automatically. It is designed to be run manually in response to repeated REPT MMGR004 error reports.

⇒ NOTE 4:

The File System Disk Space Compaction (FSCMPT) audit is not run routinely. It may be invoked manually to alleviate fragmentation of a file system. Unlike the other file system audits (FSBLK and FSLINK), the default mode for the file system compaction audit is correcting.

⇒ NOTE 5:

The action to be taken for an audit failure is specified in the output manual page for that audit.

System Integrity Subsystem

The system integrity subsystem is a collection of programs, processes, commands, database records, and performance records that work together to ensure the reliability of operating system software. Of the processes that make up the software integrity subsystem, the System Integrity Monitor (SIM) is probably the most important. The SIM is the process that is primarily responsible for monitoring and maintaining software integrity.

Because audit control makes up only a portion of the system integrity subsystem, this chapter will not attempt to describe the entire software integrity subsystem — only that part that deals with audit control. Throughout the remainder of this chapter, the audit-control portion of the system integrity subsystem will be referred to as the “audit-control subsystem.”

The audit-control subsystem, shown in Figure 14-1, consists of the following:

- **System Integrity Monitor (SIM)**
The SIM is a high-level, kernel process that is responsible for controlling, scheduling, and dispatching audits.
- **Equipment Configuration Database (ECD)**
The ECD is the database in which the audit records containing information about audit execution and error recovery are stored.
- **Audits**
The audits check the validity of data structures and, when necessary, corrects them.
- **System Integrity Output Formatter (SIOF)**
The SIOF is a program that outputs the results of audit execution in a consistent format.
- **Manual Input Commands**
The manual input commands are used to manually control audit execution and request audit status information. Refer to the “Audit User Input Commands” section.
- **Audit Requests**
The audit requests are requests from other processes to run audits.
- **Plant Measurements System (PMS) Database**
The Plant Measurements Database (PMDb) is the database in which counts of audit attempts and failures are stored.
- **REPT AUDSTAT Output Messages**
The REPT AUDSTAT output messages periodically report inhibited or blocked audits.
- **Audit Interface Libraries**
Audit interface libraries are a set of functions which audits use to communicate with the SIM.

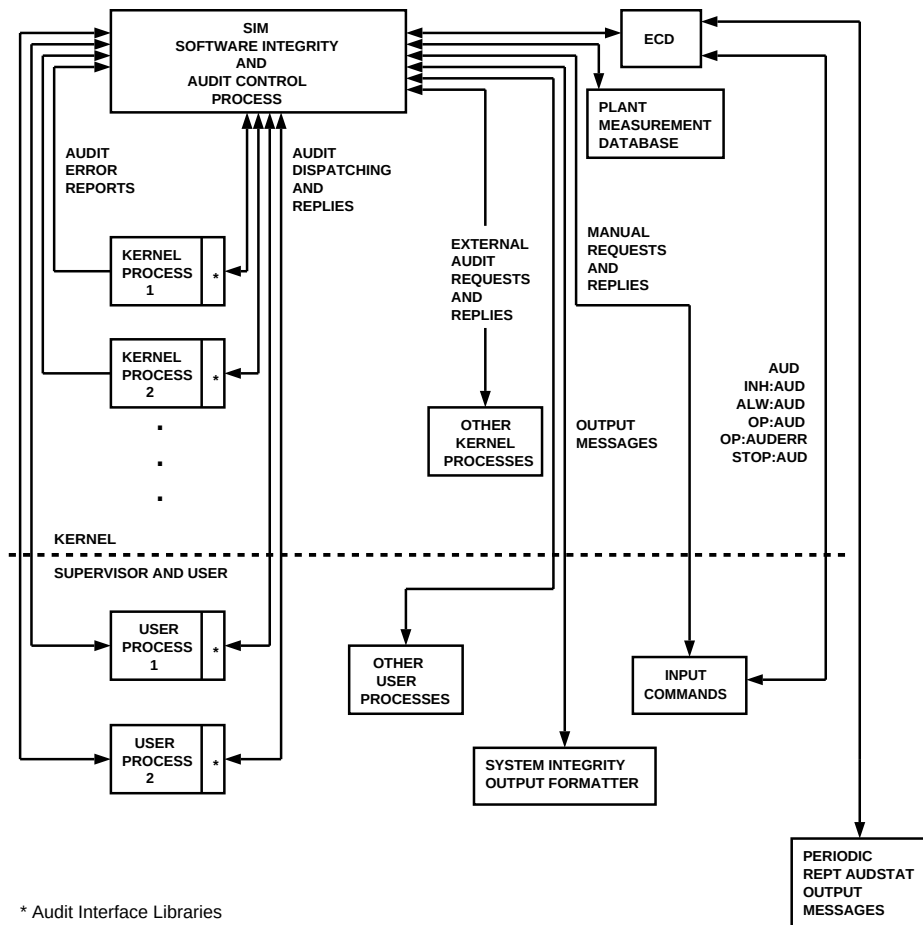


Figure 14-1. Audit-Control Subsystem

Audit-Control Subsystem

The audit-control subsystem categories audits as one of the following:

- Audits requested manually by someone entering an input command.
- Audits requested externally by a process.
- Audits dispatched routinely (automatically) by SIM.
- Audits demanded by SIM as part of a recovery action.

In all cases, SIM decides if and when the audit should run.

When SIM decides to dispatch the audit, SIM accesses the ECD and reads the audit's record. From the record, SIM finds where the audit resides. If the audit resides in a transient process, SIM will create the process. If the audit resides in a fixed process, SIM locates the process. SIM dispatches the audit process by:

- Sending a message
- Sending a message with *E_AUD event*
- Sending by *event* only.

The audit will check the accuracy of a data structure, and, in most cases, try to correct any errors it finds. When the audit has completed, it reports the results to SIM via the audit interface library.

The SIM then takes action based on the results of the audit execution. The SIM updates the PMDB, and, if appropriate, sends a message to the system integrity output formatter to output a message. If excessive errors were found during the audit execution, SIM takes recovery action.

Software Integrity Records

All software integrity information used by SIM is stored in the ECD. There are three types of records.

- A SIM control record that contains information that regulates the percentage of system time to be spent running audits, the parameters that control the scheduling of manually requested audits, and a master-inhibit state that is used to turn off the scheduling of all routine audits.
- An audit-control record for each audit that uniquely identifies the audit by name and member number. It contains all of the information that SIM needs to interface with that audit and to take recovery action when the audit reports errors. The information for the *auderr*, *audhist*, and *audstat* forms in the ECD are taken from the audit-control record.
- An audit-instance record for each instance of data to be audited. It contains the routine scheduling information and the error counts for that instance.

Audit Identification

Audits are grouped into categories based on the type of data audit. Each audit category is referred to as a family, and each family has a name, referred to as an audit name. The audit families and their corresponding audit names are listed in Table 14-1.

Table 14-1. Audit Families and Audit Names

Audit Family	Audit Name
Control unit memory	CUMEM
Control unit status	CUSTAT
Equipment configuration database	ECD
Equipment configuration database owner	ECDOWN
File manager	FMGR
File-system block	FSBLK
File-system link	FSLINK
File-system disk space compaction	FSCMPT
Memory manager	MMGR
Message buffer	MSGBUF
Plant measurements system database	PMS
Process administration procedures	PROAD

Within each audit family, there are one or more audits. Because the names of all these audits in a family are the same, member numbers are used to distinguish among audits in the same family (for example, the message-queue audit is MSGBUF 1 and the message buffer extender audit is MSGBUF 2). Table 14-2 is a listing of audit names and their corresponding member numbers.

Each audit record has associated with it one or more audit-instance records, and each instance may have a name that is up to 19 characters long. Audit instances were designed to maintain information specific to a particular occurrence of data to be audited.

For example, the file-system audit could have an instance for each file system; the instance name would be the name of the file system to be audited and other instance information would be relevant to auditing that file system.

For an audit with only one instance, the name and member number of the audit is used to identify the audit; the instance name may be null. If an audit has more than one instance, the audit name, audit member number, and audit-instance name must be used to identify the audit.

Table 14-2. Audit Names and Member Numbers

Audit Name	Member Number	Audit Description
CUMEM	1	CU memory comparison audit
CUSTAT	1	CU hardware status audit
ECD	1	Structural audit of incore ECD
ECD	2	Structural audit of disk ECD
ECD	3	Raw-data audit of incore ECD
ECDOWN	1	ECD record ownership audit
FMGR	1	FMGR task queue audit
FMGR	2	Internal inode table audit
FMGR	3	FMGR hash table and internal inode pointer table audit
FMGR	4	File table audit
FMGR	5	Internal capability table audit
FMGR	6	FMGR buffer audit
FMGR	7	Mount table audit
FMGR	8	FMGR delayed queue audit
FMGR	9	FMGR message queue audit
FSBLK	1	File-system block audit
FSLINK	1	File-system link audit
FSCMPT	1	File-system compaction audit
MMGR	1	Segment descriptor table audit
MMGR	2	Page table audit
MMGR	3	Page descriptor table audit
MMGR	4	Segment release audit
MMGR	5	Segment unlock audit
MMGR	9	Disk swap space compaction audit
MSGBUF	1	System message queue audit
MSGBUF	2	Message buffer extender audit
PMS	1	Structural and data integrity audit of PMS database
PMS	2	PMS database recovery audit
PROAD	1	Integrity audit of process administration procedures

Audit Control and Scheduling

The following points outline the procedures that make up audit control and scheduling:

- Audit-control subsystem initialization
- Audit segmentation
- Audit dispatching
- Audit execution modes
- Blocking audits
- Audit scheduling and time consumption
- Scheduling routine audits
- Inhibiting the routine execution of audits
- Frequency groups
- Timed audits
- Error reporting and recovery
- Audit execution scenario.

Audit-Control Subsystem Initialization

To initialize the audit-control subsystem after a system bootstrap, SIM must be able to access the SIM control record, each audit record in the ECD, and each audit record in the PMS database. If any step in the initialization procedure fails, SIM outputs a message — either a Processor Recovery Message (PRM) or a REPT SIMCHK message — to inform the user. Manual action is then required to determine the cause of the problem.

The SIM temporarily inhibits all audits and outputs a message if it encounters a problem accessing the ECD while it is running routine audits. If it is unable to allow all audits, 5 minutes later it outputs another message.

For PRMs, manual recovery procedures are listed in this chapter and in 254-303-103, *Processor Recovery Messages Guide*.

For REPT SIMCHK messages, refer to this chapter and 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for action to be taken.

After fixing a problem in the audit-control subsystem, the **ALW:AUD:ALL** command may be used to reinitialize the subsystem and begin running routine audits.

Audit Segmentation

The operating system audits are grouped into two categories:

- Segmented audits run in pieces (segments). One or more execute and, after a predetermined amount of time, relinquish the Central Processing Unit (CPU). Most kernel audits execute in a segmented mode.
- Nonsegmented audits run to completion and report the amount of CPU time used to the SIM. All *UNIX* system and supervisor-level audits are nonsegmented.

Audit Dispatching

Three mechanisms are used to dispatch audits:

- Messages — Most kernel audits are message dispatched. All *UNIX* system and supervisor-level audits must be message dispatched.
- Messages accompanied by an *E_AUD event* — These messages are provided as a performance improvement for kernel-level audits only.
- Events — The message-queue audit is *event* dispatched (MSGBUF_1).

Segmented audits are dispatched on each invocation, but nonsegmented audits are dispatched only once.

Audit Execution Modes

The audit-control mechanism supports four audit execution modes:

- Routine — Audits that are executed at a given frequency or at specified times during normal system operation.
- Manual — Audits that are manually requested by means of input commands.
- Software — Audits that are requested by processes other than input commands.
- Demand — Audits that are demanded by SIM as a result of a system error. All demand audits are nonsegmented.

The audit-control subsystem is flexible enough to allow an audit to execute in any of the four execution modes depending on the nature of the audit invocation. However, all execution modes might not be appropriate for all audits. The permitted execution modes of each audit are specified in the ECD record for that audit.

Blocking Audits

When an audit is prevented from running in any mode, it is said to be blocked. Audits can be blocked either by explicit or implicit blocking.

- Explicit blocking is when a process specifically requests the blocking of an audit instance. This prevents that audit instance from being executed in any mode.
- Implicit blocking is when scheduling audits, SIM ensures that no 2 audits with the same name are run simultaneously.

Audit Scheduling and Time Consumption

The audit-control subsystem provides a time-slicing environment for kernel process audits. All kernel audits should execute in a segmented mode, if possible, because the continuous execution of kernel audits could result in the lockout of lower level processes. When scheduled routinely or by a software request, kernel audits should execute one audit segment during each invocation. When a segmented audit is requested manually, it may execute more than one segment each time it is dispatched, as specified, by a parameter in the SIM control record. It is the responsibility of each audit to ensure that it does not exceed its recommended-segment length.

The mechanism used to schedule kernel-level audits is based on parameters K_{MIN} and K_{MAX} which are stored in the SIM control record. These parameters specify the minimum and maximum percentages of system time that should be spent running kernel-level audits.

The SIM uses a periodic time-out to control the scheduling of audit segments. The scheduling interval T_K is derived from K_{MIN} to ensure that kernel-level audits utilize at least K_{MIN} percent of the total system time. A segment from a currently running kernel-level audit is dispatched at the beginning of each scheduling interval, unless the previously dispatched segment has not yet completed. If that segment completes before the end of the scheduling interval, SIM uses system idle time to dispatch additional audit segments until it has dispatched a total of $\frac{K_{MAX}}{K_{MIN}}$ segments during any T_K interval. This ensures that kernel-level audits utilize no more than K_{MAX} percent of the total system time. The actual number of segments dispatched during each scheduling interval depends upon the current load on the system. This number is defined as the kernel-level multiplier, M_K :

$$1 \leq M_K \leq \frac{K_{MAX}}{K_{MIN}}$$

Nonsegmented kernel-level audits are handled by doubling the length of the next ($X_K - 1$) scheduling intervals, where X_K is the number of segments used by the audit, based upon the CPU time it reported.

For kernel-level audits, the audit-control subsystem permits one routine audit and one requested audit to run simultaneously, dispatching alternate segments to each. While a requested audit is running, routine audits run at only half their normal speed.

Since the time-sharing of supervisor processes is controlled by the *UNIX* RTR operating system scheduler, supervisor process audits execute in nonsegmented mode. The scheduler is designed to ensure a fair and equitable distribution of CPU time among all running supervisor processes, but it does not guarantee any percentage of system time to any process. Therefore, SIM must control the amount of system time used by supervisor-level audits. This control is based on parameters S_{MIN} and S_{MAX} , which are stored in the SIM control record. These parameters specify the minimum and maximum percentages of system time that should be spent running supervisor-level audits.

When SIM dispatches a supervisor audit, it records the real time. Upon completion, SIM uses the amount of CPU time reported by the audit and the current value of the kernel-level multiplier, M_K , to calculate the amount of real time, T_S , that should elapse in order to ensure that no less than S_{MIN} and no more than S_{MAX} percent of the total system time is spent running supervisor-level audits. If the real time that elapsed during the running of the audit was greater than T_S , the audit did not utilize the full supervisor time allocation. In that case, SIM dispatches another supervisor audit immediately. If the elapsed time is less than T_S , then SIM waits until time T_S has elapsed before dispatching another supervisor-level audit.

For supervisor-level audits, the audit-control subsystem permits one routine audit and a limited number of requested audits to run for all audits, for all instances of a specific audit, and for a specific instance of an audit. The **INH:AUD** and **ALW:AUD** commands are used to control these inhibit states. Setting or resetting the master-inhibit state for all audits does not affect the inhibit states of individual audits or audit instances.

Because audit-inhibit states are stored in the ECD, they are not lost across a Phase 2 (bootstrap) initialization. However, they are lost across a Phase 3 or a Phase 4 initialization. When the system is bootstrapped in minimum configuration, all audits remain inhibited; in full configuration, SIM allows routine audit scheduling after a time interval specified in the SIM control record, unless it had been inhibited manually before the bootstrap (Phase 2 only). Manually inhibited audits are not allowed automatically by SIM.

When the system is running in full configuration, The REPT AUDSTAT output message is generated periodically to report the inhibit status of audits.

Scheduling Routine Audits

Routine audits are scheduled in two ways. Some audits need to run often taking a small amount of time to ensure the integrity of critical system resources that are in a constant state of flux. Examples of these include the message-buffer

audits and various operating system-kernel-resource audits. Audits of this type should be scheduled as frequency-group audits.

Other audits need to run only occasionally, possibly taking a much longer time, to check on resources that change less often. Examples of these include database audits and file-system audits. Audits of this type should be scheduled as timed audits.

Inhibiting the Routine Execution of Audits

Inhibiting audits prevents them from being scheduled to run routinely or by software request. This should not be confused with blocking audits, as blocked audits cannot run in any mode, whereas inhibited audits can be requested manually.

Each audit and audit instance has an inhibit state. When this state is set (either manually or by SIM), its associated audit or instance will not be scheduled for execution.

A master-inhibit state also exists. This state is independent of the individual-inhibit states. When it is set, routine audit scheduling is turned off completely. When it is cleared, routine audit scheduling is turned on.

Setting or resetting the master-inhibit state does not affect the inhibit state of individual audits or audit instances. The **INH:AUD** and **ALW:AUD** commands are used to control these inhibit states.

Because audit-inhibit states are stored in the ECD, they are not lost across a Phase 2 (bootstrap) initialization. However, they are lost across a Phase 3 or a Phase 4 initialization. When the system is bootstrapped in minimum configuration, all audits remain inhibited. In full configuration, SIM allows routine audit scheduling after a time interval specified in the SIM control record, unless it had been inhibited manually before the bootstrap (Phase 2 only). Manually inhibited audits are not allowed automatically by SIM.

When the system is running in full configuration, the REPT AUDSTAT output message is generated periodically to report the inhibit status of audits.

Frequency Groups

Each frequency group audit is assigned a frequency labeled A through H, where A is the highest frequency and H is the lowest. Each audit in frequency group A executes twice as often as each audit in frequency group B; each audit in frequency group B executes twice as often as each audit in frequency group C; and so on.

The frequency group of each audit is specified in its instance record in the ECD. Because the frequency group is instance dependent, different instances of the same audit may be assigned to different frequency groups.

The time required to complete a full scheduling cycle of routine audits depends upon the current load on the system and upon the *total* number of routinely scheduled audits (including timed audits) that are not currently inhibited. All kernel-level audits compete with one another for their time allocation (K_{MIN} to K_{MAX}), and all supervisor-level audits compete for theirs (S_{MIN} to S_{MAX}). Requested audits also use part of those time allocations. Therefore, it is not possible to guarantee that an audit in any specific frequency group will execute repeatedly at a predetermined time interval.

Timed Audits

Audits that must be executed at given times of the day should use the timed audit facility. Each audit instance contains information that allows it to be scheduled to execute at specified hours of the day (0000 to 2300) on specified days of the week (Sunday through Saturday). Different instances may be scheduled to run at different times or on different days. A given instance cannot be both a timed instance and a frequency instance; however, the same audit may have one or more frequency instances and one or more timed instances.

Error Reporting and Recovery

Errors detected by audits are grouped into categories for purposes of reporting and recovery. The error categories used by an audit are specified in its ECD record. For each category, there is an error threshold, a time interval during which errors are counted, and a list of recovery actions that should be taken if the threshold is reached. Separate error counts are accumulated for each audit instance in its instance record. No error counts are accumulated when an audit is run in the noncorrecting mode.

Recovery actions are specified for each error category as follows:

- NOACT: Take no action for this error category and recovery stage. A NOACT entry cancels all other entries for the same stage number.
- SAMAUD: Run the same audit in the demand mode.
- RUNAUD: Run a different audit in the demand mode. For audits with more than one instance, SIM assumes that the instance name of the audit to be run is the same as that of the instance that reported the errors.
- RPTAIM: Report the audit results to the Application Integrity Monitor (AIM).
- TRMPRC: Terminate one or more processes.
- PANDMP: Set up a panic dump request in case the system reboots.
- PHASE: Request a system reinitialization unless software checks are inhibited.
- DFLT: Take the default action for this recovery stage.

Stage 1 recovery is defined as those actions that are taken when a routine or requested audit reaches an error threshold. Stage 2 recovery is defined as those actions that are taken when a demand audit reports abnormal completion and reaches an error threshold. Only one action of each type may be taken during each stage of recovery. If no stage 1 recovery action is specified, then the default action taken by SIM is to run the same audit in the demand mode. For stage 2, on the other hand, SIM never runs the same audit (since that could lead to an audit being run an endless number of times).

If an audit process dies while running an audit (abort code 9), SIM attempts to run it once more in the demand mode. SIM also runs an audit once it has aborted with abort code 4, indicating that it was faulted while correcting errors. If it aborts the same way a second time, SIM inhibits the audit and outputs a message.

Output messages are printed for routine audits and software requests only if they report errors or abnormal completion. Output messages are always printed for manual and demand audits. However, when a demand audit is run as a recovery action for a manual audit that failed, its output may not be printed on the same terminal from which the initial audit input command was entered.

Audit Execution Scenario

When an audit runs, the following sequence of events typically occurs:

1. The SIM accesses the ECD audit record. If the audit resides in a transient process, SIM creates the process, otherwise it finds the audit process using the information found in the audit record.
2. The SIM dispatches the audit.
3. The audit may find and possibly correct errors. It reports each error to the audit interface library which increments an error counter. One or more of the error reports result in the printing of a raw-data output message.
4. The audit finishes its work and reports its termination status to the audit interface library which reports the total error counts to SIM. SIM updates the error counters in the ECD audit record, and a final output message is printed through System Integrity Output Formatter (SIOF). SIM also updates the audit counts of attempts and failures in the PMS database.
5. If an error threshold has been exceeded, SIM takes recovery action as specified in the audit ECD record.

Audit User Input Commands

The input commands discussed in this chapter are used to manually control the execution of system audits or request information concerning their status. These input commands:

- Allow the routine execution of audits (**ALW:AUD**).
- Request audits (**AUD**).
- Inhibit the routine execution of audits (**INH:AUD**).
- Determine the status of audits (**OP:AUD**).
- Extract error information for audits (**OP:AUDERR**).
- Analyze data produced by file-system audits (**OP:FNAME**).
- Stop a currently active audit (**STOP:AUD**).

After any of these input commands are entered, the system responds with a two-character code and possibly a short message.

Additional information about each of the input commands discussed in this chapter can be found in the following Information Products (IPs):

- 254-303-110, *PDS Input Messages Manual*
- 254-303-112, *MML Input Messages Manual*.

Allowing Audits

The **ALW:AUD** command clears the inhibit state of previously inhibited audits or audit instances and allows them to be scheduled for routine execution.

The format of the **ALW:AUD** command is as follows:

In PDS, **ALW:AUD: { ALL | a [b] | a c , INS d } !**
In MML, **ALW:AUD:DATA, { ALL | a [=b] | a = c , INS = d } ;**

Where:

- ALL** Turns on the routine scheduling of audits controlled by SIM.
- Clears a master-inhibit state that is independent of the inhibit states for individual audits.
- Causes SIM to attempt to initialize the audit-control system if it has not been initialized successfully.
- Causes SIM to reinitialize its internal control parameters according to the new content of the control record (if the control record in the ECD for SIM is modified).

- a* Audit name. (Refer to Table 14-1.)
If *a* is specified without *b*, all members of *a* are allowed.
- b* Member number — values = 1 – 255.
If a set of member numbers is specified, the members of the set should be separated by commas and the entire set surrounded by parentheses; for example, (1,2,3).
- c* Single member number — values = 1 – 255.
- d* Instance name.

After the **ALW:AUD** command is entered, the system responds with a two-character code, and possibly, a short message. The system response indicates whether the request was accepted and is documented in 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Requesting Audits

There are two ways that audits can be requested to run: manually and by software request. A software request is generated by the operating system or application processes; for example, 5ESS® switch or Operator Services Position System (OSPS). The user, however, may enter an audit request manually using the **AUD** command.

The **AUD** command allows a user to request the execution of one or more audits or audit instances and can be used to ensure the accuracy of data after a hardware or software unit has been repaired or replaced.

The format of the **AUD** command is as follows:

In PDS, **AUD:***a b [,INS c] [; {d | e | d,e | e,d}][:"f"]!*
In MML, **AUD:** *a=b[INS=c][:{d | e | d,e | e,d}][:DATA, "f"];*

Where:

- a* Audit name. (Refer to Table 14-1.)
- b* Member number — values = 1 – 255.
- c* Instance name.
- d* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit should not correct errors.

- e DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found. (A limited number of reports is generated.)
SUM generates an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.
- f Arbitrary string of data. The data is passed to the audit for it to interpret.

Although all audits use the same **AUD** format, specific audits may not have multiple member numbers or instances.

Inhibiting Audits

The **INH:AUD** command disables one or more specified audits and prevents the disabled audit from being routinely dispatched by SIM. However, an audit can be executed manually by entering the **AUD** input command. An audit should be inhibited only if it is causing system problems or when it seems to be incorrect. All audits must be inhibited using the **INH:AUD:ALL** command before modifying the incore database according to Recent Change and Verify (RC/V) procedures.

The format of the **INH:AUD** command is as follows:

In PDS, **INH:AUD: { ALL | a [b] | a c , INS d } !**
In MML, **INH:AUD:DATA,{ ALL | a [= b] | a = c , INS = d } ;**

Where:

- ALL** Turns OFF the routine scheduling of audits controlled by SIM. Sets a master-inhibit state that is independent of the inhibit states for individual audits.
- a Audit name. (Refer to Table 14-1.)
- b Member number — values = 1 – 255.
If a set of members is specified, the members of the set should be separated by commas and the entire set surrounded by parentheses; for example, (1,2,3).
- c Single member number — values = 1 – 255.
- d Instance name.

Refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*, for the meanings of system responses to the **INH:AUD** command.

Reporting Status of Audits

The **OP:AUD** command reports the status of one or more audits controlled by SIM by printing audit-control records in the ECD.

The format of the **OP:AUD** command is as follows:

In PDS, **OP:AUD: { ALL | RUN | a[b] }!**
In MML, **OP:AUD:DATA,{ ALL | RUN | a[=b] };**

Where:

ALL Specifies all audits controlled by SIM.

RUN Specifies all audits that are currently executing.

a Audit name. (Refer to Table 14-1.)

If *a* is specified without *b*, the status of each member of *a* is reported.

b Member number — values = 1 – 255.

May be specified as a single number or a list of numbers. If a list is used, numbers should be separated by commas and the entire list surrounded by parentheses; for example, (1,2,3).

For the meanings of system responses to the **OP:AUD** command, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Extracting Error Count Information for Audits

The **OP:AUDERR** command extracts information on error counts from the ECD records for an audit or audits controlled by SIM.

The format of the **OP:AUDERR** command is as follows:

In PDS, **OP:AUDERR: { ALL | a[b] | a c,INS d }!**
In MML, **OP:AUDERR: { ALL | a[=b] | DATA,a=c,INS=d };**

Where:

ALL Specifies all audits controlled by SIM.

a Audit name. (Refer to Table 14-1.)

If *a* is specified without *b*, error counts of all specified members of *a* are reported.

b Member number — values = 1 – 255.

May be specified as a single number or a list of numbers. If a list is used, numbers should be separated by commas and the entire list surrounded by parentheses; for example, (1,2,3).

c Single member number — values = 1 – 255.

d Instance name.

To interpret the system responses to the **OP:AUDERR** command, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Analyzing Data for File-System Audits

The **OP:FNAME** command is used to analyze data produced by the file-system audits (FSBLK and FSLINK). When input, **OP:FNAME** reports the full pathname of a file given a block or inode number and a file system. Pathnames can be reported for:

- One or more inode numbers within a file system
- One or more block numbers within a file system.

⇒ **NOTE:**

The **OP:FNAME** command can be used only on mounted file systems.

The format of the **OP:FNAME** command is as follows:

In PDS, **OP:FNAME:FS "a",{ INODE | BLOCK } b !**
In MML, **OP:FNAME:DATA,FS="a",{ INODE | BLOCK } = b ;**

Where:

a File system name.

b Single inode or block number.

If a range of inode or block numbers is specified, the lowest and highest values should be separated by a hyphen; for example, 2-12.

To interpret system responses to the **OP:FNAME** command, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Stopping Audits

The **STOP:AUD** input command stops a currently active audit from executing.

The format of the **STOP:AUD** command is as follows:

In PDS, **STOP:AUD: a b !**
In MML, **STP:AUD:DATA,a=b;**

Where:

- a* Audit name. (Refer to Table 14-1.)
- b* Member number — values = 1 – 255.

To interpret system responses to the **STOP:AUD** command, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Audit Output Messages and Error Reports

Information on audit execution is reported to the user via the AUD output messages. There are, however, other messages that give important audit information to the user. Some of these messages are produced in response to manual requests that the user can input when additional audit information is needed; two are generated automatically giving error and status reports. In addition, there are two PRMs that give information to aid the user during 3B20D/3B21D computers recovery when the craft interface system is inoperative.

Additional information about each of the output messages discussed in this chapter can be found in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

AUD Output Messages

The AUD output message reports audit execution results to the user. There are two types of AUD output messages.

- The first type reports raw data that pertains to specific errors found by an audit.
- The second type reports the termination status of the audit.

The data reports detailed or raw include four words of raw data designated DATA1, DATA2, DATA3, and DATA4. The definitions of these data words are specific to each audit and are documented in the audit output message manual page. Some audits also provide supplementary data with AUD error messages to facilitate analysis of the reported error conditions.

The summary messages report the termination status of audits, which may be COMPLETED, STOPPED, or ABORTED. If the termination status is COMPLETED, then the total number of errors found and the number of errors corrected by the audit are reported in the message. If the termination status is ABORTED, then an abort code is provided to specify why the audit was aborted. Table 14-3 lists the abort codes and their meanings.

Table 14-3. Audit Abort Codes

Code	Meaning
3	Audit aborted internally without completing its work.
4	Audit was faulted while correcting errors.
5	SIM could not start or dispatch the audit.
6	Error was encountered in the audit control/audit library interface.
7	Audit exceeded its time-out or segment limit.
8	SIM aborted an executing routine audit because of a blocking request.
9	Transient process that was running the audit terminated.
10	A segmented audit exceeded an error threshold.
11	Audit was blocking a needed demand audit.
12	Audit rejected the request to run from SIM.
13	SIM aborted an executing audit because the audit system is being reinitialized.

OP AUD Output Message

The OP AUD output message is generated in response to the **OP:AUD** manual request by the audit request processor. **OP:AUD** prints out the status of specified audits controlled by SIM. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for the exact format of the OP AUD output message.

OP AUDERR Output Message

The OP AUDERR output message is generated in response to the **OP:AUDERR** manual request. **OP:AUDERR** prints out the error counts that were extracted from the ECD records for audits controlled by SIM. Information, such as error counts for audits, might be useful to the user as historical data. Or, for example, it might be possible to identify categories of errors from a listing of error counts when the total number of error reports for an audit was greater than the total number of raw-data output messages the system can generate. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for the format of the OP AUDERR message.

The OP AUDERR output message simply furnishes statistics or information and no action is required by the user.

OP FNAME Output Message

The OP FNAME output message is generated in response to the **OP:FNAME** input command and prints out the full pathname of a file, given a block or inode number and a file system. This output can be used to analyze data produced by the file-system audits (FSBLK, FSCMPT, and FSLINK) and should typically be called for after doing a file-system audit. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for the format in which OP FNAME is output.

This output message provides supplementary information on a file system, and no action is required by the user.

Processor Recovery Messages

There are currently two PRMs that are generated by SIM when the audit-control subsystem fails to initialize. The PRMs report events that happened during the system initializations or when the craft interface system is not functioning normally. These two PRMs are further designated within Step B as function codes 07 and 08. Both PRMs output information that describes a function or possible error and any corrective action that should be taken. Refer to 254-303-103, *Processor Recovery Messages Guide*, for details on interpreting PRMs.

Step B, Function Code 07

The PRM designated as Step B, function code 07, indicates an audit-control-subsystem-initialization failure. This means that SIM was unable to initialize the audit-control subsystem because one of the following has occurred:

- Failure to attach to the ECD
- Failure to read the ECDORG record
- Failure to read the SIM control record
- Failure to update the SIM control record
- Failure to access the audit records in the ECD.

The following corrective action should be taken when this PRM is output:

Step	Procedure
1	Verify and, if necessary, repair the SIM control record and audit records in the ECD. Use the RC/V procedures described in the "Procedures for Modifying Audit Records" section to review or modify the database.
2	Attempt to initialize the audit-control subsystem by entering: In PDS, ALW:AUD:ALL! In MML, ALW:AUD:DATA,ALL!
3	If these actions do not initialize the audit-control subsystem, seek technical assistance per local policy.

Step B, Function Code 08

The PRM coded as Step B, function code 08, indicates that SIM was unable to attach to the PMS database. Take the following corrective action if this PRM is output on the Maintenance Teletypewriter (MTTY):

Step	Procedure
1	Verify and, if necessary, repair the PMS database. Use the PMS audit described in the "Operating System Audit" section to repair the database.
2	Attempt to attach the PMS database to SIM by entering: In PDS, ALW:AUD:ALL! In MML, ALW:AUD:DATA,ALL;
3	If SIM is still unable to attach to the PMS database, seek technical assistance per local policy.

REPT AUDSTAT Output Message

The REPT AUDSTAT output message is produced automatically by the system approximately every 30 minutes. The report prints the status of inhibited audits controlled by SIM when one of the following conditions occur:

- All audits are inhibited, that is, routine audit scheduling is turned off.
- An audit member is permitted to run in routine mode, but it is inhibited.
- An audit member is permitted to run in routine mode, but an instance of it is inhibited.
- An audit member is permitted to run in at least one mode, but an instance of it is blocked.

Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for the format of REPT AUDSTAT.

REPT SIMCHK Output Message

The REPT SIMCHK output message reports if the audit-control subsystem:

- Fails to initialize after system bootstrap
- Reinitializes its data structures
- Reports audit scheduling failures
- Reports audit subsystem interface failures.

The SIM temporarily inhibits all audits and outputs REPT SIMCHK if it has difficulty accessing the ECD when routinely running audits. If SIM cannot allow audits to run within 5 minutes, it outputs another message. Manual action is required by the user to determine the cause of the problem. For a complete explanation of the REPT SIMCHK format as output by SIM, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Table 14-4 lists the error codes reported by REPT SIMCHK that apply to the audit-control subsystem.

Table 14-5 gives the optional hexadecimal data that is associated with specific error conditions reported by REPT SIMCHK.

Table 14-4. REPT SIMCHK Error Codes

Error	Meaning
*1	System initialization failure.
2	System initialization status message received with an unknown process number.
*12	Unexpected fault received by SIM.
101	Audit-control subsystem initialization failure.
102	Audit-record initialization error.
103	The SIOF program is not running. Some output messages were lost.
104	Routine audit scheduling was not allowed.
121	Failure to get an audit record from the ECD using its Record Identification (RID).
122	Failure to update an audit record in the ECD.
123	Failure to open a keyed sequence of audit records in the ECD.
124	Failure to get a record in a sequence of audit records.
125	Failure to get an audit record using the audit name and member number.
126	Failure to read the SIM control record.
127	Failure to update the SIM control record.
128	Failure to open a sequence of audit-instance records in the ECD.
129	Failure to get a record in a sequence of audit-instance records.
130	Failure to get an audit-instance record using its RID.
131	Failure to update an audit-instance record in the ECD.
132	Failure to get an audit-instance record using its instance name and the RID of the associated audit record.
141	Audit marked active in its ECD record when it is not running.
142	Invalid message received from the process manager or the utility manager regarding an audit process.
143	Audit reply received from an inactive audit.

* Indicates conditions that may result in a system initialization phase unless software checks are inhibited.

Table 14-4. REPT SIMCHK Error Codes (Contd)

Error	Meaning
144	Time-out occurred for an inactive audit.
145	Invalid reply received from an active audit.
146	SIOF output buffer overflow; some audit output messages were lost.
147	Failure to find one and only one audit process for the specified Utility Identification (UID).
148	Audit error report received from an inactive audit.
149	Invalid request to inform an audit process when the audit's ECD record is modified.
150	Invalid request to run or stop an audit.
151	Audit blocking condition cleared because it was invalid.
152	Failure to create a transient audit process.
153	Unable to take recovery action for an audit failure.
154	Audit inhibited by SIM because it could not be run, timed out, died, or was faulted while correcting errors.
155	Routine audit scheduling malfunctioned and is being reinitialized. All routine audits are temporarily inhibited.
156	Failure to reinitialize routine audit scheduling after it had malfunctioned.
157	Routine audit scheduling has been temporarily inhibited because SIOF is not running.
158	Failure to reinitialize routine audit scheduling after SIOF stopped running.
159	An audit-error report specified on invalid error category.
161	Audit system reinitializing because of error found in kernel-level-audit scheduling.
162	Audit system reinitializing because of error found for a kernel-level-audit time-out.
163	Audit system reinitializing because of error found for a routine supervisor-level-audit time-out.

Table 14-4. REPT SIMCHK Error Codes (Contd)

Error	Meaning
164	Audit system reinitializing because of error found for a requested supervisor-level-audit time-out.
165	Audit system reinitializing because SIM received a reply from an inactive audit.
166	Audit system reinitializing because SIM received a time-out for a nonrunning audit.
167	Audit system reinitializing because SIM received a raw-data report from an inactive audit.
168	Audit system reinitializing because SIM received an unexpected fault.
169	Audit system reinitializing because SIM found eligible audits not dispatched.
181	Failure to access an audit record in the PMDB.
182	Failure to create an audit record in the PMDB.
183	Failure to open the PMDB.
190	Audit has taken too much time to execute.
*200	Failure of the AIM process to reset the application sanity timer within the time-out interval.
201	Failure to activate or deactivate the application sanity timer.
202	The AIM was not connected to port PT_AIM when a Phase 0 was requested.
206	The AIM was not connected to port PT_AIM for reception of overload faults.
207	The AIM was not connected to port PT_AIM for soft-switch request.

* Indicates conditions that may result in a system initialization phase unless software checks are inhibited.

Table 14-4. REPT SIMCHK Error Codes (Contd)

Error	Meaning
208	The AIM was not connected to port PT_AIM for notification of Control Unit (CU) switch completion.
209	Faulty AIM port during craft initialization.
210	Faulty Input/Output Processor (IOP) port during craft initialization.
211	The AIM was not connected to port PT_AIM for notification of completion of diverge request.
212	The AIM was not connected to port PT_AIM for converge request.
213	The AIM was not connected to port PT_AIM for notification of completion of converge request.
300	Unrecognized acknowledgment message received by SIM.
301	Unrecognized USRACK message received by SIM.
306	Message with invalid type received by SIM.
307	Unrecognized MSFAULT message received by SIM.
310	Unrecognized Operating System Trap (OST) entry received by SIM.
401	Failure to respond to a request from another process for information about the most recent system initialization.
602	Overload status change.
605	Kernel-level-overload monitor process failed to initialize.
606	Supervisor-level-overload monitor process failed to initialize.

Table 14-5. Optional Hexadecimal Data for REPT SIMCHK Error Conditions

Error	Hexadecimal Data																		
1	Boot progress indicators, from left to right (hex digit 0 to hex digit 7). <table> <tr> <th>Hex Digit</th><th>Process</th></tr> <tr> <td>0</td><td>File manager</td></tr> <tr> <td>1</td><td>Disk driver</td></tr> <tr> <td>2</td><td>Process manager</td></tr> <tr> <td>3</td><td>Error interrupt handler</td></tr> <tr> <td>4</td><td>Capability manager</td></tr> <tr> <td>5</td><td>Scheduler</td></tr> <tr> <td>6</td><td>Utility manager</td></tr> <tr> <td>7</td><td>Memory manager</td></tr> </table> If a boot-process-indicator message is received from a kernel process, the appropriate hex digit is set to 1; otherwise, the hex digit remains at 0.	Hex Digit	Process	0	File manager	1	Disk driver	2	Process manager	3	Error interrupt handler	4	Capability manager	5	Scheduler	6	Utility manager	7	Memory manager
Hex Digit	Process																		
0	File manager																		
1	Disk driver																		
2	Process manager																		
3	Error interrupt handler																		
4	Capability manager																		
5	Scheduler																		
6	Utility manager																		
7	Memory manager																		
2	Process number of the process sending the message (ms_from).																		
12	Fault code, followed by the virtual address within SIM at which the fault occurred.																		
102	ECD RID of the audit record that was not initialized.																		
121	The RID used in attempt to get an audit record from the ECD.																		
122	The RID used in attempt to update an audit record in the ECD.																		
126	The RID of the SIM control record (as specified in the ECDORG record).																		
127	The RID of the SIM control record (as specified in the ECDORG record).																		
128	If not zero, the RID of the audit record for which a sequence of instance records could not be opened.																		
130	The RID used in attempt to get an audit-instance record from the ECD.																		
131	The RID used in attempt to update an audit-instance record in the ECD.																		
132	The RID of the audit record for which an associated instance record could not be obtained from the ECD.																		

Table 14-5. Optional Hexadecimal Data for REPT SIMCHK Error Conditions (Contd)

Error	Hexadecimal Data
141	The RID of the audit record in error.
142	The RID returned in the message (ms_ident).
143	The RID of the audit record as specified in the reply.
144	The RID of the audit record as specified in the time-out identification.
145	The RID of the audit record as specified in the reply.
147	The UID (pcode) of the process as specified in the audit record.
148	The RID of the audit record as specified in the error report.
149	The RID of the audit record as specified in the request, followed by the process number from which the request was received.
150	The RID of the audit record as specified in the request.
151	The RID of the audit-instance-record-process number of the blocking process and the elapsed time in milliseconds since the audit instance was blocked.
152	The RID of the audit record for which the audit process was not created.
153	The RID of the recovery audit that could not be run.
154	The RID of the audit that was inhibited.
159	The RID of the audit whose error was reported.
300	Message identity (ms_ident).
306	Message type (ms_type).
307	The fault code specified in the message.
310	The invalid OST number followed by the process number of the calling process.

Table 14-5. Optional Hexadecimal Data for REPT SIMCHK Error Conditions (Contd)

Error	Hexadecimal Data
602	Overload fault codes: 81 – Spooler output process overload cleared. 82 – Spooler output process overload. 85 – Data link input buffer overload cleared. 86 – Data link input buffer overload. 87 – Data link output buffer overload. 88 – Data link output buffer. 8B – IOP overload cleared. 8C – IOP command queue overloaded. 8D – An IOP was bootstrapped because of overload. 8F – Overload cleared on one Disk File Controller (DFC). 90 – A DFC is overloaded. 91 – File manager overload cleared. 92 – File manager overloaded. 95 – Supervisor/user-level lockout cleared. 96 – Supervisor/user-level process lockout. 97 – Timed entry overload cleared. 98 – Kernel process timed entries slow. 99 – Kernel-level lockout cleared. 9A – Kernel-level process lockout. 9C – Module 1 memory allocation reduced to 60 percent. 9D – Module 1 memory at 80 percent allocated [only for generics with the Expanded Main Memory (EMM) feature]. 9E – Insufficient memory available to grant a memory allocation (module 1) (only for generics with the EMM feature).

Table 14-5. Optional Hexadecimal Data for REPT SIMCHK Error Conditions (Contd)

Error	Hexadecimal Data
	9F – Dispatcher Control Table (DCT) overload cleared. A0 – DCT 70 percent full. For <i>UNIX</i> RTR Operating System Release 6.3 and later, the DCT is 85 percent full. A1 – DCT overflow. A2 – The Segment Descriptor Entry (SDE) table overload cleared. A3 – The SDE table nearly full. A4 – Segment descriptor table overflow. A5 – Disk swap space overload cleared. A6 – Disk swap space 80 percent full. A7 – Memory overload cleared. A8 – Kernel process memory has grown to within $\frac{1}{4}$ megabyte of its maximum size. This has reduced the size of swappable memory to within $\frac{1}{4}$ megabyte of its minimum size. A9 – Kernel process memory overflow. Swappable main memory is at minimum size. AC – Message buffer overload cleared. AD – Message buffer 70 percent full. AE – Message buffer 90 percent full.

⇒ NOTE:

If you wish to review a record of audit activity for historical purposes, refer to the most recent PMS report — the *REPT OP PMCR* output message. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for complete information on this report.

Corrective Action for Error Codes 101, 103, 104, 126, 127, 156, 158, and 183

The following corrective action should be taken for audit-system initialization failures — specifically error codes 101, 103, 104, 126, 127, 156, 158, and 183.

Step	Procedure
1	Inhibit audit scheduling by: In PDS, INH:AUD:ALL! In MML, INH:AUD:DATA,ALL;
2	Verify and, if necessary, repair the <i>/etc/siof</i> program, the SIM control record and audit records in the ECD, and/or the audit records in the PMS database.
3	If the data printed with the output message does not include an audit RID, go to Step 4. However, if the data printed with the output message includes an audit RID: In PDS, OP:AUD:ALL! In MML, OP:AUD:DATA,ALL; to determine the audit name and member number associated with the RID.
4	Attempt to initialize the audit-control subsystem: In PDS, ALW:AUD:ALL! In MML, ALW:AUD:DATA,ALL;
5	If these actions do not initialize the audit-control subsystem, seek technical assistance per local policy.

Corrective Action for Error Code 141

For error code 141, use the RID of the audit record and the **OP:AUD:ALL** command to determine the name and member number of the audit. Then use the **STOP:AUD** or **STP:AUD** input command to clear the audit record.

Corrective Action for Error Code 146

For error code 146, use the **OP:STATUS:PROCESS** input command to determine whether or not the SIOF program is running. The filename of SIOF is */etc/siof*.

If it is running, no further action is required. If it is not running, use the **INH:AUD** input command with the ALL option to turn off routine audit scheduling.

Repair the SIOF program, then use the **INIT:ULARP** input command to start it up. (This should result in the printing of several old AUD output messages that had been stored in the audit-output buffer.)

After SIOF has been started successfully, use the **ALW:AUD** input command with the ALL option to turn routine audit scheduling on again.

Corrective Action for Error Codes 147, 152, and 154

Error codes 147, 152, and 154 indicate failures that involve audit processes. To correct the error, first verify that a valid process file exists for the audit.

If the audit process in question is transient, the process file should be stored in the directory */audprc* and its filename should be listed in the ECD audit record. Use the **RCV:MENU:RCVECD** input command to read the record. (Refer to the "Procedure for Modifying Audit Records" section.)

If the audit process in question is quasi-killable, identify the process via its UID. The ECDOWN audit is a quasi-killable process. Its process file can be accessed via the pathname */prc/cdn*.

If you are responding to error code 154, also perform the following procedure:

Step	Procedure
1	If the audit process is still running and is either quasi-killable or transient, terminate the audit process by entering: In PDS, STOP:EXC:ANY, FN "a", UCL! In MML, STP:EXC:DATA,ANY, FN= "a", UCL; Where a is the pathname of the audit process.
2	Attempt to run the audit manually. Use the AUD input command.
3	If the error does not recur, enter the ALW:AUD command to allow the audit to run. If the error repeats, escalate to the next level of technical expertise at your location.

Corrective Action for Error Code 159

For error code 159, use the **OP:AUD** command to identify the audit causing the problem. This error is because of a missing *auderr* form.

Corrective Action for Error Code 190

For error code 190, use the **OP:AUD** command to identify the audit causing the problem. There are two possible reasons for the audit taking too much time to execute: the audit may be incorrectly reporting actual time instead of CPU time, or the audit may need to be broken down into several units.

Unless corrected, supervisor-audit scheduling could be adversely affected resulting in the irregular scheduling of lower frequency-level-supervisor audits. A maximum of nine of these REPTs will be printed in one 24-hour period of time.

Procedures for Modifying Audit Records

How to Modify Audit Records

The audit forms that make up audit records can be accessed by means of the RC/V interactive tool. The procedures described in this chapter show you how to use RC/V to:

- Review, validate, or change some or all of the audit forms that make up an existing audit record.
- Insert the audit forms necessary to create new audit records in the ECD.

Before using the procedures in this chapter, please note the following:

- The RC/V forms have key fields for which values must be entered before processing can continue. In the case of the audit forms, the key fields simply identify the particular audit form you wish to access. The audit forms and their key fields are identified as follows:
 - *audrec* has two key fields which identify the audit name and member number.
 - *auderr* has three key fields which identify the audit name, member number, and error category number.
 - *audinst* has three key fields which identify the audit name, member number, and instance name. (If the audit has only one instance, the instance name does not have to be entered for the third key field; just leave the instance name blank and press the **RETURN** key.)

⇒ **NOTE:**

An instance name can contain up to 19 characters and can be any file system. The file-system audit could have an instance for each file system. At least one *audinst* form must be entered for each instance to be audited and each *audrec* form that is entered.

For example, if you were going to do a File-System Block Audit (FSBLK) on instance names */dev/root*, */dev/etc*, */dev/db*, or */dev/tmp*, the ECD audit-instance records are already supplied with the *UNIX* RTR operating system. However, if you were going to do a FSBLK audit on a different file system (instance name), you would have to create the proper audit-instance records.

- To list all keys associated with a particular form type, use the *klist* form of RC/V. (The RC/V manual has a detailed description on how to use the *klist* form.) The *klist* is only valid outside the **trbegin-trend** pair.
- After entering a value in an RC/V session, you must press the **RETURN** key.
- Refer to 254-303-104, *Recent Change and Verify Manual*, for complete information on how to use RC/V, RC/V navigation commands, illustrations of audit forms, and definitions of audit form fields.

How to Invoke RC/V

The RC/V activity can be invoked in a couple of ways. One way is by entering the poke command, **199**, on the MTTY. This causes the *rcvparams* form to be displayed on the MTTY. Once this form is displayed, enter the name of the database that you wish to review in the first field and press the **RETURN** key. Legal values for the database name are as follows:

- **incore** — Incore copy of database.
- **root** — Disk copy of application database.
- **rootdmly** — Disk copy of dmertononly database.
- **broot** — Backup application database (valid only if the backup database partition is mounted) on */broot/database*.
- **brootdmly** — Backup dmertononly database (valid only if the backup database partition is mounted) on */broot/database*.

For the second field, enter **y** if you wish only to review audit forms in the database without making any changes to them or enter **n** if you wish to change, insert, or delete audit forms in the database. You can let all the other fields default by pressing the **RETURN** key for each one.

RCV Parameter Info	(1/1) (Recent Change and Verify)
1.database_name: _____	
2.reviewonly:_____	3.journaling:_____
4.print_file: _____	
5.echo_file: _____	
6.flatedcd:_____	7.sysgen:_____

Another way to invoke RC/V is by entering the **RCV:MENU:RCVECD** command.

⇒ **NOTE:**

If the **RCV:MENU:RCVECD** command is entered on an MTTY, the STDIN and STDOUT options *must* be used and redirected. STDIN and STDOUT do not have to be used if the **RCV:MENU:RCVECD** command is entered on a terminal other than an MTTY.

The following is the format of the **RCV:MENU:RCVECD** command:

In PDS,

RCV:MENU:RCVECD, DB "a", TERM b, STDIN "c", STDOUT "d", e!

In MML,

RCV:MENU:RCVECD, DB="a", TERM=b, STDIN="c", STDOUT="d", e;

Where:

a	Name of the database you wish to access. Legal values are:
incore	Incore copy of database.
root	Disk copy of application database.
rootdmly	Disk copy of dmertonly database.
broot	Backup application database (valid only if the backup database partition is mounted).
brootdmly	Backup dmertonly database (valid only if the backup database partition is mounted).

- b* Terminal type on which the RC/V audit forms are to be displayed. Valid terminal types are VT100, VT52, 5620, 630, ADM3A, ADM42, HP2621, GNAT, and DUMB. Default is VT100.
- c* Pathname of the file for redirection of standard input.
- d* Pathname of the file for redirection of standard output, such as */dev/null*.
- e* NREVIEW or REVIEW. Use NREVIEW when you need to make changes to the audit forms. Use REVIEW when you only want to review the audit forms.

When RC/V is successfully invoked, the following form selection prompt is displayed on the screen. If the form selection prompt does not appear, an error occurred during invocation. Try again.

```
Recent Change and Verify
=====

Enter Form Name:
```

How to Review Audit Forms

The following steps show how to review audit forms without making any changes to them. You can use this procedure to quickly review an audit form, for example, when you need to see how an audit is scheduled to run, whether it is correcting or noncorrecting, or to see the time-out value of the audit. You can also use this procedure to quickly validate an audit record by accessing and then validating each of the audit forms that make up the record.

Step	Procedure
1	Invoke RC/V. If you invoke RC/V with the poke command, enter a value of y for the review only field of the <i>rcvparams</i> form. If you invoke RC/V with the RCV:MENU:RCVECD command, use the REVIEW option.
2	At the Enter Form Name prompt, enter the name of the form you wish to review or validate (<i>audrec</i> , <i>auderr</i> , <i>audinst</i> , <i>audhist</i> , or <i>audstat</i>) and press the RETURN key. The screen should clear, and the first page of the blank form should be displayed.
3	Enter the values for the key fields. (Enter the name, member number, and possibly, the instance name of the audit form you wish to review. Refer to section "How to Modify Audit Records", if needed, for a description of the key fields on the audit form.) (a) If you entered legal values for the key fields, the system will fill in the current values for the remaining fields. The cursor is moved to the bottom of the screen. (b) If you entered invalid values for the key fields, the message FORM NOT FOUND will be printed in the upper right corner of the screen and the cursor is returned to the beginning of the first key field. If this happened, reenter the correct values for the key fields.

Step	Procedure
4	The cursor is now at the bottom of the screen at a line that prompts you to select an operation — review, validate, screen number, or print. (a) If you wish to review the form of another audit, type r and the cursor will go back to the first key field of the form. You can then repeat Steps 3 and 4. (b) If you wish to validate the form, type v. The system will run a check on the form and will then print a message that tells you if errors were found. (c) If you are reviewing an <i>audrec</i> form and you wish to see its second page, type 2.
5	When you have finished reviewing the audit forms, type < in response to the operation selection prompt located at the bottom of the screen. This takes you to the screen that contains the form selection prompt. Type < again to return to the shell. No further action is required.

How to Change Audit Forms

The following procedure describes how to change the fields on an audit form. This procedure can be useful when an audit's time-out value needs to be changed, or when an audit needs to be scheduled differently.

Step	Procedure
1	If you are modifying a disk database, continue to Step 2. If you are modifying the incore database, inhibit the routine scheduling of all audits by entering the following on the MTTY: In PDS, INH:AUD:ALL! In MML, INH:AUD:ALL; Inhibit the diagnostics by entering the following: In PDS, INH:DMQ:SRC ALL In MML, INH:DMQ:SRC ALL The system should respond with either OK or NG.

Step	Procedure
2	Invoke RC/V. If you invoke RC/V using the poke command, enter a value of n for the review only field in the <i>rcvparams</i> form. If you invoke RC/V using the RCV:MENU:RCVECD command, use the NREVIEW option. When RC/V is successfully invoked, the screen clears and the form selection prompt appears. If the form selection prompt does not appear, an error occurred during invocation. Try again.
3	Enter trbegin at the form selection prompt. The system displays a form for trbegin with the key field "<i>tr_name</i>" having a value of "TRBEGIN". Press the RETURN key. The cursor moves to the bottom of the screen and you are prompted to select an operation — execute, change, validate, or print. Type e and press the RETURN key. The screen eventually clears, and the form selection prompt is displayed again.
4	Enter the name of the form you wish to modify (<i>audrec</i>, <i>auderr</i>, or <i>audinst</i>). The screen once again clears, and an operation selection prompt appears at the bottom of the screen. Enter u for update, and press the RETURN key. The screen clears, and the first page of the blank form is displayed.
5	Enter the values for the key fields. (Enter the name, member number, and possibly the instance name of the audit form you wish to modify. Refer to section "How to Modify Audit Records", if needed, for a description of the key fields on the audit form.) (a) If you entered legal values for the key fields, the system will fill in the current values for the remaining fields and the cursor is moved to the bottom of the screen. (b) If you entered invalid values for the key fields, the message FORM NOT FOUND will be printed in the upper right corner of the screen and the cursor is returned to the beginning of the first key field. If this happened, reenter the correct values for the key fields.

Step	Procedure
6	Change the values of fields to other legal values by typing c. The prompt at the bottom of the screen changes to Change field:. Type the number of the field you want to change. The cursor jumps to that field. Type the new value for the field, and press the RETURN key. The cursor returns to the bottom of the screen, and you are prompted for the next field to be changed. If you have more fields to change, repeat Step 6. If you are finished, press the RETURN key to exit the change mode. The prompt to select an operation reappears. Type u for update, and press the RETURN key. If the update is successful, the FORM UPDATED!! message appears in the upper right corner of the screen.
7	If you want to modify other forms, type < to take you back to the form selection prompt and repeat Steps 4 through 6.
8	When no more forms need to be updated, type < to get back to the form selection prompt. Once there, enter trend. The screen clears and a form for <i>trend</i> is displayed. This form has four fields, the first of which is a key field "<i>tr_name</i>" with a value of "TREND." Press the RETURN key. The other fields should be defaulted. Press the RETURN key for each of them.  CAUTION: <i>Overriding defaults of these data fields can cause an inconsistent database.</i>
9	Type e to execute the <i>trend</i> form. This causes data consistency checks to run. If the checks run successfully, the changes are made to the database and the form selection prompt reappears. If the checks are unsuccessful, you remain in the transaction state and have the options of making the changes needed to fix the inconsistencies or of aborting the transaction. (To abort the transaction, type ! and exit the RC/V session without completing the <i>trend</i> form.)

Step	Procedure
10	When you are ready to exit the RC/V session, type < to return to the form selection prompt and once there, type < again. This returns you to the shell.
11	If you have modified the incore database, enter the following command on the MTTY: In PDS, ALW:AUD:ALL! and ALW:DMQ:SRC ALL In MML, ALW:AUD:ALL; and ALW:DMQ:SRC ALL This turns routine scheduling of audits back on and causes the modified records to be included in the routine scheduling cycle.
12	If the change was to the incore database and should be permanent, perform an activate (by using the <i>activate</i> form of RC/V) to update the disk database or make the same changes to the disk database, usually called root. The <i>activate</i> form is executed outside trbegin-trend pair and in the NREVIEW mode. Enter yes in the key field (<i>copy_inc_to_disk:</i>). When the execute is successful, the disk database contains the same information that is in the incore database.

How to Insert Audit Forms

The following procedure describes how to insert audit forms into the database. This procedure can be used to add an instance to an existing audit (by inserting an *audinst* form for the existing audit into the ECD), or to add an entirely new audit (by inserting an *audrec* form and the appropriate number of *auderr* and *audinst* forms for the audit into the ECD).

Step	Procedure
1	If you are inserting audit forms into a disk database, continue to Step 2. If you are inserting audit forms into the incore database, inhibit the routine scheduling of audits by entering INH:AUD:ALL! on the MTTY. The system should respond with either OK or NG.
2	Invoke RC/V. ▪ If you invoke RC/V using the poke command, enter a value of n for the review only field in the <i>rcvparams</i> forms. ▪ If you invoke RC/V using the RCV:MENUE:RCVECD command, use the NREVIEW option. When RC/V is successfully invoked, the screen clears and the form selection prompt appears. If the form selection prompt does not appear, an error occurred during invocation. Try again.
3	Enter trbegin at the form selection prompt. The system displays a blank form for <i>trbegin</i> and waits for you to enter a value for the key field. 1.tr_name: Enter trbegin. The cursor moves to the bottom of the screen and you are prompted to select an operation — execute, change, validate, or print. Type e, and press the RETURN key. The screen eventually clears, and the form selection prompt is displayed again.
4	Enter the name of the form you wish to insert (<i>audrec</i>, <i>auderr</i>, or <i>audinst</i>). The screen clears, and an operation selection prompt appears at the bottom of the screen. Enter i for insert, and press the RETURN key. The screen clears, and the first page of the blank form is displayed.
5	The system prompts you to enter a value for each field. Enter the values for the fields. After the values for all of the fields have been entered, the prompt to select an operation reappears.

Step	Procedure
6	Type i for insert, and press the RETURN key. If the insert is successful, the FORM INSERTED!! message appears in the upper right corner of the screen.
7	If you want to insert other forms, type < to take you back to the form selection prompt and repeat Steps 4 through 6.
8	When you have completed the NREVIEW session of RC/V, type < to get back to the form selection prompt. Once there, enter trend. The screen clears, and a blank form for <i>trend</i> is displayed. This form has four fields, the first of which is a key field that you must fill in. Enter trend for the key field. 1.tr_name: All the other fields should be defaulted. Press the RETURN key for each of them.  CAUTION: <i>Overriding defaults of these data fields can cause an inconsistent database.</i>
9	Type e to execute the form. This causes data consistency checks to run. (a) If the checks run successfully, the changes are made to the database and the form selection prompt reappears. (b) If the checks are unsuccessful, you remain in the transaction state and have the option of making the changes needed to fix the inconsistencies or of aborting the transaction. (To abort the transaction, type ! and exit the RC/V session without completing the <i>trend</i> form.)
10	Exit the RC/V session by returning to the form selection prompt and typing < to return to the shell.

Step	Procedure
11	If you inserted audit forms into the incore database, enter the following command on the MTTY: In PDS, ALW:AUD:ALL! In MML, ALW:AUD:ALL; This turns routine scheduling of audits back on and causes the new audit records to be included in the routine scheduling cycle.
12	If changes should be permanent and database was incore, do an activate to update the disk database.

Operating-System Audits

This section provides detailed information about the members of the audit-control subsystem. Each audit in the audit-control subsystem detects and isolates errors in the *UNIX* RTR operating system data structures. The user can use the audits for system maintenance in conjunction with other tools, such as RC/V procedures and operating system diagnostic reports. These audits help define system hardware and software problems and also provide historical information for tracing back through events that occurred to the origins of the errors reported by the audits.

The audit-control subsystem begins operating when the system software is initialized in full configuration and runs on a regular basis to determine the sanity of the system unless audits are manually inhibited. These routine audits are invoked periodically by the audit-control subsystem as specified in the ECD.

To help troubleshoot problems, the user can manually request audits via the **AUD** input commands. Each **AUD** input command generates an AUD output message. Output messages are also generated when errors are detected during the continual, automatic operation of the audit system. (Refer to the “System Integrity Subsystem” section for additional information on audit output messages.)

The **AUD** input commands are handled by the audit-request processor, a program designed to receive the data transmitted by the craft shell and to generate calls to SIM containing information about the requested audit. In turn, SIM transmits audit results of how audits execute to the system-integrity-output formatter, which generates output messages for audit errors and audit completions.

The system-integrity-output formatter provides a consistent format for AUD output messages when reporting audit results to the user. The user must interpret the AUD output messages and error reports and respond, if possible, to the error reports.

This section is organized into 12 subsections — one corresponding to each audit family. Each subsection provides the following information:

- Description of the audit
- Explanation of how to request an audit (or audits) in that family using the user **AUD** input command
- Information on how to interpret AUD output messages and error reports
- Guidelines on how to respond to audit-error reports
- Illustrations of the RC/V audit forms that populate the ECD.

Control Unit Memory Comparison Audit

The Control Unit Memory (CUMEM) comparison audit compares the active and standby main store units in the 3B20D/3B21D computers. It runs by manual request only and reports a completed termination status after all of the main store has been audited.

⇒ NOTE:

The CUMEM audit is a manually invoked audit that can take more than 90 minutes to complete in a 16-megabyte main store configuration. It should be requested only when the processor configuration is ACTIVE/STANDBY. Excessive error counts during the running of this audit (in correcting mode) may result in a hardware reconfiguration with one of the CUs being marked Out of Service (OOS).

The CUMEM audit compares the content of the on-line main store with that of the standby unit. It considers the on-line main store to be correct and reports most errors against the standby when mismatches occur. If the on-line cache is the source of the problem, the errors are reported against the on-line cache unit.

If the on-line or off-line CU exceeds its error threshold and is taken out of service, the CUMEM audit aborts.

Requesting the CUMEM Audit

To request the CUMEM audit, use the following command:

In PDS, **AUD:CUMEM 1** [*a* | *b* | *a,b*]!
In MML, **AUD:CUMEM=1**[*a* | *b* | *a,b*];

Where:

- a* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
- b* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found plus an audit-completion message. (A limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

Interpreting AUD CUMEM Output Messages and Error Reports

The AUD CUMEM 1 output messages report the termination status of the audit and provide information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “a” is an error number that identifies the type of error that occurred. DATA1 for this audit contains an assert number which identifies the audit-test number that failed; however, for error numbers 30, 34, or 36, the DATA1 field contains the main store address at which the test failed. Descriptions of the CUMEM error numbers are listed in Table 14-6. The meanings of the DATA1 assert numbers are listed in Table 14-7. The following output message format may vary among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD CUMEM 1 ERROR a REPORT

DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the CUMEM audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors the CUMEM audit found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The CUMEM audit returns four error categories. At this time, the categories are undefined and reserved for future development.

Table 14-6. CUMEM Audit Error Numbers

Category	Error Number	Meaning
0,3	2	Bad data in ECD.
2,3	4	ECD access function failed.
0	6	Maintenance channel failure.
0	8	Reading off-line memory via update circuit failed.
1	10	Kconfig function call failed.
0	12	CU was found to have the wrong major state in ECD.
0	24	PMS database access failure.
0	28	Internal Processor Control Process Audit (PCPAUD) failure.
0	30	Off-line store access failure.
1	32	Memory audit internal error.
0	34	Off-line memory mismatch.
0	36	On-line cache memory mismatch.
0	38	CUMEM run when off-line was not standby.

Table 14-7. CUMEM Error Reports

Assert Number	Meaning
0x2000	Memory audit did not complete its entire memory block comparison.
0x2001	Kconfig function failure <i>rptrerr</i> .
0x2002	Kconfig function failure <i>rptrerr</i> .
0x2003	Kconfig function failure <i>fltdrv</i> .
0x2004	CUMEM called when off-line CU was not standby.

Under some error conditions, the CUMEM audit may produce REPT PCPAUD FAULT output messages. This message has the following format:

```
REPT PCPAUD FAULT TYPE a ASSERT NO b
```

Where:

a Fault type.

b Assert number.

For a detailed explanation of the REPT PCPAUD FAULT output message and a list of fault types and assert numbers, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Responding to AUD CUMEM Error Reports

For error numbers 30, 34, or 36, take the following actions:

Error Number	Action to Be Taken
30	The off-line CU should have been removed automatically. Diagnose, repair, if necessary, and restore the CU. Then switch CUs and remove, diagnose, repair, and restore the other CU.
34	If the off-line CU was not taken out of service, remove it. Diagnose, repair, if necessary, and restore the CU.
36	If the on-line CU was not taken out of service, switch CUs and remove it. Diagnose, repair, if necessary, and restore the CU.

If you receive error numbers other than 30, 34, or 36 on the AUD CUMEM error report, contact your next level of technical assistance.

CUMEM Recent Change Forms

This section contains illustrations of the RC/V audit forms for the CUMEM audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for CUMEM 1**

Audit Record	audrec	(1/2) (Recent Change and Verify)
1.audname:CUMEM	2.memno:1	3.prname:_____
4.timeout_int:60	5.maxseg:4000	6.utility_id:133
7.segmented:y	8.appaud:n	9.correcting:y
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:15
13.fixdpnum:y	14.prc_num:0xe	

	audrec	(2/2)
Modes of Execution		
15.rtn_mode_permit:n	16.rtn_dp_type:_____	17.rtn_dp_data:0x0
18.dmd_mode_permit:n	19.dmd_dp_type:_____	20.dmd_dp_data:0x0
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x61
24.sft_mode_permit:n	25.sft_dp_type:_____	26.sft_dp_data:0x0

auderr Form for CUMEM 1, Category 0

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:CUMEM	2.memno:1	3.category:0	4.threshold:200		
5.time_interval:18000	6.alarm_lvl:I	7.rtn_corr:100	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

auderr Form for CUMEM 1, Category 1

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:CUMEM	2.memno:1	3.category:1	4.threshold:200		
5.time_interval:18000	6.alarm_lvl:I	7.rtn_corr:100	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for CUMEM 1**

Audit Instance Info		audinst		(1/1)
				(Recent Change and Verify)
1.audname:CUMEM		2.memno:1	3.instname:_____	
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Control Unit Hardware Status Audit

The Control Unit Status (CUSTAT) audit periodically verifies the CU hardware status. Its main function is to guarantee a useful CU hardware configuration. Inconsistent hardware status and unacceptable hardware configurations are corrected when possible. The CUSTAT audit checks the following:

- Hardware status information in the System Status Registers (SSRs) and the ECD is checked for consistency and validity.
- The backup maintenance channel bits in the Hardware Status Register (HSR) of the on-line processor are also checked.

The CUSTAT interprets both a maintenance channel failure and a read off-line store failure as the other processor is being powered down. The audit does not fail any tests because the other processor is powered down as long as the other processor is not marked active or standby.

In addition, CUSTAT performs a memory comparison of active and standby main store units similar to CUMEM. Failures are reported against the on-line cache unit or the off-line CU.

Excessive error counts during the running of this audit (in correcting mode) may result in a hardware reconfiguration with one of the CUs being marked OOS.

Requesting the CUSTAT Audit

To request the CUSTAT audit, use the following command:

In PDS, **AUD:CUSTAT 1** [*a* | *b* | *a,b*]!
In MML, **AUD:CUSTAT=1** [*a* | *b* | *a,b*];

Where:

- a* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
- b* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found plus an audit-completion message. (A limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

Interpreting AUD CUSTAT Output Message and Error Reports

The AUD CUSTAT 1 output messages report the termination status of the audit and provide information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “a” is an error number that identifies the type of error that occurred. The DATA1 field in this output message contains an assert number which identifies the audit test number that failed; however, for error numbers 30, 34, or 36, the DATA1 field contains the main store address at which the test failed. Descriptions of the CUSTAT error numbers are listed in Table 14-8, and the meanings of the DATA1 assert numbers are listed in Table 14-9. The following output message format may vary among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD CUSTAT 1 ERROR a REPORT

DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors the ECD audit found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The CUSTAT audit returns four error categories. At this time, the categories are undefined and are reserved for future development.

Table 14-8. CUSTAT Audit Error Numbers

Category	Error Number	Meaning
0,3	2	Bad data in ECD.
2,3	4	ECD access function failed.
0	6	Maintenance channel failure.
0	8	Reading off-line memory via update circuit failed.
1	10	Kconfig function call failed.
0	12	CU was found to have the wrong major state in ECD.
0,1	14	On-line SSR value was incorrect.
0,1	16	Off-line SSR value was incorrect.
0,1	18	On-line HSR value was incorrect.
0	20	Off-line HSR value was incorrect.
0	22	Cache hit-miss ratio is unacceptable.
0	24	PMS database access failure.
0	26	Audit-control library function call failure.
0	28	Internal PCPAUD failure.
0	30	Off-line store access failure.
1	32	Memory audit internal error.
0	34	Off-line memory mismatch.
0	36	On-line cache memory mismatch.
0	42	EAI buffers mismatch.

Table 14-9. CUSTAT Error Reports

Assert Number	Meaning
0x1000	The RID for the on-line CU is IDNULL; stop the audit.
0x1001	Could not get the on-line CU Unit Control Block (UCB) from the ECD; stop the audit.
0x1002	The on-line CU UCB indicates device type other than DEV_CU; stop the audit.
0x1003	The RID for on-line is same as the RID for off-line; stop the audit.
0x1004	Could not get the off-line CU UCB from the ECD; stop the audit.
0x1005	The off-line CU UCB indicates device type other than DEV_CU; stop the audit.
0x1006	Could not get the on-line CU minor device chain table from the ECD; stop the audit.
0x1007	The <i>mdct</i> pointer for the on-line CU has too many <i>m_minxt</i> entries (indicates that there were more than two CUs).
0x1008	The <i>mdct</i> pointer for the on-line CU has too many <i>m_minxt</i> entries (indicates that there were more than two CUs).
0x1009	<i>m_minxt[0]</i> equals <i>m_minxt[1]</i> indicating that the CU0 and CU1 have the same RIDs; stop the audit.
0x100a	<i>m_minxt[0]</i> does not equal the RID for the on-line or off-line CU; stop the audit.
0x100b	<i>m_minxt[1]</i> does not equal the RID for the on-line or off-line CU; stop the audit.
0x1100	Faulty maintenance channel circuit.
0x1101	Off-line memory access failure.
0x1102	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x1103	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x1105	Maintenance Channel (MCH) and update circuit not functional; but the other CU is in the standby state.

Table 14-9. CUSTAT Error Reports (Contd)

Assert Number	Meaning
0x1106	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x1107	Could not change the state of the off-line CU UCB to UNAV.
0x1108	Could not change the state of the off-line CU UCB to UNAV.
0x1109	Could not get the off-line CU UCB from the ECD.
0x110b	Could not change the manual request field of the off-line CU UCB to ON.
0x110c	Could not change the state of the off-line CU UCB to OOS.
0x110d	Could not change the manual request field of the off-line CU UCB to OFF.
0x110e	Could not get the off-line CU UCB from the ECD.
0x1200	Wrong major state in the UCB for the on-line CU.
0x1201	Unable to change the on-line CU to active state.
0x1202	The off-line SSR indicates its Central Control (CC) is on-line (SSR_CC).
0x1203	PCPAUD was unable to initialize the off-line CU.
0x1204	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x1205	The off-line CU UCB has major state of active.
0x1206	PCPAUD was unable to remove the off-line CU, that is marked active, by faulting the driver.
0x1207	The on-line CU in simplex mode has the SSR_FOFL asserted.
0x1208	HSR bit 8 is asserted in the on-line HSR (backup maintenance channel bit).
0x1209	HSR bit 8 or HSR bit 9 is asserted in the on-line HSR.
0x120a	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x120b	The SSR_DSAB (IO disable) bit is asserted in the on-line SSR.
0x120c	The SSR_ISOD (isolate DMA of MAS bus) bit is asserted in the on-line SSR.

Table 14-9. CUSTAT Error Reports (Contd)

Assert Number	Meaning
0x120d	The SSR_ISC0 or SSR_ISC1 (initialization sequence control bits) is asserted in the off-line SSR.
0x1300	The on-line processor is in the update mode and the off-line processor is trying to isolate the on-line CC off the main store bus (SSR_ISOU is asserted in off-line).
0x1301	Unable to fault the driver for the off-line CU.
0x1302	The off-line processor is in the update mode or the off-line processor has SSR_ISOU asserted.
0x1303	PCPAUD was unable to initialize the off-line CU.
0x1304	PCPAUD was unable to remove the off-line standby CU by faulting the driver.
0x1305	The simplex processor in the update mode.
0x1306	The on-line processor is not isolating the other CC off the main store bus (SSR_ISOU not asserted in on-line).
0x1400	The on-line CU UCB is not marked as updated.
0x1401	Could not get the on-line CU UCB from ECD; stop the audit.
0x1402	Unable to mark the on-line CU UCB as updated.
0x1403	The off-line CU is standby, but its ECD is not marked updated or the on-line CU is not in update mode.
0x1404	Unable to fault the driver for the CU.
0x1405	The on-line CU is not in update mode, the off-line CU UCB is marked updated.
0x1406	Unable to set the off-line CU UCB to not update.
0x1500	Duplex configuration, but the on-line CU specifying simplex (SSR asserted).
0x1501	Duplex configuration, but the off-line CU specifying simplex.
0x1502	Simplex configuration, but the processor is specifying duplex.
0x1503	The off-line SSR has SSR_FONL asserted.
0x1504	The on-line CU forced on-line (SSR_FONL asserted), the off-line CU is not forced off-line (SSR_FONL not asserted).

Table 14-9. CUSTAT Error Reports (Contd)

Assert Number	Meaning
0x1505	The on-line CU not forced on-line (SSR_FONL not asserted), the off-line CU forced off-line (SSR_FOFL not asserted).
0x1506	Both the on-line and off-line CU think they are CU1 (SSR_CCID asserted).
0x1507	Both the on-line and off-line CU think they are CU0 (SSR_CCID not asserted).
0x1508	The off-line CU has the SSR_PCLR bit asserted.
0x1509	PCPAUD <i>initofl</i> function did not complete successfully.
0x150a	HSR bit 8, 9, or 31 asserted in the off-line CU.
0x150b	PCPAUD was unable to initialize the off-line CU.
0x150c	ECD access function <i>gnxt</i> failed; stop the audit.
0x150d	PCPAUD was unable to find the UCB for the on-line CU CC.
0x150e	ECD access function <i>gnxt</i> failed; stop the audit.
0x150f	PCPAUD was unable to find the UCB for the off-line CU CC.
0x1510	CU1 CC does not have <i>u_opccid</i> set in its UCB (<i>u_equip</i>).
0x1511	CU0 CC has <i>u_opccid</i> set in its UCB (<i>u_equip</i>).
0x1512	The off-line CU is CU0, but <i>u_upccid</i> is set in the CC UCB (<i>u_equip</i>).
0x1513	The off-line CU is CU1, but <i>u_opccid</i> is not set in the CC UCB (<i>u_equip</i>).
0x1514	Duplex configuration, the on-line CC UCB specifying simplex (<i>u_opsimdup</i> not set in <i>u_equip</i> field of UCB).
0x1515	Duplex configuration, the off-line CC UCB specifying simplex (<i>u_opsimdup</i> not set in <i>u_equip</i> field of UCB).
0x1516	Simplex configuration, the CC UCB specifying duplex (<i>u_opsimdup</i> set in <i>u_equip</i> field of UCB).
0x1517	The on-line CU has the SSR_PCLR bit asserted.
0x1518	ECD access of the on-line MASC UCB failed.
0x1519	ECD access of the off-line MASC UCB failed.

Table 14-9. CUSTAT Error Reports (Contd)

Assert Number	Meaning
0x151a	Memory controller type indicators inconsistent between the SSR and the MASC UCB in the off-line CU.
0x151b	Memory controller type indicators inconsistent between the SSR and the MASC UCB in the off-line CU.
0x151d	Could not get the off-line CU UCB from the ECD.
0x1600	EAI buffers in CU 0 and CU 1 mismatch.
0x1601	Unable to get the EAI IPB from the off-line CU.
0x2000	Memory comparison did not complete its entire memory block comparison.
0x2001	Kconfig function failure <i>rpterr</i> .
0x2002	Kconfig function failure <i>rpterr</i> .
0x2003	Kconfig function failure <i>fltdrvr</i> .

Under some error conditions, the CUSTAT audit may produce additional output messages. These reports have the following format:

```
REPT PCPAUD FAULT TYPE a ASSERT NO b
```

Where:

a Fault type.

b Assert number.

For a detailed explanation of the REPT PCPAUD FAULT output message and a list of fault types and assert numbers, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Responding to AUD CUSTAT Error Reports

For error numbers 4, 6, 8, 14, 16, 18, 20, 22, 24, 30, 34, 36, or 42, take the action designated in Table 14-10. For any other error numbers, contact your next level of technical assistance.

Table 14-10. AUD CUSTAT Error Messages

Error Number	Action to Be Taken
4	Run the ECD 1 audit.
6	If the off-line CU was not taken out of service, remove it. Diagnose, repair, if necessary, and restore the CU. Then switch CUs and remove, diagnose, repair, and restore the other CU.
8	The off-line CU should have been removed automatically. Diagnose, repair, if necessary, and restore the CU. Then switch CUs and remove, diagnose, repair, and restore the other CU.
14	Take action on this error only if it occurs repeatedly. Switch CUs, then remove, diagnose, repair, and restore the affected CU.
16	Take action on this error only if it occurs repeatedly. Remove, diagnose, repair, and restore the off-line CU.
18	Take action on this error only if it occurs repeatedly. Switch CUs, then remove, diagnose, repair, and restore the affected CU.
20	Take action on this error only if it occurs repeatedly. Remove, diagnose, repair, and restore the off-line CU.
22	Switch CUs. Remove the faulty CU, replace the cache memory board, diagnose the CU, and restore it. Then switch CUs again (that is, return to the original configuration) and rerun the audit.
24	Run the PMS 1 audit.
30	The off-line CU should have been removed automatically. Diagnose, repair, if necessary, and restore the CU. Then switch CUs and remove, diagnose, repair, and restore the other CU.
34	If the off-line CU was not taken out of service, remove it. Diagnose, repair, if necessary, and restore the CU.
36	If the on-line CU was not taken out of service, switch CUs and remove it. Diagnose, repair, if necessary, and restore the CU.
42	Clear the EAls with Poke 14 on the EAI page of the MTTY.

CUSTAT Recent Change Forms

This section contains illustrations of the RC/V audit forms for the CUSTAT audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for specific values.

***audrec* Form for CUSTAT 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:CUSTAT	2.memno:1	3.prname:_____	
4.timeout_int:60	5.maxseg:10	6.utility_id:133	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:15	
13.fixdpnum:y	14.prc_num:0xe		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x61	
18.dmd_mode_permit:n	19.dmd_dp_type:_____	20.dmd_dp_data:0x0	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x61	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x61	

***auderr* Form for CUSTAT 1, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:CUSTAT	2.memno:1	3.category:0	4.threshold:20		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:20		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for CUSTAT 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:CUSTAT	2.memno:1	3.category:1	4.threshold:20		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:20		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for CUSTAT 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:CUSTAT	2.memno:1	3.category:2	4.threshold:20		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:20		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for CUSTAT 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:CUSTAT	2.memno:1	3.category:3	4.threshold:3		
5.time_interval:600	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for CUSTAT 1**

Audit Instance Info		audinst		(1/1)
				(Recent Change and Verify)
1.audname:CUSTAT		2.memno:1	3.instname:_____	
4.freq:B	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

ECD Audits

The ECD audit verifies the description of hardware and selected software resources that are stored in the ECD. The audit is divided into three parts (called members), each invoked by a separate input command.

Three copies of the ECD are kept at all times, one in core memory and two on disk.

ECD 1 does a structural audit of the incore ECD.

ECD 2 does a structural audit of either the operating system (ECD) or the application (APPLCD).

ECD 3 produces a raw-data audit of the incore ECD.

The ECD audits are routinely scheduled as timed audits that run in the correcting disk ECDs mode.

Auditing the ECD requires correct information in the Low-Level Access (LLA) system. Therefore, ECD audits may invoke an LLA audit.

⇒ NOTE:

Do not run an ECD audit while a recent change is being applied or during off-line boot.

Requesting ECD Audits

To request an audit of the incore ECD, use the following command:

In PDS, **AUD:ECD a** [*b* | *c* | *b,c*]!

In MML, **AUD:ECD=a**[:*b* | :*c* | :*b,c*];

To request a structural audit of the operating system ECD, use the following command:

In PDS, **AUD:ECD 2,INS "ecd"** [*b* | *c* | *b,c*]!

In MML, **AUD:ECD=2,INS="ecd"**[:*b* | :*c* | :*b,c*];

To request a structural audit of the application ECD, use the following command:

In PDS, **AUD:ECD 2,INS "appecd"** [*b* | *c* | *b,c*]!

In MML, **AUD:ECD=2,INS="appecd"**[:*b* | :*c* | :*b,c*];

In all three commands:

- a* The type of incore audit.
1 specifies a structural audit; 2 specifies a raw-data audit.
- b* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
When the audit is run in the correcting or noncorrecting mode and if an error is found in the Data Manipulation Language (DML) structure's user rootid on the incore database, the rootid will be reset by default. This happens even though the message does not indicate that it is reset in the noncorrecting mode.
- c* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found and an audit-completion message. (Only a limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.

Interpreting AUD ECD Output Messages and Error Reports

The AUD ECD output messages report the termination status of the ECD audits and provide information about errors that were found while the audits were executing.

If the ECD 1 or ECD 3 finds errors while executing, detailed reports similar to the following are output for up to 15 errors (unless a summary report was requested). If the ECD 2 audit finds errors while executing, up to 20 detailed reports are output. The DATA1 through DATA4 fields are always zero in the AUD ECD error messages. The variable "*f*" is an error number that identifies the type of error that occurred. The variable "*g*" is a message that further describes the error that was identified by the error number.

Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

```
AUD ENV=RTR ECD a [b] ERROR f{CORR |NON CORR}  
REPORT {STARTED | INPROGRESS | COMPLETED}  
DATA1 X'00000000 DATA2 X'00000000  
DATA3 X'00000000 DATA4 X'00000000
```

NO SUPPLEMENTARY DATA

```
AUD ENV=RTR ECD a [b] ERROR f{CORR | NON CORR}  
REPORT {STARTED |INPROGRESS | COMPLETED}  
{SUPPLEMENTARY DATA BLOCK NUMBER 1  
SUPPLEMENTARY DATA FILE} g d
```

The ECD error numbers are described in Table 14-11. Complete lists of supplementary data messages may be found in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors the ECD audit found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the "System Integrity Subsystem" section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Three categories of errors are reported by the AUD ECD output message: 3, 4, and 5.

- Category 3 errors:
 - Cause the audit to stop.
 - Cannot be corrected by the audit.
 - Error number starts with a 3.

- Category 4 errors:
 - Do not stop the audit.
 - Cannot be corrected by the audit.
 - Error number starts with a 4.
- Category 5 errors:
 - Do not stop the audit.
 - Are corrected when the audit is in the correcting mode.
 - Are not corrected when the audit is not in the correcting mode.
 - Error number starts with a 5.

Table 14-11. ECD Audit Error Messages

Category	Error Number	Meaning
3	3000	Cannot open incore ECD.
3	3001	<i>malloc</i> failure.
3	3002	<i>read</i> failure.
3	3003	Audit ECD has more records than incore ECD.
3	3004	Incore ECD has more records than audit ECD.
3	3005	<i>lla__rdget</i> failure for incore ECD to check RCV semaphore.
3	3006	Audits blocked during RCV session – try later.
3	3007	<i>lla_ainit</i> failure for structural audit.
3	3008	<i>lla_irel</i> failure for incore ECD.
3	3009	<i>lla_dbrel</i> failure.
3	3010	Cannot open disk ECD.
3	3011	<i>lla_nsgrid</i> failure.
3	3012	<i>getport</i> failure.
3	3013	<i>getseg</i> failure.
3	3014	<i>lla_iatt</i> failure for incore ECD.
3	3015	<i>lla_opensq</i> failure for incore ECD.
3	3016	Audit-copy corrupted.
3	3017	<i>fopen</i> failure.

Table 14-11. ECD Audit Error Messages (Contd)

Category	Error Number	Meaning
3	3018	<i>fread</i> failure.
3	3019	<i>rewind</i> failure.
3	3020	<i>fwrite</i> failure.
3	3021	Disk ECD time stamp does not match incore ECD. Incorrect or missing audit ECD, request assistance in correcting this error.
3	3022	Driver Program Identification (PID) number notification table entries out of space.
3	3023	Makeseg failure - cannot create new segment.
3	3024	Audits blocked during off-line boot – try later.
4	4000	<i>lla_rinfo</i> failure.
4	4001	Incore RID does not match audit.
4	4002	<i>lla_rdget</i> failure.
4	4003	Record size of incore ECD does not match record size audit ECD.
4	4004	Record type of incore does not match audit ECD.
4	4005	<i>lla_uprid</i> failure.
4	4006	Unable to notify driver of local copy record changes.
4	4007	Possible disk ECD inconsistency (structure corrected).
4	4008	Structural audit cannot continue without previous corrections.
5	5000	Record corrected.
5	5001	Record needs correction.

Table 14-12 gives a complete list of supplementary data messages found on the AUD ECD output message page.

If the audits are unable to interface with the audit-control subsystem, AUD output messages cannot be issued. In that event, the supplementary data portion of each error message will be printed as a separate output message.

The LLA audit output messages come in the following types:

- DIAG — Diagnostic only
- RCVY — Error detection and associated recovery actions
- UNRCVY — Detection of an unrecoverable error.

A percent character, “%”, followed by an *x*, *lx*, *d*, and *ld* in an error message means that the appropriate value is substituted into the output message at this point.

- lx* long (4 byte) hexadecimal number
- x* integer hexadecimal number
- ld* long (4 byte) decimal number
- d* integer decimal number.

Table 14-12. LLA Audit Error Messages

Error Code	Status	Message	Description
100	RCVY	STRAUD00	Incore database sanity identification (ID) found to have incorrect value 0x%lx.
101	RCVY	STRAUD01	Offset to Software Demand Paging (SDP) SPACE structure being reset from 0x%lx to 0x%lx.
102	RCVY	STRAUD02	Offset to user data section being reset from 0x%lx to 0x%lx.
103	UNRCVY	STRAUD03	User rootid in DMLHEAD(=0x%lx) was reinitialized.
104	RCVY	STRAUD04	Startup database size being reset from 0x%lx to 0x%lx.
105	RCVY	STRAUD05	Space maxpage being reset from 0x%lx to 0x%lx.
106	RCVY	STRAUD06	Number of SDP space allocation stacks being reset from 0x%lx to 0x%lx.
107	RCVY	STRAUD07	Pointer to SDP stacks being reset from 0x%lx to 0x%lx.
108	RCVY	STRAUD08	SDP space pointer to environ being reset from 0x%lx to 0x%lx.
109	RCVY	STRAUD09	ENVIRON manager pointer being reset from 0x%lx to 0x%lx.
110	RCVY	STRAUD10	SDP manager frame pointer being reset from 0x%lx to 0x%lx.
111	RCVY	STRAUD11	SDP manager page ID being reset from 0x%lx to 0x0.
112	RCVY	STRAUD12	SDP manager use count being reset from 0x%lx to 0x1.

Table 14-12. LLA Audit Error Messages (Contd)

Error Code	Status	Message	Description
113	RCVY	STRAUD13	SDP manager next index being reset from 0x%lx to 0x1.
114	RCVY	STRAUD14	SDP manager previous index being reset from 0x%lx to 0x1.
115	RCVY	STRAUD15	SDP manager page ID being reset from 0x%lx to IDNULL.
116	RCVY	STRAUD16	SDP manager use count being reset from 0x%lx to 0x0.
117	RCVY	STRAUD17	SDP manager next index being reset from 0x%lx to 0x0.
118	RCVY	STRAUD18	SDP manager previous index being reset from 0x%lx to 0x0.
200	DIAG	SDPAUD00	Stack entry 0x%lx next ID overwritten with 0x%lx.
201	DIAG	SDPAUD01	Stack entry 0x%lx on %d list size %ld out of range.
202	RCVY	SDPAUD02	Resetting depth of %d control stack from %ld to %ld.
203	RCVY	SDPAUD03	Resetting next ID of %d control stack from 0x%lx to 0x%lx.
204	RCVY	SDPAUD04	Stack entry 0x%lx next ID being reset from 0x%lx to 0x%lx.
205	RCVY	SDPAUD05	Resetting next ID of %d control stack from 0x%lx to 0x%lx.
206	DIAG	SDPAUD06	Four-byte stack entry 0x%lx top found to have incorrect value 0x%lx.

Table 14-12. LLA Audit Error Messages (Contd)

Error Code	Status	Message	Description
207	RCVY	SDPAUD07	Resetting top itemid of four-byte stack from 0x%lx to 0x%lx.
208	RCVY	SDPAUD08	Resetting depth of four-byte stack from %ld to %ld.
209	RCVY	SDPAUD09	Resetting top of four-byte stack from 0x%lx to 0x%lx.
210	TCVY	SDPAUD10	Four-byte stack entry 0x%lx next ID being reset from 0x%lx to 0x%lx.
300	UNRCVY	DMLAUD00	Unable to load DMLHEAD structure with SDP.
301	RCVY	DMLAUD01	DML sanity ID at itemid = 0 was found to have incorrect value 0x%lx.
302	RCVY	DMLAUD02	DML page size found to have incorrect value %d, should be %d.
303	RCVY	DMLAUD03	Number of RIDs in ridblk per page found to have incorrect value %d, should be %d.
304	DIAG	DMLAUD04	Start of set header list found to have incorrect value 0x%lx.
305	DIAG	DMLAUD05	Set start points to bad set header at 0x%lx.
306	DIAG	DMLAUD06	End of set header list found to have incorrect value 0x%lx.
307	DIAG	DMLAUD07	Set end points to bad set header at 0x%lx.
308	RCVY	DMLAUD08	Set start itemid being reset to 0x%lx from 0x%lx.
309	RCVY	DMLAUD09	Set end itemid being reset to 0x%lx from 0x%lx.

Table 14-12. LLA Audit Error Messages (Contd)

Error Code	Status	Message	Description
310	DIAG	DMLAUD10	Start of RID block list found to have incorrect value 0x%lx.
311	DIAG	DMLAUD11	RID block start points to bad RID block header at 0x%lx.
312	DIAG	DMLAUD12	End of RID block list found to have incorrect value 0x%lx.
313	DIAG	DMLAUD13	RID block end points to bad RID block at 0x%lx.
314	RCVY	DMLAUD14	RID block start itemid being reset to 0x%lx from 0x%lx.
315	RCVY	DMLAUD15	RID block end itemid being reset to 0x%lx from 0x%lx.
316	RCVY	DMLAUD16	SPACE page size found to have incorrect value %d, should be %d.
317	RCVY	DMLAUD17	SPACE curleft being reset from 0x%lx to 0x%lx.
400	RCVY	SETAUD00	Set header sanity ID at 0x%lx found to have incorrect value 0x%lx.
401	RCVY	SETAUD01	Previous link field in set header at 0x%lx being reset from 0x%lx to 0x%lx.
402	RCVY	SETAUD02	Next link field in set header at 0x%lx being reset from 0x%lx to 0x%lx.
500	RCVY	RBLAUD00	RID block sanity ID at 0x%lx found to have incorrect value 0x%lx.
501	RCVY	RBLAUD01	Previous link field in RID block at 0x%lx being reset from 0x%lx to 0x%lx.
502	RCVY	RBLAUD02	Next link field in RID block at 0x%lx being reset from 0x%lx to 0x%lx.
600	RCVY	BCKAUD00	Record sanity ID at 0x%lx found to have incorrect value 0x%lx.

Table 14-12. LLA Audit Error Messages (Contd)

Error Code	Status	Message	Description
601	RCVY	BCKAUD01	Record at 0x%lx back link being reset from 0x%lx to 0x%lx.
602	RCVY	BCKAUD02	RID block slot at 0x%lx being reset from 0x%lx to 0x%lx.
603	RCVY	BCKAUD03	RID block slot 0x%lx bad, being reset from 0x%lx to 0x%lx.
700	RCVY	ORPAUD00	Resetting ridblk slot at 0x%lx to 0x%lx from 0x%lx.
800	RCVY	FREAUD00	RID block 0x%lx being put on free RID block list at head.
801	RCVY	FREAUD01	RID block 0x%lx on free list has no free slots.
900	DIAG	FRLAUD00	RID block 0x%lx next free link found to have incorrect value 0x%lx.
901	DIAG	FRLAUD01	dmlhead link to start of RID block free list found to have incorrect value 0x%lx.
1000	RCVY	FRLAUDR00	RID block 0x%lx next free link being reset from 0x%lx to 0x%lx.
1001	RCVY	FRLAUDR01	Resetting free RID block ID in dmlhead from 0x%lx to 0x%lx.
1200	UNRCVY	SHDAUD00	Set 0x%lx ordinal in both header and accessid bad.
1201	RCVY	SHDAUD01	Set 0x%lx ordinal being reset from 0x%x to 0x%x.
1202	RCVY	SHDAUD02	Set 0x%lx number of access methods being reset from %d to %d.

Table 14-12. LLA Audit Error Messages (Contd)

Error Code	Status	Message	Description
1203	RCVY	SHDAUD03	Set 0x%lx cluster flag being reset from 0x%x to 0x%x.
1204	RCVY	SHDAUD04	Set 0x%lx, access method %d, accessid being reset from 0x%lx to 0x%lx.
1205	RCVY	SHDAUD05	Set 0x%lx, access method %d, page size being reset from 0x%x to 0x%x.
1206	RCVY	SHDAUD06	Set 0x%lx, access method %d, key length being reset from %d to %ld.
1300	RCVY	DMLAUDR0	Start and end RID block links being reset from 0x%lx, 0x%lx to 0x%lx, 0x%lx.
1301	RCVY	DMLAUDR1	Start and end set header links being reset from 0x%lx, 0x%lx to 0x%lx, 0x%lx.

Responding to AUD ECD Error Reports

If the ECD audit was run in the noncorrecting mode, run it again in the correcting mode. If the error persists, notify your next level of technical assistance.

ECD Recent Change Forms

This section contains illustrations of the RC/V audit forms for the ECD audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for ECD 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:ECD	2.memno:1	3.prname:ecdaud	
4.timeout_int:6000	5.maxseg:1	6.utility_id:418	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:transient	11.sched_grp:supv	12.num_er_repts:15	
13.fixdpnum:n	14.prc_num:_____		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x7d	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x7d	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x7d	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x7d	

***audrec* Form for ECD 2**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:ECD	2.memno:2	3.prname:ecdaud	
4.timeout_int:6000	5.maxseg:1	6.utility_id:418	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:transient	11.sched_grp:supv	12.num_er_repts:15	
13.fixdpnum:n	14.prc_num:_____		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x7d	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x7d	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x7d	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x7d	

***audrec* Form for ECD 3**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:ECD	2.memno:3	3.prname:ecdaud	
4.timeout_int:6000	5.maxseg:1	6.utility_id:418	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:transient	11.sched_grp:supv	12.num_er_repts:20	
13.fixdpnum:n	14.prc_num:_____		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x7d	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x7d	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x7d	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x7d	

***auderr* Form for ECD 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:1	3.category:3	4.threshold:100		
5.time_interval:6000	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 1, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:1	3.category:4	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 1, Category 5**

Audit Error Recovery Record		auderr		(1/1)	
		(Recent Change and Verify)			
1.audname:ECD	2.memno:1	3.category:5	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 2, Category 3**

Audit Error Recovery Record		auderr		(1/1)	
		(Recent Change and Verify)			
1.audname:ECD	2.memno:2	3.category:3	4.threshold:100		
5.time_interval:6000	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 2, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:2	3.category:4	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 2, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:2	3.category:5	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 3, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:3	3.category:3	4.threshold:100		
5.time_interval:6000	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 3, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:ECD	2.memno:3	3.category:4	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for ECD 3, Category 5**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:ECD	2.memno:3	3.category:5	4.threshold:255		
5.time_interval:6000	6.alarm_lvl:I	7.rtn_corr:255	8.dmd_corr:255		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	NOACT	_____	_____	_____	_____
2	NOACT	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***audinst* Form for ECD 1**

Audit Instance Record		audinst		(1/1)	
				(Recent Change and Verify)	
1.audname:ECD	2.memno:1	3.instname:_____			
4.freq:D	5.t_sun :y	12.t_0000:y	20.t_0800:n	28.t_1600:n	
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n	
	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n	
	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n	
	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n	
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n	
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n	
		19.t_0700:n	27.t_1500:n	35.t_2300:n	

***audinst* Form for ECD 2, ecd**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:ECD	2.memno:2	3.instname:ecd		
4.freq:T	5.t_sun :y	12.t_0000:y	20.t_0800:y	28.t_1600:y
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:y	16.t_0400:y	24.t_1200:y	32.t_2000:y
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for ECD 2, appecd**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:ECD	2.memno:2	3.instname:appecd		
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:y	14.t_0200:y	22.t_1000:y	30.t_1800:y
	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:y	26.t_1400:y	34.t_2200:y
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for ECD 3**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:ECD		2.memno:3		3.instname:_____
4.freq:G	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :y	15.t_0300:y	23.t_1100:n	31.t_1900:n
	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

ECD Owner Audit

The Equipment Configuration Database Owner (ECDOWN) audit verifies the entries in the ECD manager's requester table. Basically, the ECDOWN audit ensures that no software resources are owned by an inactive or dead process. This information includes records of software resources assigned to the different active processes and devices. The ECDOWN audit is part of the ECD manager.

The ECDOWN audit is scheduled to run routinely as a timed audit in the correcting mode.

The ECDOWN audit has only one member and can be requested by either SIM or the user.

⇒ NOTE:

Do not run an ECDOWN audit while a recent change or a manual diagnostic is being applied.

Requesting the ECDOWN Audit

To request an ECDOWN audit, use the following command:

In PDS, **AUD:ECDOWN 1** [*a* | *b* | *a, b*]!
In MML, **AUD:ECDOWN=1**[*a* | *b* | *a,b*];

Where:

- a* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
- b* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found and an audit-completion message. (Only a limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.

Interpreting AUD ECDOWN Output Messages and Error Reports

The AUD ECDOWN 1 output messages report the termination status of the ECDOWN audit and provide information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “a” is an error number that identifies the type of error that occurred. The variable “b” is a supplementary data block number, and the variable “c” is an explanation of the error identified by the error number. The DATA1 through DATA4 fields are always zero for this audit. The ECDOWN error numbers are described in Table 14-13. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

```
AUD ECDOWN 1 ERROR a REPORT  
DATA 1 X'00000000 DATA 2 X'00000000  
DATA 3 X'00000000 DATA 4 X'00000000  
SUPPLEMENTARY DATA BLOCK b c
```

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors ECDOWN found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The ECDOWN audit uses two error categories.

- Category 1 — Errors are not correctable.
- Category 2 — Errors are correctable.

Table 14-13. ECDOWN Audit Error Messages

Category	Error Number	Meaning
1	100	Unable to release special device file; RID = 0x%lx.
1	101	Unable to release UCB; RID = 0x%lx.
1	102	Improper reserved entity — not Special Device File (SDF) or UCB.
2	200	Special device file released; RID = 0x%lx.
2	201	UCB released; RID = 0x%lx.
2	202	Special device file should be released; RID = 0x%lx.
2	203	UCB should be released; RID = 0x%lx.

Responding to AUD ECDOWN Error Reports

If the ECDOWN audit was run in the noncorrecting mode, rerun it in the correcting mode. In an effort to clear the problem, run the following command:

lducb *unit_number (subunit_number) (rel)*

If the error persists, call your next level of technical assistance.

ECDOWN Recent Change Forms

This section contains illustrations of the RC/V audit forms for the ECDOWN audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for ECDOWN 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:ECDOWN	2.memno:1	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:65	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:n	14.prc_num:_____		

		audrec	(2/2)
Modes of Execution			
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x5	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x5	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x5	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x5	

***auderr* Form for ECDOWN 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:ECDOWN		2.memno:1		3.category:1		4.threshold:255	
5.time_interval:65535		6.alarm_lvl:I		7.rtn_corr:255		8.dmd_corr:255	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		

***auderr* Form for ECDOWN 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:ECDOWN		2.memno:1		3.category:2		4.threshold:255	
5.time_interval:65535		6.alarm_lvl:I		7.rtn_corr:255		8.dmd_corr:255	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		

***audinst* Form for ECDOWN 1**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)		
1.audname:ECDOWN	2.memno:1	3.instname:_____		
4.freq:E	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:y
		19.t_0700:n	27.t_1500:n	35.t_2300:n

File-Manager Audits

The File Manager (FMGR) audits verify the data in the file manager task table and other internal buffers.

Nine correcting, frequency-group audits make up the FMGR family.

- FMGR 1: The FMGR task table audit checks the validity of certain fields that describe an active task, and if more than ten tasks are active, looks for active tasks that have been road blocked for more than 7 minutes. When a road-blocked task is found, it is torn down, its resources are freed, and an acknowledgment is sent to the requester. Besides being scheduled routinely with the other FMGR audits, this audit is requested by the file manager when its task table is in or near overload.
- FMGR 2: The FMGR inode audit verifies the integrity of the incore inode table by checking certain fields that track the current state of the inode (such as the status, use count, and time stamp fields). FMGR 2 compares the use count field against the number of opens on a file, and then checks the count field's status. FMGR 2 also checks the time stamp field, and if it indicates that the disk inode has not been updated within the past 3 hours, FMGR compares the disk inode with the incore inode.
- FMGR 3: The FMGR inode hash table audit verifies the integrity of the inode hash table and the inode free chain. The inode hash table chains inodes with the same hash value together to speed up the search for an inode. FMGR 3 validates these chains and ensures that each incore inode is either on the free chain or on one of the hash chains.
- FMGR 4: The FMGR file table audit verifies the integrity of the file manager's internal file table. FMGR 4 validates the fields that link a file table entry to an incore inode, to other file table entries, and to capability

table entries. (A file table entry is made for each open on a file.) FMGR 4 also compares its count field with the number of capability table entries that are linked to the count field.

- FMGR 5: The FMGR capability table audit validates the capability table. A major purpose of this audit is to detect and remove capabilities that belong to dead processes. FMGR 5 also checks the integrity of the chain of capabilities that are linked to a file table entry.
- FMGR 6: The FMGR buffer audit validates the management of the internal buffers that buffer Input/Output (IO) to or from the file manager's data or a requester's address space. The internal buffers are linked together into device hash chains to facilitate the search, and the free buffers are also linked into the free chain. FMGR 6 validates these linked chains and the buffer status. Additionally, FMGR 6 checks the validity of locks on file-system super blocks.
- FMGR 7: The FMGR mount table audit validates the inode pointers and buffer pointers in the mount table.
- FMGR 8: The FMGR delayed queue audit ensures that the open and mount delayed queues are consistent with the current number of active, open, and mount tasks.
- FMGR 9: The FMGR delayed queue audit checks for message buffer overload in the file manager's message queue.

The file manager audits are routinely scheduled in the correcting mode. Any of these audits may be manually requested, particularly if routine audit scheduling has been disabled and there is reason to suspect that a resource has been disabled.

Any errors that occur are corrected automatically by each audit.



CAUTION:

FMGR 1 and FMGR 5 are recovery audits and should not be inhibited without consulting your next level of technical support.

Requesting FMGR Audits

To request an audit of the FMGR resources, use the following command:

In PDS, **AUD:FMGR** a [:b | :c | :b,c]!
In MML, **AUD:FMGR**=a [:b | :c | :b,c];

Where:

- a* Specifies the member number of the FMGR audit.
Values = 1 through 9.

 - 1 Task queue audit
 - 2 Internal-inode-table audit
 - 3 Hash table and internal-inode-pointer-table audit
 - 4 File-table audit
 - 5 Internal-capability-table audit
 - 6 Buffer audit
 - 7 Mount-table audit
 - 8 Delayed-queue audit
 - 9 Message-queue audit.
- b* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
- c* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found plus an audit-completion message. (A limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

Interpreting AUD FMGR Output Messages and Error Reports

The AUD FMGR output messages report the termination status of the FMGR audits and provides information about errors that were found while the audits were executing.

If any of the FMGR audits find errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “*b*” is an error number that identifies the type of error that occurred. The DATA fields have different meanings for each of the FMGR audits. The FMGR error numbers are described in Table 14-14. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD FMGR a ERROR b REPORT
DATA1 X’c DATA2 X’d
DATA3 X’e DATA4 X’f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors the FMGR audit found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Table 14-14. FMGR Audit Errors

Category	Error Number	Meaning
		<i>FMGR 1 — Task queue audit</i>
5	2	Task header — invalid task ID
0	3	Task header — invalid sleep address
5	4	Task header — invalid task stack address
5	5	Task header — invalid message address
5	6	Task header — task time-out
0	7	Task header — invalid time stamp
		<i>FMGR 2 — Inode audit</i>
2	12	Inode — bad i_count value
0	13	Inode — bad i_wcount value
0	14	Inode — invalid task ID
0	15	Inode — invalid time stamp
3	16	Inode — disk/incore mismatch
2	17	Inode — invalid file table pointer
2	19	Inode — invalid mount pointer
2	25	Inode — incorrect in-use count
0	26	Inode — invalid inode table size
0	27	Inode — invalid degrowth page marker
0	28	Inode — incorrect mount table pointer
0	92	Inode — invalid open count for driver
		<i>FMGR 3 — Inode hash table audit</i>
2	21	Inode hash — bad value
4	22	Inode pointer — bad chain
4	24	Inode pointer — invalid chain pointer
		<i>FMGR 4 — File table audit</i>
0	31	File — invalid FLOCK flag
2	32	File — invalid inode pointer
2	33	File — invalid capability pointer

Table 14-14. FMGR Audit Errors (Contd)

Category	Error Number	Meaning
2	34	File — invalid file pointer
0	35	File — invalid task ID
2	36	File — bad use count of open files
0	39	File — incorrect count of file table entries in use
2	40	File — invalid free list
0	90	File — invalid file table size
0	91	File — invalid degrowth page marker
		<i>FMGR 5 — Capability table audit</i>
1	41	Capability — nonexistent process
2	42	Capability — invalid file pointer
2	43	Capability — invalid capability pointer
2	44	Capability — invalid capability pointer in file table entry
2	45	Capability — invalid capability chain
0	46	Capability — incorrect count of capability table entries currently in use
2	47	Capability — invalid free list
0	48	Capability — incorrect capability table size
0	49	Capability — incorrect degrowth page marker
		<i>FMGR 6 — Buffer audit</i>
4	51	Buffer — bad available list
4	52	Buffer — bad device chain
0	54	Buffer — invalid available list pointer
0	55	Buffer — invalid device pointer
0	56	Buffer — invalid buffer address
2	57	Buffer — invalid mount pointer
0	58	Buffer — invalid task ID
5	59	Buffer — input/output time-out
0	60	Buffer — bad time stamp
0	61	Buffer — bad super block ilist lock
0	62	Buffer — bad super block free lock
0	64	Buffer — Invalid total number of free inodes

Table 14-14. FMGR Audit Errors (Contd)

Category	Error Number	Meaning
0,2 0 2 0	72	<i>FMGR 7</i> — Mount table audit
	74	Mount — invalid super block buffer pointer
	75	Mount — invalid mount inode pointer
	77	Mount — invalid root inode pointer
2 2		<i>FMGR 8</i> — Delayed queue audit
	82	Delayed queue — bad number, open
	83	Delayed queue — bad number, mount
5		<i>FMGR 9</i> — Message queue audit
	81	Overload — Large message queue

Responding to AUD FMGR Error Reports

If any FMGR audit reports an error, take the appropriate action suggested in Table 14-15. Use the correcting (default) mode on all manual requests. If errors are still present after you have taken the indicated action, contact your next level of technical assistance.

FMGR Audit Error Categories

The FMGR audits use six error categories as follows:

- Category 0 — Errors that are easily corrected (for example, resetting a bit or individual value). There is no recovery action when the maximum is exceeded.
- Category 1 — Errors that involve multiple structures of the same audited resource (for example, rebuilding the inode pointer chain). The audit aborts when the maximum is exceeded.
- Category 2 — Errors that require searches of structures other than the type being audited and usually require restoring part of the audited resource, category 1 (for example, rebuilding the file pointers of an inode table entry). There is no recovery action when the maximum is exceeded.

- Category 3 — Errors that require input/output (for example, rereading the super block of a mounted file system). The audit aborts when the maximum is exceeded.
- Category 4 — Errors that require a total rebuild of a structure (for example, restoring a mount table entry). The recovery action when the maximum is exceeded — Phase Level 1.
- Category 5 — Errors that require a task tear down (for example, a task that has timed out). The recovery action when the maximum is exceeded — Phase Level 1.

Table 14-15. FMGR Audit Error Reports

Audit Reporting Error	Action
FMGR 1	Rerun audit.
FMGR 2	Rerun FMGR 2, then run FMGR 3.
FMGR 3	Rerun FMGR 3, then run FMGR 2.
FMGR 4	Rerun FMGR 4, then run FMGR 5.
FMGR 5	Rerun FMGR 5, then run FMGR 4.
FMGR 6	Rerun audit.
FMGR 7	Rerun audit.
FMGR 8	Rerun audit.
FMGR 9	Run FMGR 1.

FMGR Recent Change Forms

This section contains illustrations of the RC/V audit forms for the FMGR audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for FMGR 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:1	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

		audrec	(2/2)
Modes of Execution			
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x1a	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x1a	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x1a	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x1a	

***audrec* Form for FMGR 2**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:2	3.prname:_____	
4.timeout_int:600	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 3**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:3	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 4**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:4	3.pname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 5**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:5	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 6**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:6	3.prname:_____	
4.timeout_int:600	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 7**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:7	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 8**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:8	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x31	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x31	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x31	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x31	

***audrec* Form for FMGR 9**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FMGR	2.memno:9	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:evmsg	17.rtn_dp_data:0x1a	
18.dmd_mode_permit:y	19.dmd_dp_type:evmsg	20.dmd_dp_data:0x1a	
21.man_mode_permit:y	22.man_dp_type:evmsg	23.man_dp_data:0x1a	
24.sft_mode_permit:y	25.sft_dp_type:evmsg	26.sft_dp_data:0x1a	

***auderr* Form for FMGR 1, Category 0**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:FMGR	2.memno:1	3.category:0	4.threshold:50		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:16	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 1, Category 5**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:FMGR	2.memno:1	3.category:5	4.threshold:50		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:16	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 2, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:2	3.category:0	4.threshold:20		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:40		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 2, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:2	3.category:2	4.threshold:50		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:50	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 2, Category 3**

Audit Error Recovery Record						auderr (1/1) (Recent Change and Verify)
1.audname:FMGR	2.memno:2	3.category:3	4.threshold:8			
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:24			
9.error_recovery						
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	

***auderr* Form for FMGR 2, Category 5**

Audit Error Recovery Record						auderr (1/1) (Recent Change and Verify)
1.audname:FMGR	2.memno:2	3.category:5	4.threshold:8			
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:24			
9.error_recovery						
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	

***auderr* Form for FMGR 3, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:3	3.category:2	4.threshold:50		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:50	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 3, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:3	3.category:4	4.threshold:16		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:16	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 4, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:4	3.category:0	4.threshold:100		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:50	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 4, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:4	3.category:2	4.threshold:100		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:50	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 5, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:5	3.category:1	4.threshold:2		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 5, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:5	3.category:2	4.threshold:100		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:50	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 6, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.category:0	4.threshold:50		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:32	8.dmd_corr:64		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 6, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.category:1	4.threshold:8		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 6, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.category:2	4.threshold:50		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:32	8.dmd_corr:64		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 6, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.category:4	4.threshold:16		
5.time_interval:600	6.alarm_lvl:A	7.rtn_corr:16	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 6, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.category:5	4.threshold:8		
5.time_interval:600	6.alarm_lvl:A	7.rtn_corr:8	8.dmd_corr:32		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 7, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:7	3.category:0	4.threshold:16		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:16		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 7, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:7	3.category:1	4.threshold:16		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:16		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 7, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:7	3.category:2	4.threshold:16		
5.time_interval:60	6.alarm_lvl:A	7.rtn_corr:8	8.dmd_corr:16		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FMGR 7, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)							
1.audname:FMGR		2.memno:7		3.category:3		4.threshold:16					
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:8		8.dmd_corr:16					
9.error_recovery											
10.stage		11.action		12.rdata1		13.rdata2		14.raudname		15.rmemno	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	

***auderr* Form for FMGR 8, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)							
1.audname:FMGR		2.memno:8		3.category:2		4.threshold:6					
5.time_interval:60		6.alarm_lvl:A		7.rtn_corr:3		8.dmd_corr:6					
9.error_recovery											
10.stage		11.action		12.rdata1		13.rdata2		14.raudname		15.rmemno	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	
—		_____		_____		_____		_____		_____	

***auderr* Form for FMGR 9, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:9	3.category:5	4.threshold:1		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	RUNAUD	_____	_____	FMGR	1
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***audinst* Form for FMGR 1**

Audit Instance Record		audinst		(1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:1	3.instname:_____			
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n	
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n	
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n	
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n	
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n	
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n	
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n	
		19.t_0700:n	27.t_1500:n	35.t_2300:n	

***audinst* Form for FMGR 2**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR	2.memno:2	3.instname:_____		
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FMGR 3**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR	2.memno:3	3.instname:_____		
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FMGR 4**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR		2.memno:4		3.instname:_____
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FMGR 5**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR		2.memno:5		3.instname:_____
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FMGR 6**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:6	3.instname:_____	
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n
		19.t_0700:n	27.t_1500:n
			28.t_1600:n
			29.t_1700:n
			30.t_1800:n
			31.t_1900:n
			32.t_2000:n
			33.t_2100:n
			34.t_2200:n
			35.t_2300:n

***audinst* Form for FMGR 7**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)	
1.audname:FMGR	2.memno:7	3.instname:_____	
4.freq:G	5.t_sun :n	12.t_0000:n	20.t_0800:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n
		19.t_0700:n	27.t_1500:n
			28.t_1600:n
			29.t_1700:n
			30.t_1800:n
			31.t_1900:n
			32.t_2000:n
			33.t_2100:n
			34.t_2200:n
			35.t_2300:n

***audinst* Form for FMGR 8**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR	2.memno:8	3.instname:_____		
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FMGR 9**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FMGR	2.memno:9	3.instname:_____		
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

File-System Block Audit

The FSBLK audit verifies the integrity and consistency of the free block list, the block usage bit map, and the inodes of the file system specified by the instance name.

It detects out-of-range block assignments in inodes and duplicate ownership of blocks by two or more inodes. In the correcting mode, the block usage bit map, the free block list, and the free inode list are rebuilt.

The *UNIX* RTR operating system supplies ECD audit-instance records for running the FSBLK audit on the following file systems:

- */dev/root*
- */dev/etc*
- */dev/db*
- */dev/tmp*
- */dev/broot*
- */dev/betc*
- */dev/bdb*

The FSBLK audit for these file systems should be scheduled as a timed audit to be run at specified times during the day and week and when the application load is minimal.

Since the correction of errors found may alter a file in a way the application did not intend, it is suggested that the audit be scheduled to be run in the noncorrecting mode. The audit may then be run in the correcting mode after the identity of the affected files is determined. To run the FSBLK audit on another file system, the application must add an audit-instance record for that file system.

This audit may use considerable processing time, especially for a large system or on a file system with a lot of current activity.

The FSBLK audit may be run on mounted or unmounted file systems.

⇒ NOTE:

Any instance of this audit will be blocked during a field update or a program update involving the file system named in the instance. Similarly, a field or program update may not be started while the FSBLK audit of a file system affected by the update is running.

When you have reason to suspect internal damage of a file system, you may manually request FSBLK 1. Run the audit first in the noncorrecting mode. If errors are reported, and they are correctable, you can then rerun the audit in the correcting mode.

Requesting the FSBLK Audit

To request an FSBLK audit, use the following command:

```
In PDS,      AUD:FSBLK 1, INS "a" [:b | :c | :b,c] [:PARAM "d"]!  
In MML,      AUD:FSBLK=1,INS="a"[:b | :c | :b,c] [:PARAM "d"];
```

Where:

- a The instance to be audited.

The following are seven possible instance values:

```
/dev/root  
/dev/etc  
/dev/db  
/dev/tmp  
/dev/broot  
/dev/betc  
/dev/bdb
```

To run the FSBLK audit on another file system, the application must add an audit-instance record for the file system.

- b CORR or NCG — default = NCG.

CORR specifies that an audit should correct errors.

NCG specifies that an audit should not correct errors.

- c DETL or SUM — default = DETL.

DETL generates both a raw-data report for each error found and an audit-completion message. Only a limited number of raw-data reports are generated.

SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.

- d Parameter BACKUP is specified if it is desired that the audit check the file system for the appropriateness of generating a backup tape. If this is specified, the audit may report as error conditions that would not normally be considered errors but could result in the creation of an improper backup tape. These errors are not correctable.

Interpreting AUD FSBLK Output Messages and Error Reports

The AUD FSBLK 1 output messages report the termination status of the FSBLK audit and provides information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “*b*” is an error number that identifies the type of error that occurred. The FSBLK error numbers are described in Table 14-16. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

```
AUD FSBLK 1 a ERROR b REPORT  
DATA1 X'c DATA2 X'd  
DATA3 X'e DATA4 X'f
```

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors FSBLK found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The FSBLK audit uses two error categories.

- Category 1 — Errors generally indicate noncorrectable system errors apart from the audit. These errors will cause the audit to terminate prematurely. No recovery action is required.
- Category 3 — Errors indicate problems with an individual file in the file system being audited. File system corruption has occurred.

Table 14-16. FSBLK Audit Errors

Category	Error Number	Meaning
1	1	Bad super block label.
3	2	Bad file mode.
3	3	Bad file size.
3	4	Bad block — the block is out of the file-system block range.
3	5	Duplicate block number.
3	6	Bad address blocks.
2	7	A temporary file was found in the file system when the audit was invoked with PARAM=BACKUP. This condition does not represent an actual file system error, but if a backup tape is made with this condition present, the backup tape could contain file system errors. There is no corrective action that is possible or needed for this condition.
1	8	Audit initialization failed.
1	9	Unable to open device file.
1	10	Unable to get status of device file.
1	11	Device file not block type.
1	12	Unable to allocate message.
1	13	Unable to queue message.
1	14	Unable to add segments.
1	16	Unable to access block or get buffer to read block.
1	17	Unable to get free block.
1	18	Unable to access inode.
1	19	Unable to remove a segment.
1	20	Cannot mount device.
1	21	Unable to get bitmap segment ID.
1	22	Bit map IO failed.
1	26	The device file is open for writing.
1	35	The file manager segment list is full.

Responding to AUD FSBLK Error Reports

Even in the correcting mode, FSBLK does not correct all the errors it finds. Errors that are not corrected and the action that should be taken are listed in Table 14-17.

Table 14-17. Special Error Messages

Error Number	Action to Be Taken
1,2	Noncorrectable file system damage has occurred; back up the file system from the backup copy.
3,4,5,6	Use the input command OP:FNAME to print the full pathname of the damaged file. To correct the error, rerun the FSBLK audit in the correcting mode.
9,10,11	The device file for the file system does not exist or it is the wrong type. Check and repair/create the device file that defines the file system.
26	The device file is open for writing. Try again later.
8,12-22	An error in accessing the disk or in getting system resources has occurred. Check the disk status and/or system resource usage. Rerun the FSBLK audit when the system is quiet and disks are active.

FSBLK Recent Change Forms

This section contains illustrations of the RC/V audit forms for the FSBLK audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for FSBLK 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FSBLK	2.memno:1	3.prname:_____	
4.timeout_int:600	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:n	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:15	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x30	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x30	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x30	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x30	

***auderr* Form for FSBLK 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSBLK	2.memno:1	3.category:1	4.threshold:1000		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FSBLK 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSBLK	2.memno:1	3.category:3	4.threshold:39		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:20	8.dmd_corr:20		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for FSBLK 1, /dev/db**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSBLK	2.memno:1	3.instname:/dev/db		
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:y	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSBLK 1, /dev/etc**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSBLK	2.memno:1	3.instname:/dev/etc		
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:y	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

audinst Form for FSBLK 1, /dev/root

Audit Instance Record		audinst		(1/1) (Recent Change and Verify)	
1.audname:FSBLK		2.memno:1		3.instname:/dev/root	
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n	
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n	
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n	
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n	
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n	
	10.t_fri :y	17.t_0500:y	25.t_1300:n	33.t_2100:n	
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n	
		19.t_0700:n	27.t_1500:n	35.t_2300:n	

audinst Form for FSBLK 1, /dev/tmp

Audit Instance Record		audinst		(1/1) (Recent Change and Verify)	
1.audname:FSBLK		2.memno:1		3.instname:/dev/tmp	
4.freq:T	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n	
	6.t_mon :y	13.t_0100:y	21.t_0900:n	29.t_1700:n	
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n	
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n	
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n	
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n	
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n	
		19.t_0700:n	27.t_1500:n	35.t_2300:n	

***audinst* Form for FSBLK 1, /dev/bdb**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSBLK	2.memno:1	3.instname:/dev/bdb		
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSBLK 1, /dev/betc**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSBLK	2.memno:1	3.instname:/dev/betc		
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSBLK 1, /dev/broot**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSBLK		2.memno:1	3.instname:/dev/broot	
4.freq:_	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

File-System Link Audit

The FSLINK audit verifies the linkage among the directories and inodes of the file system specified by the instance name.

It detects a disagreement between the link count in an inode and the number of references to the inode in the directories of the file system. In the correcting mode, it corrects the inode link count that does not agree with the number of references, invalidates a directory entry with an invalid inode number, and moves an inode that has no directory references to a *Lost and Found* directory.

The *UNIX* RTR operating system supplies ECD audit-instance records for running the FSLINK audit on the following file systems:

- */dev/root*
- */dev/etc*
- */dev/db*
- */dev/tmp*
- */dev/broot*
- */dev/betc*
- */dev/bdb*

It is recommended that the FSLINK audit for these file systems be scheduled as a timed audit to be run at specified times during the day and week and when the application load is minimal.

Since the correction of errors found may alter a file in a way the application may not intend, it is suggested that the audit be scheduled to be run in the noncorrecting mode. The audit may then be run in the correcting mode after the identity of the affected files is determined. To run the FSLINK audit on another file system, the application must add an audit-instance record for that file system.

This audit may use considerable processing time, especially for a large file system or on a file system with a lot of current activity.

The FSLINK audit may be run on mounted or unmounted file systems.

Requesting the FSLINK Audit

When you have reason to suspect internal damage of a file system, you may manually request FSLINK 1. Run the audit first in the noncorrecting mode. If errors are reported, and they are correctable, you can then rerun the audit in the correcting mode. To request an FSLINK audit, use the following command:

```
In PDS,      AUD:FSLINK 1, INS "a" [:b | ;c | ;b,c] [:PARAM "d"]!  
In MML,      AUD:FSLINK=1,INS="a"[:b | :c | :b,c] [:PARAM "d"];
```

Where:

- a The instance to be audited.

The following are seven possible instance values:

```
/dev/root  
/dev/etc  
/dev/db  
/dev/tmp  
/dev/broot  
/dev/betc  
/dev/bdb
```

To run the FSLINK audit on another file system, the application must add an audit-instance record for the file system.

- b CORR or NCG — default = NCG.
CORR specifies that an audit should correct errors.
NCG specifies that an audit should not correct errors.
- c DETL or SUM — default = DETL.
DETL generates both a raw-data report for each error found and an audit-completion message. Only a limited number of raw-data reports are generated.
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.
- d Parameter BACKUP is specified if it is desired that the audit check the file system for the appropriateness of generating a backup tape. If this is specified, the audit may report as error conditions that would not normally be considered errors but could result in the creation of an improper backup tape. These errors are not correctable.

Interpreting AUD FSLINK Output Messages and Error Reports

The AUD FSLINK 1 output messages report the termination status of the FSLINK audit and provide information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “*b*” is an error number that identifies the type of error that occurred. The DATA fields have different meanings for different error numbers. The FSLINK error numbers are described in Table 14-18. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD FSLINK 1 a ERROR b REPORT

DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors FSLINK found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The FSLINK audit uses two error categories.

- Category 1 — Errors generally indicate noncorrectable system errors apart from the audit. These errors will cause the audit to terminate prematurely. No recovery action is required.
- Category 2 — Errors indicate problems with the file system directory structure. File-system corruption has occurred.

Table 14-18. FSLINK Audit Errors

Category	Error Number	Meaning
1	1	Bad super block label.
2	2	Mutilated inode in the inode table.
2	3	Bad directory entry.
2	4	Bad directory block (block out of the file system block range).
2	5	Bad link count.
2	6	Unreferenced inode.
2	7	Link count of directory inode less than two.
1	8	Audit initialization failed.
1	9	Cannot open device file.
1	10	Cannot get status of device file.
1	11	Device file not block type.
1	12	Cannot allocate message.
1	13	Cannot queue message.
1	14	Cannot add segments.
1	16	Cannot get block or buffer to read block.
1	18	Cannot get inode.
1	19	Unable to remove a segment.
1	20	Cannot mount device.
1	26	The device file is open for writing.
2	27	Cannot reference current directory from parent.
2	28	Current directory inode does not have a parent directory inode.
2	29	Current directory has a bad inode.
2	30	Current directory references a free inode.
2	31	Current directory references multiple parent directories.
2	32	Current directory not reachable from file-system root.
2	33	Active, but an unreferenced file was detected in the file system when the audit was invoked with PARM=BACKUP.
1	35	File manager segment list is full.

Responding to AUD FSLINK Error Reports

Even in the correcting mode, FSLINK may not correct all the errors it finds. Another process may be using the inode; therefore, the audit may not need to be run more than once to correct the errors. Errors that are detected and/or corrected and the action that should be taken are listed in Table 14-19.

Table 14-19. Special Error Messages

Error Number	Action to Be Taken
1,2,4,7	Noncorrectable file system damage has occurred. Back up the file system from the backup copy.
3,5, 27-32	Use the input command OP:FNAME to print the full pathname of the damaged file. To correct the error, run the FSLINK audit in correcting mode. If the audit has detected multiple errors, then file-system corruption has occurred. Save a copy of all FSLINK audit error messages and seek technical assistance.
6	The unreferenced file cannot be reached in the file system by any name. To correct errors, rerun the audit in the correcting (CORR) mode.
8,12,13,14,16, 18,19, and 20	A disk access error or system resource access error has occurred. Check the disk status and/or system resource usage. Rerun the FSLINK 1 audit when the system is quiet and the disks are active.
9,10,11	The device file for the file system does not exist or is the wrong type. Check and repair/create the device file that defines the file system.
15 and 17	Seek technical assistance.
26	Rerun the audit later. If error persists, seek technical assistance.
35	This error occurs when the maximum number of file manager tasks are running and using all of the available file manager segment table slots. Rerun the audit at a time when there are no other file system audits running. If the problem persists, execute a level 3 craft initialization. If the problem still persists, create an off-line memory dump tape and seek technical assistance.

FSLINK Recent Change Forms

This section contains illustrations of the RC/V audit forms for the FSLINK audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for FSLINK 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:FSLINK	2.memno:1	3.prname:_____	
4.timeout_int:600	5.maxseg:1	6.utility_id:4	
7.segmented:n	8.appaud:n	9.correcting:n	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:15	
13.fixdpnum:y	14.prc_num:0x4		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x2f	
18.dmd_mode_permit:n	19.dmd_dp_type:msg	20.dmd_dp_data:0x2f	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x2f	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x2f	

***auderr* Form for FSLINK 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSLINK	2.memno:1	3.category:1	4.threshold:1000		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FSLINK 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSLINK	2.memno:1	3.category:2	4.threshold:39		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:20	8.dmd_corr:39		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for FSLINK 1, /dev/db**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)		
1.audname:FSLINK		2.memno:1	3.instname:/dev/db	
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:y	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSLINK 1, /dev/etc**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)		
1.audname:FSLINK		2.memno:1	3.instname:/dev/etc	
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:y	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

audinst Form for FSLINK 1, /dev/root

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSLINK	2.memno:1	3.instname:/dev/root		
4.freq:T	5.t_sun :y	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:y	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:y	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:y	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

audinst Form for FSLINK 1, /dev/tmp

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSLINK	2.memno:1	3.instname:/dev/tmp		
4.freq:T	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :y	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:y	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSLINK 1, /dev/bdb**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSLINK	2.memno:1	3.instname:/dev/bdb		
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :y	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSLINK 1, /dev/betc**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSLINK	2.memno:1	3.instname:/dev/betc		
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :y	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for FSLINK 1, /dev/broot**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSLINK		2.memno:1	3.instname:/dev/broot	
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:y	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :y	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

File-System Compaction Audit

The FSCMPT audit provides a facility for merging free disk blocks of a file system to form a large free contiguous space. This will alleviate the problem of fragmentation of free disk space in a file system that interferes with the allocation of space for contiguous files. The facility to compact a file system will be executed as a file-system-correcting audit that can be invoked by the user.

An extent is a set of contiguous blocks within a file system. A file system becomes fragmented when the total number of free blocks available for extent files is larger than the size of an allocation request, but the size of each free extent is less than the required amount.

After the *UNIX* RTR operating system has been updated several times, the free blocks of the file system become distributed into small contiguous sets separated by used blocks. From this fragmented state, the compaction audit will create a large set of contiguous free blocks by merging some of the free blocks together.

To compact the file system, the compaction audit first surveys the file system to determine the following:

- The total number of free blocks in the file system
- The portions of used blocks allocated to regular and contiguous/extent files
- The size of the largest extent file in the file system.

This information together with the *max/min* parameter is used to set a compaction goal (the number of free blocks that it will attempt to merge into a single free extent). This number will be less than the total number of free blocks.

The number and distribution of *regular* file blocks has a large impact on the compaction overhead and may further limit the compaction goal. Because of the cost in system time and resources, the audit tries to avoid moving *regular* files, preferring to move contiguous/extent files instead. The *cond* parameter restricts the number of regular file blocks that will be moved to 511, the size of the free block chain from which blocks for regular files are taken. If the *ucl* parameter is specified, the number of regular file blocks that will be moved is limited only by availability of relocation space.

Each file system on which the compaction audit will be run should be carefully reviewed and thoroughly tested under appropriate load conditions. Both the outcome and performance of the file-system compaction audit is dependent on the application environment.

The following rules apply when compacting a file system:

- The audit should be run during periods of light system activity.
- The amount of activity occurring in the file system undergoing compaction should be kept to a minimum, if possible.
- The audit should NOT be run as a routine audit.

The compaction audit may be run in a noncorrecting or correcting mode. If the compaction audit is run in the noncorrecting mode, the compaction goal that has been determined is reported with an error code 2. The information in the report includes the following:

- DATA1 — Total number of free blocks in the file system.
- DATA2 — Number of contiguous free blocks to be created by compaction.
- DATA3 — Current largest number of contiguous free blocks.
- DATA4 — A projection of the number of IO jobs required to do the compaction.

Each of the DATA items is reported in hexadecimal notation.

In the correcting mode, after mapping the file system and setting the compaction goal, used blocks are relocated in order to merge the free blocks into a large free extent. At the conclusion of the compaction, an error code 3 report contains the following information:

- DATA1 — Total number of free blocks in the file system.
- DATA2 — Number of contiguous free blocks created by the compaction effort.
- DATA3 — Largest number of contiguous free blocks before compaction.
- DATA4 — The number of IO jobs involved in doing the compaction.

All error codes except 2 and 3 report error conditions that prevent the audit from completing. For the interpretation and recommended action for each error code, refer to the appropriate AUD FSCMPT output message based on your release of the UNIX RTR operating system in 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

When to Use the Compaction Audit

If an attempt to allocate a number of blocks for a contiguous or extent file fails, that is, the number allocated is less than the amount requested, the following procedures should be used.

Step	Procedure
1	Using one of the following two methods, determine whether there are sufficient free blocks in the file system that contains the file. (a) At the <i>UNIX</i> system terminal, enter: df<filesystem> (b) At the craft terminal, enter: In PDS, OP: STATUS : FREEDISK, FS "<filesystem>"! In MML, OP: STATUS : FREEDISK, FS ="<filesystem>"; If the number of free blocks is not smaller than the desired amount, continue to the next step. Otherwise, some files may need to be removed from the file system to make more free space.
2	Run the FSBLK audit in noncorrecting mode by entering: In PDS, AUD:FSBLK 1, INS "<filesystem>";NCG! In MML, AUD:FSBLK=1,INS="<filesystem>":NCG; If no errors are reported, continue to the next step. If errors are reported, refer to 254-303-111, <i>PDS Output Messages Manual</i>, 254-303-113, <i>MML Output Messages Manual</i>, or the "File-System Block Audit" section in this chapter, for interpretation and recommended action to be taken.
3	Run the FSBLK audit in correcting mode by entering: In PDS, AUD:FSBLK 1,INS "<filesystem>";CORR! In MML, AUD:FSBLK=1,INS="<filesystem>":CORR; This causes all free blocks to be returned to the pool of free blocks available for both regular and extent files. If the required number of contiguous blocks still cannot be allocated, the file-system compaction audit may be able to merge enough free blocks for the allocation.
4	Perform the "How to Use the Compaction Audit" section.

How to Use the Compaction Audit

The input message that runs the file-system compaction audit can be entered manually on the maintenance terminal or craft shell terminal.

Step	Procedure
1	Run the compaction audit in noncorrecting mode with increasingly liberal parameters as follows: <pre>AUD:FSCMPT 1,INS "<filesystem>";NCG:PARAM "MIN, COND"! AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MIN, COND"; AUD:FSCMPT 1,INS "<filesystem>";NCG:PARAM "MAX, COND"! AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MAX, COND"; AUD:FSCMPT 1,INS "<filesystem>";NCG:PARAM "MIN, UCL"! AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MIN, UCL"; AUD:FSCMPT 1,INS "<filesystem>";NCG:PARAM "MAX, UCL"! AUD:FSCMPT=1,INS="<filesystem>":NCG:PARAM="MAX, UCL";</pre> The projected number of contiguous free blocks that the compaction audit will be able to create when run in the correcting mode is reported in DATA2 of the error code 2 report. For most file systems, this projected number will increase as you progress through the list. When DATA2 in the output message is large enough, continue to the next step.
2	Run the audit in correcting mode with the pair of parameters selected. An error code 3 reports the compaction completed with DATA2 specifying the number of contiguous free blocks now available in the file system. This may be less than the amount projected in Step 1. In this case, you may rerun the audit in the correcting mode with the same parameters or return to Step 1, using the next pair of parameters in the list.

NOTE:

The file-system compaction audit can run on mounted or unmounted file systems.

Requesting the FSCMPT Audit

When running the FSCMPT audit, it should first be run in the noncorrecting mode to determine the results and cost of compacting the file system. Then it should be run in the correcting mode with the selected parameters to compact the file system. To request an FSCMPT audit, use the following command:

```
In PDS,      AUD:FSCMPT a, INS "b" [;c ;d | ;c,d] [:PARAM "{e | f | e,f}"]!  
In MML,      AUD:FSCMPT=a,INS="b" [;c ;d | ;c,d] [:PARAM="{e | f | e,f}"];
```

Where:

a Member number of the FSCMPT audit. The 3B20D/3B21D computer audits are member number 1.

b The instance to be audited. Instance names are the special device files corresponding to the file system audited.

The following are three possible instance values:

```
/dev/db  
/dev/etc  
/dev/root
```

To run the FSCMPT audit on another file system, the application must add an audit-instance record for the file system.

c CORR or NCG — default = NCG.

CORR audit should correct errors and complete the compaction.

NCG audit should not correct errors and should not do the compaction.

d DETL or SUM — default = DETL.

DETL generates both a raw-data report for each error found and an audit-completion message. Only a limited number of raw-data reports are generated.

SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.

e MIN or MAX — default = MIN.

MIN perform or report for a minimum effort compaction.

MAX perform or report for a maximum effort compaction.

f COND or UCL — default = COND.

COND limits the number of regular file blocks relocated to achieve compaction.

UCL allows an unlimited number of regular file blocks to be relocated to achieve compaction.

Interpreting AUD FSCMPT Output Messages and Error Reports

The AUD FSCMPT 1 output messages report the termination status of the FSCMPT audit and provide information about errors that were found while the audit was executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “*b*” is an error number that identifies the type of error that occurred. The DATA fields have different meanings for different error numbers. The FSCMPT error numbers are described in Table 14-20. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD FSCMPT 1 a ERROR b REPORT

DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors FSCMPT found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The FSCMPT audit uses three error categories.

- Category 1 — Errors generally indicate noncorrectable system errors apart from the audit. These errors will cause the audit to terminate prematurely. No recovery action is required.
- Category 3 — Errors indicate problems with an individual file in the file system being audited. File-system corruption has occurred.
- Category 5 — Errors are informational only and not actual errors.

Table 14-20. FSCMPT Audit Errors

Category	Error Number	Meaning
1	1	Bad super block label.
5	2	Not an error; contains information. Refer to the "How to Use the Compaction Audit" section.
5	3	Not an error; contains information. Refer to the "How to Use the Compaction Audit" section.
3	4	Bad block. The block is out of the file-system block range.
3	5	Duplicate block number.
3	6	Bad address block.
1	8	Audit library initialization failed.
1	9	Unable to open device file.
1	10	Unable to start device file.
1	11	Device file not a block device.
1	12	Unable to allocate a message.
1	13	Unable to queue a message.
1	14	Unable to add segments.
1	16	Unable to access block (or get buffer to read block).
1	17	Unable to get a free block.
1	18	Unable to access inode.
1	19	Unable to remove a segment.
1	20	Cannot mount the device.
1	21	Unable to get bit map segment ID.
1	22	Bit map IO failed.
1	23	No available structure for mapping allocated blocks for the compaction.
1	24	Concurrent file-system activity is interfering with the compaction effort.
1	25	IO attempt failed.
1	26	The device file is open for writing.
1	35	The file manager segment list is full.

Responding to AUD FSCMPT Error Reports

Even in the correcting mode, FSCMPT does not correct all the errors it finds. Errors that are not corrected and the action that should be taken are listed in Table 14-21.

Table 14-21. Special Error Messages

Error Number	Action to Be Taken
1	Error may not be corrected. Back up the file system from backup source.
4-6	For block errors run the block audit in the noncorrecting mode. If it finds an error, run the block audit in the correcting mode.
8-23	Error may not be corrected; contact your next level of technical assistance.
24	Run this audit with reduced file-system activity.
25-26	Error may not be corrected; contact your next level of technical assistance.
35	This error occurs when the maximum number of file manager tasks are running and using all of the available file manager segment table slots. Rerun the audit at a time when there are no other file system audits running. If the problem persists, execute a level 3 craft initialization. If the problem still persists, create an off-line memory dump tape and seek technical assistance.

FSCMPT Recent Change Forms

This chapter contains illustrations of the RC/V audit forms for the FSCMPT audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for FSCMPT 1**

Audit Record	audrec	(1/2) (Recent Change and Verify)
1.audname:FSCMPT	2.memno:1	3.pname:_____
4.timeout_int:600	5.maxseg:1	6.utility_id:0x4
7.segmented:n	8.appaud:n	9.correcting:y
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:15
13.fixdpnum:y	14.prc_num:0x4	

Modes of Execution	audrec	(2/2)
15.rtn_mode_permit:n	16.rtn_dp_type:msg	17.rtn_dp_data:0x2e
18.dmd_mode_permit:n	19.dmd_dp_type:msg	20.dmd_dp_data:0x2e
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x2e
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x2e

***auderr* Form for FSCMPT 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSCMPT	2.memno:1	3.category:1	4.threshold:1000		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FSCMPT 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:FSCMPT	2.memno:1	3.category:3	4.threshold:100		
5.time_interval:3600	6.alarm_lvl:A	7.rtn_corr:50	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for FSCMPT 1, Category 5**

Audit Error Recovery Record						auderr	(1/1)
						(Recent Change and Verify)	
1.audname:FSCMPT		2.memno:1		3.category:5		4.threshold:100	
5.time_interval:600		6.alarm_lvl:I		7.rtn_corr:20		8.dmd_corr:20	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		

***audinst* Form for FSCMPT 1, /dev/db**

Audit Instance Record					audinst	(1/1)
					(Recent Change and Verify)	
1.audname:FSCMPT		2.memno:1		3.instname:/dev/db		
4.freq:	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n		
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n		
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n		
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n		
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n		
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n		
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n		
		19.t_0700:n	27.t_1500:n	35.t_2300:n		

audinst Form for FSCMPT 1, /dev/etc

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSCMPT		2.memno:1	3.instname:/dev/etc	
4.freq:	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

audinst Form for FSCMPT 1, /dev/root

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:FSCMPT		2.memno:1	3.instname:/dev/root	
4.freq:T	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Memory-Manager Audits

The operating system Memory Manager (MMGR) administers all memory-allocation requests and manipulates the memory-management resources accordingly. The resources managed by the MMGR include those associated with the 3B20D/3B21D computers main memory (segments and pages) and swap space on disk. In order to ensure continuous operating system operation, it is essential that these resources be protected from loss and corruption. The MMGR audits are designed to ensure the integrity of MMGR resources and to prevent their permanent loss.

There are six MMGR audits, numbered 1, 2, 3, 4, 5, and 9. The first three audits (MMGR 1 audit, MMGR 2 audit, and MMGR 3 audit) are designed to run in order and repeatedly.

For example, a lost page table is recovered by running the MMGR 2 audit, followed by the MMGR 1 audit, followed by the MMGR 2 audit; with each audit in the correcting mode. A similar sequence is required for the recovery of lost pages. Therefore, each of the MMGR 2 and MMGR 3 audits should be rerun only after the MMGR 1 audit has run. For example, a lost page is recovered by running the MMGR 3 audit, followed by the MMGR 1 audit, followed by the MMGR 2 audit, followed by the MMGR 1 audit, followed by the MMGR 2 audit, followed by the MMGR 3 audit with each audit in the correcting mode. The routine scheduling of the MMGR audits follows this sequence. Also, the first three audits should all be allowed or all be inhibited.

- MMGR 1 audit:
 - Validates the list of free Segment Descriptor Entries (SDEs).
 - Recovers lost SDEs that are not allocated and not on the free list.
 - For each SDE for an in-memory segment, validates the linkages to its page table and the Page Descriptor Entries (PDEs) associated with its pages.
- MMGR 2 audit:
 - Validates the list of free page tables.
 - Recovers lost page tables that are not validly linked to an SDE and not on the free list.
- MMGR 3 audit:
 - Validates the list of free pages.
 - Recovers lost pages that are not linked to a segment and not on the free list.

- MMGR 4 audit:
 - Detects and completes an interrupted breakdown of a segment.
 - Releases the segment with its associated page table and pages.
- MMGR 5 audit:
 - Detects old IO locked segments. A segment slotted for release but locked in memory for an IO job is saved until the job completes and the segment is unlocked. It is possible that under certain conditions this segment may never be unlocked. After 30 minutes, it is assumed that the segment is forgotten.
 - Releases these segments.
- MMGR 9 audit:
 - Compacts disk swap space.
 - Rearranges allocated swap space to provide a large free space consisting of contiguous blocks. This audit is run only when manually invoked, usually in response to repeated REPT MMGR004 output messages. These messages signal that the available free swap space fragmented into sets of contiguous free blocks too small to satisfy a request and that an attempt has been made to compact the swap space cheaply.

The compaction done by the audit, however, is not cheap. While this compaction is running, the memory manager is not able to swap supervisor processes into memory. As a result, the only supervisor processes that run during this interval are those already in memory. The real-time interval varies with system activity, usually several seconds.

Requesting the MMGR Audits

To request a MMGR 1 through 5 audit, use the following command:

In PDS, **AUD:MMGR a [;b | ;c | ;b,c] !**
In MML, **AUD:MMGR=a[:b | :c | :b,c] ;**

Where:

- a Specifies member number of the audit (1 – 5):
- 1 Segment descriptor table audit
 - 2 Page table audit
 - 3 Page descriptor table audit
 - 4 Segment release audit
 - 5 Segment unlock audit.

- b* CORR or NCG — default = CORR.
CORR — Audit should correct errors.

NCG — Audit does not correct errors.
- c* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found followed by an audit-completion message. (A limited number of reports are generated.)

SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

The MMGR 9 audit runs only when manually invoked. It is usually requested when repeated REPT MMGR004 output messages.

In PDS, **AUD:MMGR 9!**
In MML, **AUD:MMGR=9;**

Interpreting AUD MMGR Output Messages and Error Reports

The AUD MMGR output messages report the termination status of the MMGR audits and provide information about errors that were found while the audits were executing.

If the MMGR 1 audit finds errors while executing, detailed reports similar to the following report are output for up to ten errors (unless a summary report was requested). If the MMGR 9 audit finds work to do, a detailed report containing associated information is output. The variable “*b*” is an error number that identifies the type of error that occurred. The DATA fields mean different things for different error numbers. The MMGR error numbers are described in Table 14-22. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD MMGR a ERROR b REPORT
DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors MMGR found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the "System Integrity Subsystem" section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

The MMGR audit uses six error categories.

- Category 0 — Used only for the swap space compaction audit (**AUD:MMGR 9**) and has no real meaning.
- Category 1 — Resets Page Table (PGT), PDE, and SDE status flags; corrects global free counts; corrects SDE structure values.
- Category 2 — Adds unreferenced PDEs/PGTs to free lists.
- Category 3 — Corrects global pointers and counts; releases SDEs and associated resources; corrects PDE structure values.
- Category 4 — Rebuilds SDE, PGT, and PDE free lists. For segments with invalid page entries, page overlaps, and for segment overlaps, the condition is cleared and associated killable processes are terminated.
- Category 5 — Assigned to messages which are not currently in use and has no real meaning at this time.

Table 14-22. MMGR Audit Error Messages

Category	Error Number	Meaning
3	10	One or more of the following globals contain incorrect values: pointer to first segment descriptor, pointer to last segment descriptor, and/or number of segment descriptors.
3	20	One or more of the following globals contain incorrect values: pointer to first page table, pointer to last page table, and/or number of page tables.
3	30	One or more of the following globals contain incorrect values: pointer to first page descriptor and/or pointer to last page descriptor.
4	101	The SDE free list contains an invalid link.
4	102	An SDE free list member is allocated.
1	103	The global count of free SDEs contains an incorrect value.
4	104	The SDE free list is circular.
1	105	The on-free-list bit is not set in an SDE that is on the free list.
3	106	An SDE with zero status and not linked to any page table is not on the free list.
1	107	An SDE with nonzero status or valid page table pointer is not marked allocated.
1	108	An SDE is linked to an empty page table.
1	109	A page table linked to an SDE is not marked allocated.
1	110	An SDE contains an invalid page table pointer.
3	112	The PDE for a page linked to an SDE does not contain the correct segment index.
3	113	The PDE for a page linked to an SDE does not contain the correct page index.
1	114	The in-memory page count in an SDE is incorrect.

Table 14-22. MMGR Audit Error Messages (Contd)

Category	Error Number	Meaning
1	115	The PDE for a page linked to an SDE is not marked allocated.
4	116	The page table for a segment contains an invalid page address. No process that has this segment in its address space is nonkillable.
4	118	A segment contains the same physical page of memory at two different virtual addresses. No process that has this segment in its address space is nonkillable.
4	120	Two or more in-memory segments contain the same physical page of memory. No process that has the segment(s) in its address space is nonkillable.
4	122	Two or more in-memory segments have the same page table and, therefore, contain the same pages of memory. No process that has the segment(s) in its address space is nonkillable.
4	201	The list of free page tables contains an invalid link.
4	202	A page table on the free list is allocated.
1	203	The global count of free page tables contains an incorrect value.
4	204	The linked list of free page tables is circular.
1	205	The on-free-list bit is not set in a page table that is on the free list.
2	206	A page table is neither linked to an SDE nor is on the free list.
4	301	The list of free pages contains an invalid link index.

Table 14-22. MMGR Audit Error Messages (Contd)

Category	Error Number	Meaning
4	302	The PDE for a page on the free list is allocated.
4	303	The global count of free pages contains an incorrect value.
4	304	The free page list is circular.
1	305	The on-free-list bit is not set in the PDE for a page that is on the free list.
2	306	A page is neither linked to an SDE nor on the free list.
3	401	The breakdown (release) of a segment did not complete.
3	501	A segment has been IO locked for 30 minutes or more.
0	1	The MMGR 9 audit has completed the compaction of swap space — not a real error.
0	2	The MMGR 9 audit was invoked in the noncorrecting mode which prevents the audit from doing anything.
0	3	The MMGR 9 audit was unable to move a segment. Therefore, it was unable to complete the swap space compaction.

Responding to MMGR Audit Reports

All the MMGR audits are correcting audits. Unless run in noncorrecting mode, they correct all the errors they find with the exception of those in category 5. The errors in category 5 involve corruption of memory segments accessed by nonkillable processes. The only way to remove these errors is a level 2 or above phase. This severe recovery action may not be desirable for an application or at all periods of time. Therefore, it is recommended that in the database audit records, the error threshold be set at 1 (a single such error) and the recovery action be set to RPTAIM (report the audit results to AIM). The AIM process can then choose a recovery appropriate to the current conditions. It may ignore the error, call a Phase 2, or postpone a phase to a convenient time.

MMGR Recent Change Forms

This section contains illustrations of the RC/V audit forms for the MMGR audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for MMGR 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:1	3.prname:_____	
4.timeout_int:200	5.maxseg:250	6.utility_id:6	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x50	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x50	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x50	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x50	

***audrec* Form for MMGR 2**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:2	3.prname:_____	
4.timeout_int:200	5.maxseg:250	6.utility_id:6	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x50	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x50	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x50	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x50	

***audrec* Form for MMGR 3**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:3	3.prname:_____	
4.timeout_int:200	5.maxseg:750	6.utility_id:6	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

Modes of Execution			audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x50		
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x50		
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x50		
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x50		

***audrec* Form for MMGR 4**

Audit Record			audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:4	3.prname:_____		
4.timeout_int:120	5.maxseg:100	6.utility_id:6		
7.segmented:y	8.appaud:n	9.correcting:y		
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10		
13.fixdpnum:y	14.prc_num:0x6			

Modes of Execution			audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x50		
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x50		
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x50		
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x50		

***audrec* Form for MMGR 5**

Audit Record	audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:5	3.prname:_____
4.timeout_int:120	5.maxseg:100	6.utility_id:6
7.segmented:y	8.appaud:n	9.correcting:y
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10
13.fixdpnum:y	14.prc_num:0x6	

Modes of Execution	audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x50
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x50
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x50
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x50

***audrec* Form for MMGR 9**

Audit Record	audrec	(1/2) (Recent Change and Verify)
1.audname:MMGR	2.memno:9	3.prname:_____
4.timeout_int:60	5.maxseg:1	6.utility_id:0
7.segmented:n	8.appaud:n	9.correcting:y
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:1
13.fixdpnum:y	14.prc_num:0x1	

audrec (2/2)

Modes of Execution

15.rtn_mode_permit:n	16.rtn_dp_type:_____	17.rtn_dp_data:0x0
18.dmd_mode_permit:n	19.dmd_dp_type:_____	20.dmd_dp_data:0x0
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0xf
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0xf

***auderr* Form for MMGR 1, Category 1**

auderr (1/1)
(Recent Change and Verify)

Audit Error Recovery Record

1.audname:MMGR	2.memno:1	3.category:1	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:1	3.category:2	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:0		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:1	3.category:3	4.threshold:20		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 1, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:1	3.category:4	4.threshold:5		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:1	8.dmd_corr:10		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—

***auderr* Form for MMGR 1, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:1	3.category:5	4.threshold:1		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:1	8.dmd_corr:5		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	RPTAIM	—	—	—	—
2	RPTAIM	—	—	—	—
1	SAMAUD	—	—	—	—
1	PANDMP	—	—	—	—
—	—	—	—	—	—
—	—	—	—	—	—

***auderr* Form for MMGR 2, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:2	3.category:1	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 2, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:2	3.category:2	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 2, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:2	3.category:3	4.threshold:20		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:8	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 2, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:2	3.category:4	4.threshold:1000		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 3, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:3	3.category:1	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 3, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:3	3.category:2	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 3, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)							
1.audname:MMGR		2.memno:3		3.category:3		4.threshold:20					
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:8		8.dmd_corr:50					
9.error_recovery											
10.stage		11.action		12.rdata1		13.rdata2		14.raudname		15.rmemno	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	

***auderr* Form for MMGR 3, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)							
1.audname:MMGR		2.memno:3		3.category:4		4.threshold:1000					
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:1		8.dmd_corr:1					
9.error_recovery											
10.stage		11.action		12.rdata1		13.rdata2		14.raudname		15.rmemno	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	
—		—		—		—		—		—	

***auderr* Form for MMGR 4, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:4	3.category:3	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:30	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 5, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MMGR	2.memno:5	3.category:3	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:30	8.dmd_corr:100		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MMGR 9, Category 0**

Audit Error Recovery Record						auderr	(1/1)
						(Recent Change and Verify)	
1.audname:MMGR		2.memno:9		3.category:0		4.threshold:32767	
5.time_interval:1		6.alarm_lvl:I		7.rtn_corr:1		8.dmd_corr:1	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	

***audinst* Form for MMGR 1**

Audit Instance Record						audinst	(1/1)
						(Recent Change and Verify)	
1.audname:MMGR		2.memno:1		3.instname:_____			
4.freq:E	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n			
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n			
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n			
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n			
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n			
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n			
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n			
		19.t_0700:n	27.t_1500:n	35.t_2300:n			

***audinst* Form for MMGR 2**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:MMGR	2.memno:2	3.instname:_____		
4.freq:E	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for MMGR 3**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:MMGR	2.memno:3	3.instname:_____		
4.freq:E	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for MMGR 4**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:MMGR	2.memno:4	3.instname:_____		
4.freq:F	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for MMGR 5**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:MMGR	2.memno:5	3.instname:_____		
4.freq:F	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for MMGR 9**

Audit Instance Record		audinst (1/1) (Recent Change and Verify)		
1.audname:MMGR		2.memno:9	3.instname:_____	
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Message-Buffer Audits

The Message Buffer (MSGBUF) audits verify the allocation of the units carrying messages among the active operating system processes. The MSGBUF audits prevent faults and message buffer overloads by correcting damaged message queues and performing garbage collection on old messages. MSGBUF has two audits (called members): one is called the message-buffer-queue audit and the other is called the message-buffer-extender-block audit. In most cases, the audit corrects the errors it finds.

The MSGBUF is routinely initiated by SIM at intervals that depend on the number of other audits scheduled. Either member can also be initiated by the user.

Only one member of MSGBUF can be requested at a time. If you want to run both members, input two requests.

The MSGBUF audits are frequency-group audits that run in the correcting mode.

Requesting MSGBUF Audits

To request an audit of the message buffers, use the following command:

In PDS, **AUD:MSGBUF a [;b | ;c | ;b,c] !**
In MML, **AUD:MSGBUF=a[:b | :c | :b,c] ;**

Where:

- a* Specifies the type of MSGBUF audit by member number (1 or 2).
 1 is the Message-buffer-queue audit.
 2 is the Message-buffer-extender-block audit.
- b* CORR or NCG — default = CORR.
 CORR specifies that an audit should correct errors.
 NCG specifies that an audit does not correct errors.
- c* DETL or SUM — default = DETL.
 DETL generates a raw-data report for each error found and an audit-completion message. (Only a limited number of raw-data reports are generated.)
 SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors found.

Interpreting AUD MSGBUF Output Messages and Error Reports

The AUD MSGBUF output messages report the termination status of the MSGBUF audits and provide information about errors the audits found while executing.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). The variable “*b*” is an error number that identifies the type of error that occurred. The DATA fields mean different things for different errors. The MSGBUF error numbers are described in Table 14-23. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

AUD MSGBUF a ERROR b REPORT

DATA1 X'c DATA2 X'd
DATA3 X'e DATA4 X'f

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors MSGBUF found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section. For more detailed information, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

If MSGBUF found an error, a code number is output. Table 14-23 gives the error numbers and their meanings. The error categories are defined as follows.

For the message-buffer-queue audit (Error Number Series 100):

- Category 0 — Errors corrected by updating a field in either a message buffer, a message extender, or a process Dispatcher Control Table Entry (DCTE).
- Category 1 — Errors corrected by removing a message from a queue and returning the message to the original sender, requeuing the message, or terminating a process.
- Category 2 — Not used.

- Category 3 — Not used.
- Category 4 — Not used.
- Category 5 — Errors corrected by rebuilding a process message queue.

For the message-buffer-extender-block audit (Error Number Series 200):

- Category 0 — Errors corrected by updating a field in a message buffer, freeing a message, returning a message to the original sender, or returning invalid data to the audit in an audit-fault message.
- Category 1 — Error corrected by returning message to original sender.
- Category 2 — Error corrected by freeing a nonqueued message.
 - owned by supervisor process
 - owned by kernel-audit process
 - owned by the kernel
 - in response to audit-fault message.
- Category 3 — Errors corrected by faulting a process or terminating a killable process.
- Category 4 — Not used.
- Category 5 — Not used.

Table 14-23. MSGBUF Audit Error Messages

Category	Error Number	Meaning
5	110	Bad message queue head pointer.
5	111	Bad message queue link pointer.
5	112	Allocation bit not set in message extender.
0	113	Bad on-queue bit in message extender.
0	114	Allocation bit not set in message header.
1	115	Bad message-to field in message and bad message owner in message extender.
0	116	Bad message owner in message extender.
0	117	Bad message-to field in message.
5	118	Message queue out of order.
1	119	Old audit-fault message queued to a nonkillable process.
1	120	Old audit-fault message queued to a killable process.
1	121	Old queued message (except an old queued audit-fault message or an old acknowledgment message queued to a supervisor process).
5	122	Process has a circular message queue.
5	123	Bad message queue tail pointer.
0	124	Bad message queue block count in process DCTE.
1	125	Old acknowledgment message queued to a nonkillable supervisor process.
1	126	Old acknowledgment message queued to a killable supervisor process.

Table 14-23. MSGBUF Audit Error Messages (Contd)

Category	Error Number	Meaning
0	200	Allocated message with an invalid owner field.
2	201	Old allocated message.
0	214	Allocation bit is not set in message header.
1	216	Queued message with an invalid owner.
3	219	Old audit-fault message queued to a nonkillable process.
3	220	Old audit-fault message queued to a killable process.
1	221	Old queued message (except an old queued audit-fault message).
0	270	Dequeued message with an invalid owner field.
3	271	Old dequeued audit-fault message owned by a nonkillable process.
3	272	Old dequeued audit-fault message owned by a killable process.
2	273	Old dequeued message (except an audit-fault message).
3	274	Over nine old nonqueued messages (excluding audit-fault messages) owned by a killable process.
0	290	Audit-fault message response has an invalid message status.
0	291	Invalid status for a message pointer in an audit-fault message.

Responding to AUD MSGBUF Error Reports

The MSGBUF audit corrects most errors it finds. Errors that indicate that some corrective or investigative action should be taken are listed in Table 14-24.

Table 14-24. Special Error Messages

Error Number	Action to Be Taken
119	Notify your next level of technical assistance because the error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
120	Notify your next level of technical assistance because the error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
121	Notify your next level of technical assistance because the error indicates that a process may be having problems in processing its messages.
125	Notify your next level of technical assistance because the error indicates that a process has probably mishandled an acknowledgment message.
126	Notify your next level of technical assistance because the error indicates that a process has probably mishandled an acknowledgment message.
200	The error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process-administration audit (PROAD 1) to complete the process termination.
201	Notify your next level of technical assistance because the error indicates that a process may be having problems in processing its messages.
216	The error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process-termination audit (PROAD 1) to complete the process termination.

Table 14-24. Special Error Messages (Contd)

Error Number	Action to Be Taken
219	This error indicates that a process has probably mishandled an audit-fault message. Run the message-buffer-queue audit (MSGBUF 1); if the error is not found and corrected, then run the message-buffer-extender audit (MSGBUF 2) again. Notify your next level of technical assistance for corrective action. It is likely that errors 201 and 273 will not be corrected if this error occurs.
220	This error indicates that a process has probably mishandled an audit-fault message. Run the message-queue audit (MSGBUF 1); if the error is not found and corrected, then run the message-extender audit (MSGBUF 2) again. Notify your next level of technical assistance for corrective action. It is likely that errors 201 and 273 will not be corrected if this error occurs.
221	This error indicates that a process may be having problems processing its messages. Run the message-queue audit (MSGBUF 1); if the error is not found and corrected, then run the message-extender audit (MSGBUF 2) again.
270	This error indicates a problem in the ownership of a message. This may occur if a breakdown happens in a process termination. Run the process-administration audit (PROAD 1) to complete the process termination.
271	Notify your next level of technical assistance because this error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
272	Notify your next level of technical assistance because this error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
273	Notify your next level of technical assistance because this error indicates that a process may be having problems in processing its messages.

Table 14-24. Special Error Messages (Contd)

Error Number	Action to Be Taken
274	Notify your next level of technical assistance because this error indicates that a process may be having problems in processing its messages.
290	Notify your next level of technical assistance because this error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.
291	Notify your next level of technical assistance because this error indicates that a process has probably mishandled an audit-fault message. It is likely that errors 201 and 273 will not be corrected if this error occurs.

MSGBUF Recent Change Forms

This section contains illustrations of the RC/V audit forms for the MSGBUF audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for MSGBUF 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:MSGBUF	2.memno:1	3.prname:_____	
4.timeout_int:60	5.maxseg:75	6.utility_id:6	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

		audrec	(2/2)
Modes of Execution			
15.rtn_mode_permit:y	16.rtn_dp_type:event	17.rtn_dp_data:0x200000	
18.dmd_mode_permit:y	19.dmd_dp_type:event	20.dmd_dp_data:0x20	
21.man_mode_permit:y	22.man_dp_type:event	23.man_dp_data:0x8000	
24.sft_mode_permit:y	25.sft_dp_type:event	26.sft_dp_data:0x10	

***audrec* Form for MSGBUF 2**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:MSGBUF	2.memno:2	3.prname:_____	
4.timeout_int:60	5.maxseg:250	6.utility_id:6	
7.segmented:y	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x52	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x52	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x52	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x52	

***auderr* Form for MSGBUF 1, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MSGBUF	2.memno:1	3.category:0	4.threshold:60		
5.time_interval:60	6.alarm_lvl:M	7.rtn_corr:20	8.dmd_corr:6144		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MSGBUF 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MSGBUF	2.memno:1	3.category:1	4.threshold:50		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:20	8.dmd_corr:2048		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MSGBUF 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:MSGBUF		2.memno:1		3.category:2		4.threshold:10	
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:3		8.dmd_corr:1024	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	

***auderr* Form for MSGBUF 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:MSGBUF		2.memno:1		3.category:3		4.threshold:10	
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:3		8.dmd_corr:1024	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	

***auderr* Form for MSGBUF 1, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MSGBUF	2.memno:1	3.category:5	4.threshold:2		
5.time_interval:60	6.alarm_lvl:M	7.rtn_corr:1	8.dmd_corr:1024		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MSGBUF 2, Category 0**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:MSGBUF	2.memno:2	3.category:0	4.threshold:30		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:2048		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for MSGBUF 2, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:MSGBUF		2.memno:2		3.category:1		4.threshold:24	
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:8		8.dmd_corr:2048	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
1	RUNAUD	_____	_____	MSGBUF	1		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		

***auderr* Form for MSGBUF 2, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)			
1.audname:MSGBUF		2.memno:2		3.category:2		4.threshold:20	
5.time_interval:60		6.alarm_lvl:I		7.rtn_corr:5		8.dmd_corr:2048	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		
-	_____	_____	_____	_____	_____		

***auderr* Form for MSGBUF 2, Category 3**

Audit Error Recovery Record						auderr	(1/1)
						(Recent Change and Verify)	
1.audname:MSGBUF		2.memno:2		3.category:3		4.threshold:6	
5.time_interval:60		6.alarm_lvl:M		7.rtn_corr:2		8.dmd_corr:2048	
9.error_recovery							
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno		
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	
-	_____	_____	_____	_____	_____	_____	

***audinst* Form for MSGBUF 1**

Audit Instance Record					audinst	(1/1)
					(Recent Change and Verify)	
1.audname:MSGBUF		2.memno:1		3.instname:_____		
4.freq:C	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n		
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n		
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n		
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n		
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n		
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n		
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n		
		19.t_0700:n	27.t_1500:n	35.t_2300:n		

***audinst* Form for MSGBUF 2**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:MSGBUF		2.memno:2	3.instname:_____	
4.freq:E	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Plant Measurements System Audits

The PMS audits verify the integrity of the incore PMDB and check the minimum number and types of sets and records in the PMDB. The PMS audits include an LLA audit and a record-integrity audit.

If PMS 1 finds errors in the incore PMDB, it retrieves a copy of the records that were mutilated from one of two PMDB copies on disk. If both PMDB copies are found to be mutilated, then PMS 1 creates a new copy and copies it to incore and disk.

⇒ NOTE:

Of the two PMS audits (PMS 1 and PMS 2), only PMS 1 can be requested manually. PMS 2 is a recovery audit that runs automatically when PMS 1 cannot correct errors it finds.

Requesting PMS Audits

If the PMS audits are unable to interface with the audit-control subsystem, one of the following error messages is output:

- PMSAUD — PLIB FAILURE
- PMSAUD — MAKESEG FAILURE
- PMSAUD — MSGENAB FAILURE
- PMSAUD — CANNOT RECEIVE MESSAGE FROM SIM.

When this happens, you may want to manually request the PMS 1 by using the following command:

In PDS, **AUD:PMS 1** [*a* | *b* | *a,b*]!
In MML, **AUD:PMS=1** [*a* | *b* | *a,b*];

Where:

- a* CORR or NCG — default = CORR.
CORR specifies that an audit should correct errors.
NCG specifies that an audit does not correct errors.
- b* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found plus an audit-completion message. (A limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

Interpreting AUD PMS Output Messages and Error Reports

The AUD PMS output messages report the termination status of the PMS audit and provide information about errors the audit found while executing.

When the manual audit, PMS 1, finds serious errors which it cannot correct, SIM demands that this audit be rerun to perform the major recovery. This causes the database to lock, and the output message REPT SIMCHK 161 to print. Major recovery actions can include retrieval of a backup copy of the database from disk or creation of a new database if no valid backup copy exists.

If the audit finds errors while executing, detailed reports similar to the following report are output (unless a summary report was requested). For detailed explanations of the AUD PMS output messages, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*. The variable “a” is the audit member number. The variable “b” is an error number that identifies the type of error that occurred. The variable “c” is a supplementary data block number, and the variable “d” is an explanation of the error identified by the error number. The DATA fields are always zero for PMS error messages. The PMS error numbers are described in Table 14-25.

AUD PMS a ERROR b REPORT

DATA1 X'00000000 DATA2 X'00000000

DATA3 X'00000000 DATA4 X'00000000

SUPPLEMENTARY DATA BLOCK NUMBER c d

Additional information can be found in the Supplementary Data portion of each message. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for complete lists of supplementary data messages. Generally, the following output message format will differ among releases. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for your specific format.

When the audit terminates, one of three audit-completion messages is generated that reports whether the audit COMPLETED, ABORTED, or STOPPED.

- If the audit COMPLETED, the audit-completion message contains the number of errors PMS found and the number of errors it corrected.
- If the audit ABORTED, the audit-completion message contains an abort code that indicates why the audit aborted.
- If the audit STOPPED, the audit-completion message simply states the audit stopped.

The basic formats of the AUD output messages are described in the “System Integrity Subsystem” section in this IP and 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Table 14-25. PMS Audit Error Messages

Category	Error Number	Meaning
0	0	<i>lcall</i> failure.
0	1	Fatal, cannot create new database.
0	2	Fatal, cannot copy new disk database to incore.
0	3	New database created; reboot the system.
1	1000	<i>setid__rec</i> mutilated.
1	1001	Requesting demand audit to recover.
2	2000	<i>setid__rec</i> mutilated.
2	2001	Fatal, incore and disk databases mutilated, new database will be created.
3	3000	Invalid return value from <i>lla__audit</i> .
3	3001	Invalid dispatched message.
3	3002	Cannot release the incore database.
3	3003	Cannot release the disk database.
3	3004	Cannot unlock incore database.
3	3005	<i>setid__rec</i> corrected.
3	3006	<i>lla__gtrid</i> failure.
3	3007	Invalid set type.
3	3008	Cannot lock the incore database.
3	3009	Disk database written to incore.
4	4000	Noncorrectable internal structure.
5	5001	Cannot attach to incore database.

Table 14-25. PMS Audit Error Messages (Contd)

Category	Error Number	Meaning
5	5002	<i>lla__ainit</i> failure.
5	5004	Cannot unlink <i>/etc/pmdb.aud</i> .
5	5005	Invalid execution mode.
5	5006	Invalid audit member number.
5	5007	Cannot attach to primary disk database.
5	5008	<i>lla__opensq</i> failure.
5	5009	Cannot get <i>setid__rec</i> .
5	5011	<i>lla__rdget</i> failure.
5	5012	<i>lla__uprid</i> failure.
5	5013	<i>lla__opnset</i> failure.
5	5014	<i>lla__seqset</i> failure.
5	5016	Cannot open <i>/etc/pmdb</i> .
5	5017	Cannot open <i>/etc/pmdb.aud</i> .
5	5018	Invalid disk name.
5	5019	Invalid audit type.
5	5020	Failed to write incore database to disk.
5	5021	<i>fstat</i> failure.
5	5022	Failed to write disk database to incore.
5	5023	<i>plib</i> failure.
5	5024	<i>makeseg</i> failure.
5	5025	<i>msgenab</i> failure.

Table 14-25. PMS Audit Error Messages (Contd)

Category	Error Number	Meaning
5	5026	Cannot receive the message from SIM.
5	5027	Internal structure corrected, refer to <i>/etc/log/pmsaud</i> .
5	5028	Cannot attach to secondary disk database – <i>/etc/pmdb.aud</i> .
5	5029	Cannot falloc disk database.
5	5030	Restored incore database from disk.
5	5031	Primary disk database is bad.
5	5032	Secondary disk database is bad.
5	5033	Restored primary disk database.
5	5034	Restored secondary disk database.

Responding to AUD PMS Error Reports

If the AUD PMS error code indicates a bad primary and/or secondary disk database(s) and the condition is not corrected when the audit is run in the correcting mode, check to make sure enough contiguous space exists on the file system to create the database(s). Seek technical assistance to do this and to correct the problem. Once the problem is corrected, rerun the audit in the correcting mode.

If the audit was run in the noncorrecting mode, rerun the audit in the correcting mode. If errors still occur, save the output and contact your next level of technical assistance.

PMS Recent Change Forms

This section contains illustrations of the RC/V audit forms for the PMS audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for PMS 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:PMS	2.memno:1	3.prname:pmsaud	
4.timeout_int:600	5.maxseg:1	6.utility_id:355	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:transient	11.sched_grp:supv	12.num_er_repts:15	
13.fixdpnum:n	14.prc_num:_____		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:n	16.rtn_dp_type:msg	17.rtn_dp_data:0x0	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x40	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x0	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x0	

***audrec* Form for PMS 2**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:PMS	2.memno:2	3.prname:pmsaud	
4.timeout_int:600	5.maxseg:1	6.utility_id:355	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:transient	11.sched_grp:supv	12.num_er_repts:10	
13.fixdpnum:n	14.prc_num:_____		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:n	16.rtn_dp_type:_____	17.rtn_dp_data:0x0	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x40	
21.man_mode_permit:n	22.man_dp_type:_____	23.man_dp_data:0x0	
24.sft_mode_permit:n	25.sft_dp_type:_____	26.sft_dp_data:0x0	

***auderr* Form for PMS 1, Category 1**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:PMS	2.memno:1	3.category:1	4.threshold:1		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:1	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	SMAUD	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PMS 1, Category 2**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:PMS	2.memno:1	3.category:2	4.threshold:3		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:3	8.dmd_corr:3		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
2	RUNAUD	_____	_____	PMS	2
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PMS 1, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PMS	2.memno:1	3.category:3	4.threshold:15		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:15	8.dmd_corr:15		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	SAMAUD	_____	_____	PMS	1
2	RUNAUD	_____	_____	PMS	2
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for PMS 1, Category 4**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PMS	2.memno:1	3.category:4	4.threshold:1		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:2	8.dmd_corr:2		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	SAMAUD	_____	_____	PMS	1
2	RUNAUD	_____	_____	PMS	2
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____
—	_____	_____	_____	_____	_____

***auderr* Form for PMS 1, Category 5**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:PMS	2.memno:1	3.category:5	4.threshold:30		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:30	8.dmd_corr:30		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
1	SAMAUD	_____	_____	PMS	1
2	RUNAUD	_____	_____	PMS	2
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PMS 2, Category 0**

Audit Error Recovery Record		auderr		(1/1)	
				(Recent Change and Verify)	
1.audname:PMS	2.memno:2	3.category:0	4.threshold:5		
5.time_interval:600	6.alarm_lvl:M	7.rtn_corr:5	8.dmd_corr:1		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PMS 2, Category 3**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PMS	2.memno:2	3.category:3	4.threshold:10		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:10		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PMS 2, Category 5**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PMS	2.memno:2	3.category:5	4.threshold:5		
5.time_interval:600	6.alarm_lvl:I	7.rtn_corr:5	8.dmd_corr:5		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for PMS 1**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:PMS	2.memno:1	3.instname:_____		
4.freq:D	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

***audinst* Form for PMS 2**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:PMS	2.memno:2	3.instname:_____		
4.freq:_	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Process-Administration Audit

The Process Administration (PROAD) audit verifies the integrity of the operating system process-administration procedures (process fork, creation, and termination) and terminates incompletely created or terminated processes. This situation is determined by the length of time spent in any of these transient states. It also terminates processes that have been suspended from executing for over 30 minutes.

If the PROAD audit is run in the correcting mode, it will complete the termination of any incompletely terminated or created processes. Otherwise, it will only list information about such processes that it found.

This audit is run routinely as part of frequency-group D by SIM and should be run in the correcting mode. This is recommended since problems with process creations or terminations can tie up critical system resources. This audit should rarely report an error. Notify your technical assistance organization if the audit begins to report errors regularly.

Requesting the PROAD Audit

To request the PROAD audit, use the following command:

In PDS, **AUD:PROAD 1** [*a* | *b* | *a,b*] !
In MML, **AUD:PROAD=1**[:*a* | :*b* | :*a,b*];

Where

- a* CORR or NCG — default = CORR.
CORR specifies that the audit should correct errors.
NCG specifies that the audit does not correct errors.
- b* DETL or SUM — default = DETL.
DETL generates a raw-data report for each error found and an audit-completion message. (A limited number of raw-data reports are generated.)
SUM generates only an audit-completion message when the audit terminates. Raw-data reports are not generated for any errors that are found.

Interpreting AUD PROAD Output Messages and Error Reports

The AUD PROAD messages report the termination status of the process-administration audit and any errors found while executing the audit. For detailed explanations of the PROAD output messages, refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*.

Table 14-26 lists the PROAD audit-error numbers printed as part of the audit report. These error numbers indicate in which stage of process termination or creation the process was found.

The PROAD audit uses the following two error categories:

- Category 1 — Errors that are associated with a breakdown in process termination. This includes processes that were suspended as a precursor to termination. Recovery action when maximum is exceeded — abort audit.
- Category 2 — Errors that are associated with a breakdown in process creation or fork. Recovery action when maximum is exceeded — abort audit.

Table 14-26. PROAD Audit Errors

Category	Error Number	Meaning
1	100	DCTE use count has been decremented (scheduler).
1	110	Term DCT has been called (scheduler).
1	120	Generic Access Package (GRASP) has been informed (scheduler).
1	130	Entry has been unlinked from the dispatch chain (scheduler).
1	140	Address Translation Buffers (ATBs) have been flushed (scheduler).
1	150	Acknowledgment message has been allocated and copied (scheduler).
1	160	DCTE has been unlinked from the ready/not ready list (scheduler).
1	170	Incarnation count has been bumped (scheduler).

Table 14-26. PROAD Audit Errors (Contd)

Category	Error Number	Meaning
1	180	Message sent to the capability manager (scheduler).
1	190	Message sent to the process manager (scheduler).
1	200	Capabilities are being removed (CMGR).
1	210	Supervisor segments are being removed (CMGR).
1	220	Kernel process segments are being removed (CMGR).
1	300	Core dump started (PMGR).
1	310	Message sent to the capability manager (PMGR).
2	500	Kpstart-pcreate kernel process has begun (PMGR).
2	510	Kernel process DCTE linked to proper chain (PMGR).
2	520	Initialization event sent to kernel process (PMGR).
2	600	Pstart-pcreate supervisor process has begun (PMGR).
2	610	Scheduler awakened (SP) (PMGR).
2	620	Pstart completed (SP) (PMGR).
2	630	Execute started (SP) (PMGR).
2	700	Fork started-pfork1 (LibC).
2	710	Dupcaps message sent to FMGR (LibC).
2	720	Pfork2 started (LibC).
2	730	Sending wake-up event to the forked process (LibC).
1	800	Process has been suspended.

In summary:

- Errors 100 through 399 reflect a breakdown in process termination with:
 - Errors 100 through 199 being states in the scheduler.
 - Errors 200 through 299 being states in the capability manager.
 - Errors 300 through 399 being states in the process manager.
- Errors 500 through 699 reflect a breakdown in process creation with:
 - Errors 500 through 599 being states in kernel process creation.
 - Errors 600 through 699 being states in supervisor process creation.
- Errors 700 through 799 reflect a breakdown in the process fork.
- Error 800 indicates that the process has been suspended from executing for over 30 minutes.

Responding to AUD PROAD Error Reports

The recovery action for both categories of errors (process creation errors and process termination errors) is the same. When the error threshold is reached within the error time interval (threshold and interval specified in the *auderr* RC/V form), the following will occur. The audit will be run in demand mode by SIM. If the threshold is reached again, the audit will be inhibited by SIM.

The recommended actions to take when many errors are being reported by the PROAD audit, are to save the output and call the technical assistance organization. By saving the output, you can tell which process and the stage of creation/termination when the errors started to be reported. As long as the audit is running in the correcting mode, then the system resources being used by the incompletely terminated or created process are being returned to the system and no system degradation should occur.

If the PROAD audit was run in the noncorrecting mode, run it again in the correcting mode. If the error persists, notify your next level of technical assistance.

PROAD Recent Change Forms

This section contains illustrations of the RC/V audit forms for the PROAD audit. The audit forms are illustrated and their fields are explained in 254-303-104, *Recent Change and Verify Manual*.

⇒ NOTE:

The following forms are for general information and are typical values for *UNIX* RTR Operating System Release 6.5. Values for other releases may be different. Refer to your ECD for your specific values.

***audrec* Form for PROAD 1**

Audit Record		audrec	(1/2) (Recent Change and Verify)
1.audname:PROAD	2.memno:1	3.prname:_____	
4.timeout_int:60	5.maxseg:1	6.utility_id:6	
7.segmented:n	8.appaud:n	9.correcting:y	
10.kill_type:nonkill	11.sched_grp:kernel	12.num_er_repts:10	
13.fixdpnum:y	14.prc_num:0x6		

Modes of Execution		audrec	(2/2)
15.rtn_mode_permit:y	16.rtn_dp_type:msg	17.rtn_dp_data:0x53	
18.dmd_mode_permit:y	19.dmd_dp_type:msg	20.dmd_dp_data:0x53	
21.man_mode_permit:y	22.man_dp_type:msg	23.man_dp_data:0x53	
24.sft_mode_permit:y	25.sft_dp_type:msg	26.sft_dp_data:0x53	

***auderr* Form for PROAD 1, Category 1**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PROAD	2.memno:1	3.category:1	4.threshold:10		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***auderr* Form for PROAD 1, Category 2**

Audit Error Recovery Record		auderr		(1/1) (Recent Change and Verify)	
1.audname:PROAD	2.memno:1	3.category:2	4.threshold:10		
5.time_interval:60	6.alarm_lvl:I	7.rtn_corr:10	8.dmd_corr:50		
9.error_recovery					
10.stage	11.action	12.rdata1	13.rdata2	14.raudname	15.rmemno
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____
-	_____	_____	_____	_____	_____

***audinst* Form for PROAD 1**

Audit Instance Record		audinst		(1/1)
				(Recent Change and Verify)
1.audname:PROAD		2.memno:1		3.instname:_____
4.freq:D	5.t_sun :n	12.t_0000:n	20.t_0800:n	28.t_1600:n
	6.t_mon :n	13.t_0100:n	21.t_0900:n	29.t_1700:n
Timed	7.t_tues:n	14.t_0200:n	22.t_1000:n	30.t_1800:n
Audit	8.t_wed :n	15.t_0300:n	23.t_1100:n	31.t_1900:n
Schedule	9.t_thur:n	16.t_0400:n	24.t_1200:n	32.t_2000:n
	10.t_fri :n	17.t_0500:n	25.t_1300:n	33.t_2100:n
	11.t_sat :n	18.t_0600:n	26.t_1400:n	34.t_2200:n
		19.t_0700:n	27.t_1500:n	35.t_2300:n

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers ***UNIX*[®] RTR Operating System**

System Maintenance Manual
Volume 3

254-303-106
Issue 1
February 2002



Copyright © 1998 Lucent Technologies. All Rights Reserved.

This information product (IP) is protected by the copyright laws of the United States and other countries. It may not be reproduced, distributed, or altered in any fashion by any entity, including other Lucent Technologies Business Units or Divisions, without express written consent. For permission to reproduce or distribute, call 1-888-LTINFO6 (1-888-584-6366).

Notice

Lucent Technologies is the successor to the business and assets of AT&T Network Systems Business Unit. Every effort was made to ensure that the information contained in this IP was complete and accurate at the time of printing. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes, and you are cautioned that the hardware, software, features, and capabilities described herein may not be the same as the product you have.

Trademarks

5ESS is a registered trademark of Lucent Technologies.

AUTOPLEX is a registered trademark of Lucent Technologies.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

WE is a registered trademark of Lucent Technologies.

All other trademarks are identified as they appear in the text.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana. Orders for additional copies may be placed from the continental United States by calling 1-888-LUCENT8 (1-888-582-3688) or faxing to 1-317-322-6484. Customers outside the continental United States may place orders by contacting the in-country Lucent Technologies market manager. Lucent Technologies personnel outside the continental United States can order additional copies by faxing to 1-317-322-6699.

Information Product Support Telephone Number

Call 1-888-LTINFO6 (1-888-584-6366) to report errors or to ask questions about this IP.

Acknowledgment

Developed by Lucent Technologies Network Systems Customer Training and Information Products.

Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 3

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiphotline@lucent.com



Lucent Technologies values your comments!

Lucent Technologies
Bell Labs Innovations



3B20D and 3B21D Computers UNIX[®] RTR Operating System System Maintenance Manual Volume 3

254-303-106

Issue 1

February 2002

Lucent Technologies welcomes your comments on this information product. Your opinion is of great value and helps us to improve.

1. Was the information product:

	Yes	No	Not applicable
In the language of your choice?	☐	☐	☐
In the desired media (paper, CD-ROM, etc.)?	☐	☐	☐
Available when you needed it?	☐	☐	☐

Please provide any additional comments:

2. Please rate the effectiveness of this information product:

	Excellent	More than satisfactory	Satisfactory	Less than satisfactory	Unsatisfactory	Not applicable
Ease of use	☐	☐	☐	☐	☐	☐
Level of detail	☐	☐	☐	☐	☐	☐
Readability and clarity	☐	☐	☐	☐	☐	☐
Organization	☐	☐	☐	☐	☐	☐
Completeness	☐	☐	☐	☐	☐	☐
Technical accuracy	☐	☐	☐	☐	☐	☐
Quality of translation	☐	☐	☐	☐	☐	☐
Appearance	☐	☐	☐	☐	☐	☐

If your response to any of the above questions is "Less than satisfactory" or "Unsatisfactory," please explain your rating.

3. If you could change one thing about this information product, what would it be?

4. Please write any other comments about this information product:

Please complete the following if we may contact you for clarification or to address your concerns:

Name: _____ Date: _____

Company/organization: _____ Telephone number: _____

Address: _____

Email address: _____ Job function: _____

If you choose to complete this form online, go to <http://www.lucent-info.com/comments>
Otherwise fax to 407 767 2760 (U.S.) or +1 407 767 2760 (outside the U.S.) or email comments to ctiph hotline@lucent.com



Contents

	About This Information Product	xxxi
	■ Purpose	xxxi
	■ Audience	xxxi
	■ Reason for Issue	xxxii
	■ How to Use This Information Product	xxxiii
	■ Conventions Used	xxxvi
	■ Safety Labels	xxxviii
	■ Related Information Products	xxxix
	■ How to Comment on This Information Product	xl
	■ Mandatory Customer Information	xl

1	Hardware Identification	1-1
	■ Introduction	1-1
	■ 3B20D Model 1 Computer Hardware	1-4
	■ 3B20D Model 2 Computer Hardware	1-18
	■ 3B20D Model 3 Computer Hardware	1-36
	■ 3B21D Computer Processor Cabinet	1-52

2	Maintenance Plan	2-1
	■ Introduction	2-1
	■ Site Maintenance Plan	2-2
	■ Automatic Recovery Actions	2-3
	■ Manual Recovery Actions	2-4
	■ Preventive Maintenance	2-4
	■ Responding to Alarms	2-6
	■ Documentation	2-17
	■ Using Automatically Generated Data for Troubleshooting	2-17
	■ Switching Control Center Interface	2-18
	■ Training	2-18
	■ Field Support	2-18

Contents

3	Recovery	3-1
	■ Introduction	3-1
	■ Responding to Automatic Recovery Actions	3-1
	■ Recovering the System Manually	3-8
	■ Using Processor Recovery Messages	3-33
	■ Responding to a Terminal Suspend	3-33
	■ Isolating Emergency Action Page Problems	3-45
	■ Recovery Using Software Backup Disk and Alternate Boot Disk Options	3-51
	■ Recovering from DIOP	3-59
	■ Loading Disk from Tape	3-85

5	Routine Exercise Procedures	5-1
	■ Introduction	5-1
	■ Purpose of REX	5-1
	■ REX Operation	5-1
	■ REX Scheduling	5-2
	■ REX Scheduling Recommendations	5-2
	■ REX Manual Commands	5-3
	■ REX Inhibits	5-7
	■ REX Inhibit Recommendations	5-8
	■ REX Inhibit Implementations	5-9
	■ REX Summary	5-12

6	Diagnostics	6-1
	■ Introduction	6-1
	■ Maintenance Input Request Administrator	6-3
	■ Automatic Diagnostics	6-7
	■ Performing Diagnostics	6-8

Contents

■ Synchronous Data Link Restore Tool	6-54
■ Locating Pins	6-57
■ Processor Recovery Messages	6-59

7	Off-Line Boot	7-1
	■ Introduction	7-1
	■ Using OFLBOOT	7-3
	■ Performing OFLBOOT	7-4
	■ Checking OFLBOOT Results	7-17
	■ Examples of Performing OFLBOOT	7-20

8	Log File Review	8-1
	■ Introduction	8-1
	■ Using Log Files	8-3
	■ Core Dump Administration	8-5

9	File-System Maintenance	9-1
	■ Introduction	9-1
	■ Checking File-System Resources	9-1
	■ Manipulating Files	9-2
	■ Using File-System Audits	9-5
	■ File Corruption Detection (SIV Only)	9-9
	■ Boot File Verification Through OFLBOOT	9-10
	■ System Backup	9-11
	■ Copying to Backup Partitions	9-13
	■ Making a Backup Disk	9-26
	■ Making a Backup Tape Copy of a System Disk	9-34
	■ Initialize Disk Partition with Zeros	9-36
	■ Copying Files to Nonactive Disks	9-37
	■ Comparing and Correcting Disk Data	9-39

Contents

■ Procedure for Handling Disk Errors	9-44

10	Maintaining Software Integrity	10-1
■	Introduction	10-1
■	Checking Operating System Resources	10-1
■	Using Operating System Audits	10-4
■	Manipulating Processes	10-6
■	Maintenance Guide for ULARP	10-7
■	Using CRON	10-18
■	Writing SMD Disk Drive Defect Tables	10-19
■	Adding to the Defect Table of a SCSI Disk Drive	10-23

11	System Update	11-1
■	Introduction	11-1
■	System Update Procedures	11-1
■	SUPR Description	11-5
■	System Update Menu Page Description	11-9
■	System Update Session Examples	11-15
■	Troubleshooting System Update Procedures	11-54
■	Input Messages	11-59

Contents

12	Craft Interface	12-1
	■ Introduction	12-1
	■ Maintenance Interface	12-1
	■ Maintenance Procedures	12-4

13	Field Update	13-1
	■ Introduction	13-1
	■ Update Methods and Processes	13-1
	■ Broadcast Warning Messages	13-6
	■ Messages Files Execution and Display	13-22
	■ Field Update Session Example	13-34
	■ Input Messages	13-73
	■ System Generation Database	13-74

14	System Audits	14-1
	■ Introduction	14-1
	■ System Integrity Subsystem	14-3
	■ Procedures for Modifying Audit Records	14-38
	■ Operating-System Audits	14-50

15	Input/Output Processor Magnetic Tape Unit	15-1
	■ Introduction	15-1
	■ Using the Magnetic Tape System	15-3
	■ Operating, Maintaining, and Diagnosing the Magnetic Tape System	15-4
	■ Writing a File from a Disk	15-7

Contents

16	SC Status and SD Point Operation	16-1
	■ Introduction	16-1
	■ SCSD ECD Records	16-10
	■ Capabilities and User Requirements	16-18
	■ SCSD Commands	16-62
	■ Growth and Degrowth Procedures	16-64
	■ Verifying SCSD Operation	16-76

17	Spare Circuit Packs	17-1
	■ Introduction	17-1
	■ General Guidelines	17-1

18	Maintenance Tools	18-1
	■ Introduction	18-1
	■ Field Test Set	18-1
	■ GRASP/EGRASP	18-3
	■ Software Debugging Tools	18-4

19	SCSI Peripheral System	19-1
	■ Introduction	19-1
	■ Firmware vs. Pumpcode	19-2
	■ Verify the Integrity of Pumpcode	19-4
	■ Installing New Pumpcode	19-5
	■ SCSI Disk Drives	19-7
	■ SCSI Tape Drives	19-9

Contents

20	Magnetic Tape Special Device Filenames	20-1
	■ Introduction	20-1
	■ Tape Device <i>logdev</i> Records	20-1
	■ IOP Magnetic Tape <i>logdev</i> Records (3B20D Computer Only)	20-2
	■ SCSI/IOP Backward Compatible <i>logdev</i> Records	20-4
	■ SCSI <i>logdev</i> Records	20-5

21	Software Update	21-1
	■ Introduction	21-1
	■ Software Update Types	21-2
	■ Space Reclamation	21-12
	■ Error Conditions	21-12
	■ Enhanced SU Capabilities	21-14
	■ SU Installation	21-16
	■ SU Maintenance	21-20
	■ Routine Operations and Maintenance Procedures	21-24
	■ Page 1940 Easy SU Installation	21-44
	■ Page 1941 SU Automation Scheduling	21-49
	■ Page 1942 SU Automation Office Profile	21-53
	■ Page 1943 SU Automation Healthcheck	21-56
	■ Page 1950 SU Maintenance	21-58
	■ Page 1960 SU Installation	21-61

22	DSCH Computer Interconnect Feature	22-1
	■ Introduction	22-1
	■ Installation Procedure	22-5
	■ Maintenance Commands for DCI	22-10
	■ SG/ECD Setup	22-12

Contents

23	UUCP Feature	23-1
	■ Introduction	23-1
	■ Scope of the UUCP Feature Chapter	23-2
	■ Supported/Unsupported UUCP Commands	23-3
	■ Feature Activation/Deactivation	23-4
	■ Operation of UUCP	23-10
	■ UUCP Administration	23-18
	■ UUCP Security	23-39
	■ Troubleshooting	23-41
	■ Error Code Interpretation	23-45
	■ UUCP Administrative Files	23-63
	■ UUCP Manual Pages	23-65

A	3B20D Computer References	A-1
	■ Introduction	A-1
	■ Control Unit Diagnostic Test Sequence	A-2
	■ Diagnostic Hierarchy	A-4
	■ Test Procedures	A-6
	■ Supplementary Phases	A-11
	■ Main Store Failure Data	A-23
	■ Processor Recovery Messages	A-29
	■ Diagnostic Information Tables — CU, DFC, and IOP	A-30

B	3B21D Computer References	B-1
	■ Introduction	B-1
	■ Control Unit Diagnostic Test Sequence	B-2
	■ Diagnostic Hierarchy	B-4
	■ Test Procedures	B-6
	■ Supplementary Phases	B-13
	■ Main Store Failure Data	B-26

Contents

- Processor Recovery Messages B-29
- Diagnostic Information Tables — CU, DFC,
IOP, and DCI B-30

Glossary

GL-1

Index

IN-1

Figures

1

Hardware Identification

1-1.	Typical 3B20D Model 1 Computer Functional Block Diagram	1-5
1-2.	3B20D Model 1 Computer Control Unit Frame	1-7
1-3.	3B20D Model 1 Computer Peripheral Control Frame	1-10
1-4.	3B20D Model 1 Computer 300-MB Moving Head Disk Frame	1-13
1-5.	3B20D Model 1 Computer SCSI Disk Cabinet	1-15
1-6.	3B20D Model 1 Computer Power Distribution Frame	1-17
1-7.	Typical 3B20D Model 2 Computer Functional Block Diagram (SMM Release 1)	1-19
1-8.	Typical 3B20D Model 2 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-20
1-9.	Typical 3B20D Model 2 Computer Functional Block Diagram (VLMM Release 6 Retrofit)	1-21
1-10.	3B20D Model 2 Computer Processor Control Frame	1-22
1-11.	3B20D Model 2 Computer 300-MB Moving Head Disk Frame	1-28
1-12.	3B20D Model 2 Computer Tape/Disk Cabinet	1-30
1-13.	3B20D Model 2 Computer SCSI Disk Cabinet	1-32
1-14.	3B20D Model 2 Computer 160-MB Minimodule Disk Frame	1-34
1-15.	3B20D Model 2 Computer Power Distribution Frame	1-35
1-16.	Typical 3B20D Model 3 Computer Functional Block Diagram (SMM Release 1)	1-37

Figures

1-17.	Typical 3B20D Model 3 Computer Functional Block Diagram (EMM Release 1 Retrofit)	1-38
1-18.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SMD DFC Unit	1-39
1-19.	Typical 3B20D Model 3 Computer Functional Block Diagram (VLMM Release 6) with SCSI DFC Unit	1-40
1-20.	3B20D Model 3 Computer Processor Cabinet	1-41
1-21.	3B20D Model 3 Computer Tape/Disk Cabinet	1-48
1-22.	3B20D Model 3 Computer SCSI Disk Cabinet	1-50
1-23.	3B20D Model 3 Computer Power Cabinet	1-51
1-24.	3B21D Computer Processor Cabinet	1-53
1-25.	3B21D Computer Growth Cabinet	1-56

2

Maintenance Plan

2-1.	3B20D Computer TN2, TN3, TN5, and TN6 Power Switches Faceplate	2-10
2-2.	3B21D Computer UN373 (CU Power Switch) Faceplate	2-12
2-3.	3B21D Computer UN375 (CU Power Switch) Faceplate	2-13
2-4.	3B21D Computer UN376 (CU Power Switch) Faceplate	2-14

Figures

3

Recovery

3-1.	Recovery Procedure (System Configuration)	3-30
3-2.	Recovery Procedure (Application Failure)	3-31
3-3.	Recovery Procedure (Post Recovery)	3-32
3-4.	MTTY/3B20D Computer Interface (Model 1)	3-47
3-5.	MTTY/3B20D Computer Interface (Model 2 or Model 3)	3-48
3-6.	MTTY/3B21D Computer Interface with a TN983 MTTYC	3-49
3-7.	MTTY/3B21D Computer Interface with a UN583 MTTYC	3-50
3-8.	Poke 120 — DFC Status Page	3-62

Figures

6

Diagnostics

6-1.	Sample Output from OP:DMQ	6-20
6-2.	Sample Diagnostic Failure	6-26
6-3.	Sample TLP Circuit Pack List	6-30
6-4.	Sample Diagnostic Listing	6-39
6-5.	Sample Output from CU DGN Audit	6-45
6-6.	Sample Output from Non-DGN CU Audit	6-51
6-7.	Backplane Pin Set	6-58

10

Maintaining Software Integrity

10-1.	ULARP ECD Record	10-7
10-2.	Control/Display Update Page	10-17

11

System Update

11-1.	System Update Procedure	11-2
11-2.	System Update Menu Page Screen	11-11
11-3.	Enter Arguments Page 1 Screen	11-13
11-4.	Enter Arguments Page 2 Screen	11-14
11-5.	System Update Menu Page	11-16

12

Craft Interface

12-1.	Maintenance Interface Configuration	12-3
12-2.	MTTY Terminal Critical Indicator Display	12-17
12-3.	Emergency Action Page for TN83	12-20
12-4.	Emergency Action Page for TN983	12-21

Figures

12-5.	Emergency Action Page for UN583 or UN597 without Alternate Boot Disk Feature (pre R21.35)	12-22
12-6.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (pre R21.35)	12-23
12-7.	Emergency Action Page for UN597 without Alternate Boot Disk Feature (R21.35 and later)	12-24
12-8.	Emergency Action Page for UN597 with Alternate Boot Disk Feature (R21.35 and later)	12-25
12-9.	Lock Screen Display for UN597	12-26
12-10.	Page Index for RTR Release 21.3 and Earlier	12-37
12-11.	Page Index for RTR Release 21.5 Through Release 21.15	12-38
12-12.	Page Index for RTR Release 21.17 and Later	12-39
12-13.	Status Summary Area Page	12-43
12-14.	Common Processor Display Page	12-45
12-15.	Common Processor Display Page in DIOP	12-46
12-16.	Common Processor Display Page — MTTY and ROP States	12-47
12-17.	Common Processor Display Page — On-Line	12-48
12-18.	Common Processor Display Page — Off-Line	12-49
12-19.	3B20D Computer Common Processor Display Page with CU FAN Alarms	12-50
12-20.	3B21D Computer Common Processor Display Page with CU FAN Alarm	12-51
12-21.	3B21D Computer Common Processor Display Page with ABD Feature	12-53
12-22.	Control/Display Update Page	12-58
12-23.	Operating System Status Page	12-60
12-24.	CFTMTC Display Page 105	12-65
12-25.	CFTMTC Display Page 105 with Menu 912 Data	12-66

Figures

12-26.	CFTMTC Display Page 106	12-67
12-27.	System Update Menu Page	12-74
12-28.	DCI Status Page	12-76
12-29.	DCI Status Page for RCC	12-77
12-30.	DCI Status Page with DSMLNK	12-78
12-31.	Field Update Page	12-83
12-32.	Typical DFC Index Page	12-89
12-33.	Typical DFC Status Page for a System Equipped with Storage Module Drive (SMD) DFCs 0 and 1	12-91
12-34.	Typical DFC Status Page for a System Equipped with SMD DFCs 2 and 3	12-92
12-35.	Typical DFC Status Page for a System Equipped with SCSI DFCs 0 and 1	12-93
12-36.	Typical DFC Status Page for a System Equipped with SCSI DFCs 2 and 3	12-94
12-37.	Typical DFC Status Page for a System, in Full DIOP, Equipped with SCSI DFCs 0 and 1	12-95
12-38.	Typical DFC Performance Page for a System Equipped with SMD DFCs 0 and 1	12-102
12-39.	Typical DFC Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-103
12-40.	Typical DFSA Index Page	12-106
12-41.	Typical DFSA Status Page for a System Equipped with SMD DFCs 0 and 1	12-108
12-42.	Typical DFSA Status Page for a System Equipped with SMD DFCs 2 and 3	12-109
12-43.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 0 and 1	12-110
12-44.	Typical DFSA Status Page for a System Equipped with SCSI DFCs 2 and 3	12-111

Figures

12-45.	Typical DFSA Status Page for a System, in FULL DIOP, Equipped with SCSI DFCs 0 and 1	12-112
12-46.	Typical DFSA Performance Page for a System Equipped with SMD DFCs 0 and 1	12-118
12-47.	Typical DFSA Performance Page for a System Equipped with SCSI DFCs 2 and 3	12-119

Figures

13

Field Update

13-1.	BWM File Structure	13-8
13-2.	BWM SCANS File	13-10
13-3.	BWM HEADER File	13-11
13-4.	BWM MSGS File (MML Format)	13-14
13-5.	Update Package States	13-18
13-6.	Field Update Page	13-23

14

System Audits

14-1.	Audit-Control Subsystem	14-4

16

SC Status and SD Point Operation

16-1.	Software Structure	16-3
16-2.	Three Levels of Software Interfacing SCSD Hardware	16-4
16-3.	Sample scsdbody Form	16-10
16-4.	3B20D Computer SCSD Point Assignments to Logical Groups Examples	16-12
16-5.	3B21D Computer SCSD Point Assignments to Logical Groups Examples	16-13
16-6.	Structure of the SCSD Option Block in the ECD	16-15
16-7.	UN33 Block Diagram	16-21
16-8.	UN933 Block Diagram	16-22
16-9.	Administrator Message Structure	16-32
16-10.	Error Codes for Administrator and Client Processes	16-34
16-11.	ECD Option Block Record	16-35
16-12.	Power Switch	16-40

Figures

16-13.	3B20D Computer Scan Point Transition Diagram	16-43
16-14.	PSM Data Structure	16-49

17 Spare Circuit Packs

17-1.	Circuit Pack Inventory and Testing Log	17-2

19 SCSI Peripheral System

19-1.	Tape Path Components	19-15
19-2.	Cleaning Points	19-19
19-3.	UN376 (NCR006-3503341), UN376C (NCR006-3300608/Comcode 407545243), and UN376E (KS-24367, L1/Comcode 407771260) SCSI DAT Drives — Front View	19-20
19-4.	UN376 and UN376B Series 1 and 2 Circuit Pack Layout	19-25
19-5.	UN376B Series 3 and Later Circuit Pack Layout	19-26
19-6.	Cassette Write Protect	19-27
19-7.	Tape Load Diagram	19-35

20 Magnetic Tape Special Device Filenames

20-1.	IOP Tape Device Partition Number Descriptor	20-2
20-2.	SCSI/IOP Tape Device Partition Number Descriptor	20-4
20-3.	SCSI Tape Device Partition Number Descriptor	20-6

Figures

21 Software Update

21-1.	Software Update Distribution	21-3
21-2.	Software Update Structure	21-5
21-3.	Example of a PDS Message File	21-6
21-4.	Software Update Page 109	21-15
21-5.	Page 1960 SU Installation Menu	21-17
21-6.	Typical 3B20D/3B21D Computer On-Line SU History	21-20
21-7.	Page 1940 Easy SU Installation Menu	21-45
21-8.	Page 1941 SU Automation Scheduling Menu	21-50
21-9.	Page 1942 SU Automation Office Profile Menu	21-53
21-10.	Page 1943 SU Automation Healthcheck Menu	21-57
21-11.	Page 1950 SU Maintenance Menu	21-58
21-12.	Page 1960 SU Installation Menu	21-62

22 DSCH Computer Interconnect Feature

22-1.	Software Architecture of the DCI Feature	22-3
22-2.	Hardware Architecture of the DCI Feature	22-4
22-3.	Example of DCI ucb Form	22-18
22-4.	Example of HA-OMP DCI ucb Form	22-20
22-5.	Example of HA-OMP Adjunct ucb Form	22-22
22-6.	Example of Normal Operations mdct Form	22-24
22-7.	Example of Diagnostic mdct Form	22-25
22-8.	Example of HA-OMP Adjunct mdct Form	22-26

Figures

22-9.	Example of Diagnostic <i>logdev</i> Form	22-27
22-10.	Example of Non-HA-OMP <i>logdev</i> Form	22-27
22-11.	Example of HA-OMP <i>logdev</i> Form	22-27
22-12.	Example of <i>chopt</i> Form	22-28
22-13.	Example of <i>dciopt</i> Form	22-28
22-14.	Example of <i>device</i> Form	22-29
22-15.	Example of <i>classdef</i> Form	22-30
22-16.	Example of <i>errthld</i> Form	22-31

A 3B20D Computer References

A-1.	BOT and EOT Marker Placement	A-22
A-2.	Sample Memory Failure Histogram	A-25

B 3B21D Computer References

B-1.	Sample Memory Failure Histogram	B-27



Tables



About This Information Product

1. IPs Supporting 3B21D Computer	xxxix



1 Hardware Identification

1-1. 3B20D Model 2 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-25
1-2. 3B20D Model 3 Computer Circuit Packs Used for SMM R1, EMM R1 Retrofit, and VLMM R6 Retrofit	1-44



3 Recovery

3-1. Initialization Levels	3-2

Tables

5

Routine Exercise Procedures

5-1.	Early REX Inhibits	5-10
5-2.	New REX Inhibits	5-11

6

Diagnostics

6-1.	Diagnostic Hierarchy Appendix Index	6-8
6-2.	Suggested Routine Test Procedures	6-11
6-3.	EX Commands	6-13
6-4.	Diagnostic Results	6-27
6-5.	Explanation of Test Skipped Reasons	6-28
6-6.	Meaning of TLP Notes	6-31
6-7.	DIAMON Error Codes	6-43
6-8.	DGN CU Error Codes	6-46
6-9.	Selected System Error Codes	6-48
6-10.	Maintenance Channel Status	6-50
6-11.	DGN Audit IOP, DFC, and DCI Error Codes	6-52

7

Off-Line Boot

7-1.	Options for the EXC:OFLBOOT Input Message	7-6
7-2.	Precheck Override Conditions	7-9
7-3.	Secondary DFC Override Condition	7-9

Tables

9

File-System Maintenance

9-1.	File-System Maintenance Commands (Routine)	9-3
9-2.	File-System Maintenance Commands (Troubleshooting)	9-3
9-3.	File-System Maintenance Commands (Permanent)	9-4

10

Maintaining Software Integrity

10-1.	Operating System Audit Commands	10-4
10-2.	Process Execution and Termination Commands	10-6
10-3.	PRM Function Codes	10-14

12

Craft Interface

12-1.	KS-22921, L1 Maintenance Terminal Configuration	12-5
12-2.	KS-23554, L5 and L6 Maintenance Terminal Configuration	12-6
12-3.	KS-23996, L1 and L5 Maintenance Terminal Configuration	12-8
12-4.	Model 5310 ROP Option Selection	12-12
12-5.	Model 577 ROP Option Selection	12-13
12-6.	Model 602 ROP Option Selection	12-15
12-7.	System Critical Indicators	12-18
12-8.	Emergency Action Page Commands	12-27
12-9.	Emergency Action Page Display Indicators	12-32
12-10.	Emergency Action Page Poke Prompts	12-34

Tables

12-11.	Common Processor Display Page Commands Without ABD Feature	12-52
12-12.	Common Processor Display Page Commands with ABD Feature	12-54
12-13.	Parameters and Options for CFTMTC Commands	12-68
12-14.	Display Page 105 Commands, Parameters, and Options for Menu Numbers	12-71
12-15.	Display Page 106 Commands, Parameters, and Options for Menu Numbers	12-72
12-16.	DCI Status Page Commands	12-79
12-17.	Field Update Scope/Action Combinations	12-85
12-18.	DFC Status Page Commands	12-98
12-19.	DFC Performance Page Menu Commands	12-104
12-20.	Disk File System Access Status Page Menu Commands	12-115
12-21.	DFSA Performance Page Menu Commands	12-120

13

Field Update

13-1.	Allowed Scope/Action Combinations	13-27

14

System Audits

14-1.	Audit Families and Audit Names	14-6
14-2.	Audit Names and Member Numbers	14-8
14-3.	Audit Abort Codes	14-23
14-4.	REPT SIMCHK Error Codes	14-27
14-5.	Optional Hexadecimal Data for REPT SIMCHK Error Conditions	14-31

Tables

14-6.	CUMEM Audit Error Numbers	14-53
14-7.	CUMEM Error Reports	14-53
14-8.	CUSTAT Audit Error Numbers	14-60
14-9.	CUSTAT Error Reports	14-61
14-10.	AUD CUSTAT Error Messages	14-67
14-11.	ECD Audit Error Messages	14-76
14-12.	LLA Audit Error Messages	14-79
14-13.	ECDOWN Audit Error Messages	14-97
14-14.	FMGR Audit Errors	14-104
14-15.	FMGR Audit Error Reports	14-107
14-16.	FSBLK Audit Errors	14-136
14-17.	Special Error Messages	14-137
14-18.	FSLINK Audit Errors	14-147
14-19.	Special Error Messages	14-148
14-20.	FSCMPT Audit Errors	14-161
14-21.	Special Error Messages	14-162
14-22.	MMGR Audit Error Messages	14-171
14-23.	MSGBUF Audit Error Messages	14-193
14-24.	Special Error Messages	14-195
14-25.	PMS Audit Error Messages	14-208
14-26.	PROAD Audit Errors	14-219

15 Input/Output Processor Magnetic Tape Unit

15-1.	Magnetic Tape System Input Commands	15-3
15-2.	MTC Diagnostic Commands	15-6

Tables

16 SC Status and SD Point Operation

16-1.	Phase Alarm Levels	16-31
16-2.	3B20D Computer Logical Group Assignments Examples	16-37
16-3.	3B21D Computer Logical Group Assignments Examples	16-38
16-4.	3B20D Computer Scan Point Transitions and Actions	16-42
16-5.	3B21D Computer Scan Point Transitions and Actions	16-44
16-6.	3B20D Computer Scan Point and Signal Distributor Point Assignments	16-53
16-7.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Large Office	16-54
16-8.	3B21D Computer Scan Point and Signal Distributor Point Assignments — Small Office	16-55

19 SCSI Peripheral System

19-1.	Cleaning Schedules and Equipment	19-13
19-2.	Cleaning Schedule Guidelines	19-16
19-3.	Description of Configuration Option Byte 15 Bit Settings	19-23
19-4.	DAT SCSI Device Identification Switch Settings	19-26
19-5.	KS-23909 9-Track Tape Capacity	19-28
19-6.	DAT Cartridge Tape Capacity	19-29
19-7.	NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT Drive Indicators	19-40
19-8.	NCR006-3503341, L1 3.5-Inch SCSI DAT Drive Indicators	19-41

Tables

20 Magnetic Tape Special Device Filenames

20-1.	IOP 9-Track Operating Parameters	20-2
20-2.	Typical <i>logdev</i> Values	20-7

21 Software Update

21-1.	File Transfer Error Conditions (SCANS/CSCANS Interface)	21-13
21-2.	Predefined SU Automation Healthcheck Conditions	21-56

23 UUCP Feature

23-1.	ED3T076-20 G57 and G58 Signal Connections	23-11
23-2.	UUCP Troubleshooting Corrective Action	23-41
23-3.	Error Codes and Definitions	23-50
23-4.	uucheck Error Messages	23-54
23-5.	uucheck Warning Messages	23-57
23-6.	uucheck Caution Messages	23-60
23-7.	uucheck Internal Error Messages	23-62

A 3B20D Computer References

A-1.	Recommended Frequency of Selected Supplementary Phases	A-11
A-2.	Supplementary Phases for CU	A-12
A-3.	Supplementary Phases for SMD DFC	A-18
A-4.	Supplementary Phases for SCSI DFC	A-19
A-5.	Supplementary Phases for IOP	A-20

Tables

A-6.	Partitioning of Address Spectrum Among Memory Arrays	A-27
A-7.	Application Channel Interface	A-31
A-8.	Central Control	A-32
A-9.	Cache Storage Unit	A-36
A-10.	Enlarged Cache Storage Unit	A-37
A-11.	Disk File Controller	A-38
A-12.	SCSI DFC	A-39
A-13.	Direct Memory Access	A-40
A-14.	Direct Memory Access Channel	A-41
A-15.	Dual Serial Channel	A-43
A-16.	Direct User Interface Controller	A-44
A-17.	Input/Output Processor	A-45
A-18.	Main Store Controller	A-46
A-19.	Default Refresh Data Parity Check Times for MASC Phase 96	A-47
A-20.	Moving Head Disk	A-48
A-21.	Magnetic Tape Controller	A-49
A-22.	Maintenance Teletypewriter Controller	A-49
A-23.	Store Address Translator	A-50
A-24.	Serial Channel	A-50
A-25.	Scan Point and Signal Distributor Controller	A-51
A-26.	Synchronous Data Link Controller	A-51
A-27.	Teletypewriter Controller	A-51
A-28.	Utility Circuit	A-52

B 3B21D Computer References

B-1.	Recommended Frequency of Selected Supplementary Phases	B-14
B-2.	Supplementary Phases for CU	B-15
B-3.	Supplementary Phases for DFC 2	B-21
B-4.	Supplementary Phases for DFC 3	B-22
B-5.	Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape	B-23

Tables

B-6.	Supplementary Phases for IOP 2	B-24
B-7.	Central Control	B-31
B-8.	Main Store Controller	B-34
B-9.	Direct Memory Access	B-36
B-10.	Direct Memory Access Channel	B-37
B-11.	Disk File Controller 2	B-38
B-12.	Disk File Controller 3	B-39
B-13.	SCSI Magnetic Tape	B-40
B-14.	Utility Circuit	B-41
B-15.	Input/Output Processor 2	B-42
B-16.	Dual Serial Channel Computer Interconnect	B-43

About This Information Product

Purpose

The System Maintenance Manual is an information product (IP) that is part of a documentation set used to support the *UNIX*® Real-Time Reliable (RTR) operating system running on Lucent Technologies 3B20D and 3B21D computers. It summarizes the corrective and preventive maintenance procedures used to operate and maintain the 3B20D Model 1, 2, and 3 computers and the 3B21D computer. The procedures range from recovering the system to analyzing the error conditions and performing diagnostics. Software diagnostic capabilities are also discussed.

Other topics are discussed that may not be considered a part of routine maintenance, such as the set up and operation of the maintenance terminal, performing updates, running audits, and the use of peripheral equipment.

Audience

This IP provides information for a variety of audiences. Anyone interested in the *UNIX* RTR operating system, 3B20D/3B21D computers, or a summary of corrective and preventive maintenance will find it useful. The primary audience for this IP includes the following:

- Hardware, software, or training course developers
- Craft or maintenance personnel
- 3B20D/3B21D computer system administrators or documentation users.

Reason for Issue

This IP is being issued to incorporate new information for 3B20D/3B21D computers and to bring the IP into compliance with new Lucent Technologies documentation standards. The major changes are as follows:

- Change the six-digit Select Code number for the *System Maintenance Manual* from 304-046 to the new nine-digit number 254-303-106.
- Add information on alarms and responding to alarms in Chapter 2, and remove references to the UN375C disk pack in Figure 2-3.
- Add circuit pack information for UN582 and UN583 in Chapters 3 and 12.
- Add SOURCE field p and h postmortem descriptions in Chapter 4.
- Add DFC to Tables 6-1, 6-2, and 6-5.
- Add TLP Note 04 for CC, CSU, DMA, and MASC in Table 6-6.
- Add new error codes and meaning to Table 6-11.
- Revise the “Copying to Backup Partitions” procedure in Chapter 9.
- Add new figures associated with the poke commands in Chapter 12.
- Add circuit pack information for UN933 in Chapter 16.
- Add circuit pack information for UN376E in Chapter 19.
- Update Appendixes A and B.

Change bars on the far right-hand side of the text are used to denote changes made to the current issue. Change bars will help you understand at a glance what has changed from the previous issue. The position of deleted text is shown by a single asterisk (*). Not all editorial changes will be marked with change bars. Occasionally, change bars do not work in tables or figures; therefore, those changes will be shown by adding a change bar to only the table or figure title.

Whenever this IP is reissued, the reason(s) for reissue will be listed in this paragraph.

How to Use This Information Product

This IP is composed of independent chapters and reference information (Table of Contents, Appendixes, Glossary, and Index). Specific topics are referenced in the Table of Contents and Index with page number locations. The Glossary defines the terminology used in this IP. This IP is organized in three volumes, as follows:

Volume 1

- Master Table of Contents
- About This Information Product
Provides general information about the IP. Information regarding audience, reason for issue, conventions, safety labels, how to order, and how to comment is included. A description of how the IP is organized and a brief description of the contents of each chapter is also provided.
- Chapter 1 — Hardware Identification
Describes the hardware in the computers and contains illustrations that show the location of specific units, circuit packs, components, and associated frames.
- Chapter 2 — Maintenance Plan
Describes the elements in the plan for maintaining the computers.
- Chapter 3 — Recovery
Provides the following recovery procedures: responding to automatic recovery actions, recovering the system manually, using processor recovery messages, responding to a terminal suspend, isolating emergency action page problems, recovering from a duplex outage, recovering from Disk Independent Operation (DIOP), and loading disk from tape.
- Chapter 4 — Error Analysis
Gives an overview of the Control Unit (CU) error messages and automatic postmortem dumps. It will help the user interpret the different types of error messages generated by the automatic fault recovery features.
- Chapter 5 — Routine Exercise Procedures
Provides a description of the Routine Exercise (REX) processes and procedures. Included are detailed descriptions and examples of the REX craft interface and recommendations for scheduling and inhibiting REX.
- Chapter 6 — Diagnostics
Gives general information for using the diagnostic capabilities of the computers, how and when to perform diagnostics, how to interpret the output, and how to use the information to clear hardware problems.

- **Chapter 7 — Off-Line Boot**
Covers the Off-Line Boot (OFLBOOT) feature for the *UNIX* RTR Operating System Release 21.1 and later. OFLBOOT is used primarily to verify the capability to boot essential files without interrupting the system operation.
- **Chapter 8 — Log File Review**
Gives general information about the history of certain automatic and manual maintenance actions that occur in the system. These histories are located in the */etc/log* directory.
- **Chapter 9 — File-System Maintenance**
Gives procedures for checking file-system resources, manipulating files, using the file-system audits, file-system corruption detection, and backing up and restoring file systems.
- **Chapter 10 — Maintaining Software Integrity**
Gives procedures for checking operating system resources by using operating system audits, the User-Level Automatic Restart Process (ULARP), the Clock Daemon (CRON), and manipulating processes.

Volume 2

- **Master Table of Contents**
- **About This Information Product**
- **Chapter 11 — System Update**
Describes system update concepts and defines procedures for implementing updates. This chapter also provides examples of successful system updates and how to back out and restore an unsuccessful update. It describes other processes and utilities related to performing system updates and lists the system update input messages with a short description of their function.
- **Chapter 12 — Craft Interface**
Provides an overview of the software required to perform the various craft interface functions. The information presented summarizes operation of the Maintenance Teletypewriter (MTTY) terminal.
- **Chapter 13 — Field Update**
Describes field update concepts and defines procedures for implementing updates. This chapter also describes Broadcast Warning Messages (BWMs) and gives some rules for applying BWMs. It also describes BWM components and states.

- Chapter 14 — System Audits
Describes the system audits available to the user and the software integrity subsystem (the system feature that monitors the operating system software reliability and stability). This chapter also provides procedural information dealing with system audits, audit output messages and error reports, and system faults.

Volume 3

- Master Table of Contents
- About This Information Product
- Chapter 15 — Input/Output Processor Magnetic Tape Unit
Describes the magnetic tape system components and gives guidelines for its operation, use, and maintenance.
- Chapter 16 — SC Status and SD Point Operation
Provides information on how to write client processes, communicate with the Scan Point and Signal Distributor Handler (SCSDH), physically connect scan points, develop messages for maintenance personnel, grow and degrow UN33/UN933 boards, and grow and degrow the Input/Output Processor (IOP).
- Chapter 17 — Spare Circuit Packs
Gives guidelines for testing and rotating spare circuit packs.
- Chapter 18 — Maintenance Tools
Describes the available maintenance tools for the 3B20D/3B21D computers: the Field Test Set (FTS) and the Generic Access Package (GRASP)/Enhanced Generic Access Package (EGRASP).
- Chapter 19 — SCSI Peripheral System
Provides information which is helpful in maintaining the Small Computer System Interface (SCSI) peripheral system.
- Chapter 20 — Magnetic Tape Special Device Filenames
Describes the magnetic tape special device filenames.
- Chapter 21 — Software Update
Provides a description of program changes that are possible within the database.
- Chapter 22 — DSCH Computer Interconnect Feature
Provides information on the Dual Serial Channel (DSCH) Computer Interconnect (DCI) feature. The feature is implemented in the RTR Release 21.3 and later software. This chapter presents an overview of the DCI feature, configuring the Equipment Configuration Database (ECD), installation procedures for the *Sun** system, and maintenance commands for the DCI board.

* Sun, Sun Microsystems, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

- Appendix A — 3B20D Computer References
Provides 3B20D computer reference information on the following topics:
CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures;
Supplementary Phases; Main Store Failure Data; Processor Recovery
Messages; and the CU, Disk File Controller (DFC), and IOP Diagnostic
Information Tables.
- Appendix B — 3B21D Computer References
Provides 3B21D computer reference information on the following topics:
CU Diagnostic Test Sequence; Diagnostic Hierarchy; Test Procedures;
Supplementary Phases; Main Store Failure Data; Processor Recovery
Messages; and the CU, DFC, IOP, and DCI Diagnostic Information
Tables.
- Glossary
- Index.

Conventions Used

Command, Filename, and Display Notations

The following notations are used to show commands and filenames in the text and displays.

NOTE:

System filenames and command names are case-sensitive, so you must enter them exactly as they are shown.

- Command names in text appear in **bold** type; for example, the **/usr/bin/ls** command. Command names in headings appear in **bold** type.
- Filenames and form names in text appear in *italic* type; for example, the */usr/lib/uucp/System* file or the *scsdbody* form. Filenames and form names in headings appear in ***bold-italic*** type.
- Text that you enter, such as a command or response to a prompt, appears in **bold** type; for example, the **ls -la** command. The convention of making text that you enter **bold** overrides other conventions; for example, enter **scsdbody** for the form name.
- Variables that appear in a command line or file appear in *italic* type; for example, **grep *username* /etc/passwd**. In this example, *username* is a variable indicating a user's name is required.

- Screen displays and system messages appear in constant width type; for example, Please enter your password. Program code listings and file listings are also shown in constant width type. Input messages are shown in **constant-width bold** type.
- Comments and explanations within a display are indented and shown in *italic* type. These are for information only and will not appear on your screen.
- A line in a file or on the computer screen that is too long to be shown as it actually appears in this IP will be shown with a backslash (\) at the end of the first line. This indicates the next line should be read as a continuation of the current line.
- Square brackets around an argument on a command line indicate that the argument is optional; for example, the **lpstat [-t]** command. In this example, the **-t** argument is optional and can be omitted.
- A vertical bar (|) between words in an argument on a command line indicates that one of the arguments is to be selected.
- The key identified on your keyboard as Return, Enter, or a bent arrow (↵) is referred to as the Return key. Occasionally, representations of this Return key will be boxed; for example, Return.
- There is an implied Return at the end of each command and menu response that you enter. Some examples do not explicitly show the Return. Where you may be expected to enter a Return (as in the case where you are accepting a menu default), the symbol <CR> is shown to indicate that you are to press the Return key.
- Key combinations appear in a hyphenated format; for example, Ctrl-d. Press and hold down the first key of a key combination while pressing the second key.
- Ellipses (three dots) on a command line indicate that the previous argument can be repeated; for example, **ls [file ...]**. In this example, multiple files can be listed after the command.
- References to manual pages are followed by their manual page location number in parentheses; for example, **mount(1M)**.

Hexadecimal Notation

Hexadecimal (base 16) numbers are denoted with a **0x** prefix; for example, 0x00A is decimal 10.

Signal Designations

The term “asserted” is used in the descriptions to mean that a signal is driven to its active state. The term “negated” is used in the descriptions to mean that a signal is driven to its inactive state.

Signal names used in the diagrams and descriptions that end in a 0 or 1 indicate the active state of the signal. Names ending in “0” are “active low” signals; names ending in “1” are “active high” signals. For example, CCIOD(31-00)1 describes the Central Control Input/Output Data bits 31 through 00 which are “active high” signals.

Equipment Locations

A coordinate numbering system is used to identify Equipment Locations (EQLs) in units and cabinets. The origin is the lower left front of the cabinet or unit. Vertical increments are measured in inches. Horizontal increments are measured in eighths of an inch. The coordinate location of a circuit pack is expressed as the horizontal and vertical location of the center lines of the connector into which the circuit pack is inserted. The location of a unit in a cabinet is identified by the placement of the lower left corner of the unit in the cabinet.

For example, a connector at EQL 004-080 is located 4 inches above the origin and 10 inches ($80 \times 0.125 = 10.0$) to the right of the origin.

Safety Labels

Safety labels are reminders used to assure the safety of personnel and to minimize service interruptions, loss of data, and damage to equipment, products, and software.

Three types of safety labels are used in Lucent Technologies documentation. The three types, in descending order of priority, are as follows:

1. **DANGER** indicates the presence of a hazard that **will** cause death or severe personal injury if the hazard is not avoided.
2. **WARNING** indicates the presence of a hazard that **can** cause death or severe personal injury if the hazard is not avoided.
3. **CAUTION** indicates the presence of a hazard that **will** or **can** cause minor personal injury or property damage if the hazard is not avoided.

This IP contains safety labels in the form of CAUTION statements.

Related Information Products

Table 1 lists by IP number and Select Code the Lucent Technologies IPs supporting the 3B21D computer. IPs with Select Codes will be converted to nine-digit IP numbers as they are reissued.

Table 1. IPs Supporting 3B21D Computer

NEW IP NUMBER	OLD SELECT CODE	TITLE
254-303-100	-	3B21D Computer Growth/Retrofit Tasks
254-303-101	-	3B21D Computer Routine Maintenance Tasks
254-303-102	-	3B21D Computer Trouble Clearing Tasks
254-303-103	303-007	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Processor Recovery Messages Guide
254-303-104	303-010	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Recent Change and Verify Manual
254-303-105	304-045	3B21D Computer Hardware Reference Manual
254-303-106	304-046	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System System Maintenance Manual
254-303-107	303-072	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System Software Troubleshooting Guide
254-303-110	303-080	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Input Messages Manual
254-303-111	303-081	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System PDS Output Messages Manual
254-303-112	303-082	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Input Messages Manual
254-303-113	303-083	3B20D and 3B21D Computers <i>UNIX</i> RTR Operating System MML Output Messages Manual

IP references will not include full titles; for example, "Refer to 254-303-105, *Hardware Reference Manual*, for more information."

How to Comment on This Information Product

Lucent Technologies welcomes your comments on this IP. Your comments will aid us in improving the quality and usefulness of Lucent Technologies documentation. Please use the Feedback Form provided at the front of this IP. If the Feedback Form is missing, mail your comments to the following address:

Lucent Technologies
Customer Training and Information Products (CTIP)
2400 Reynolda Road
Winston-Salem, NC 27106-4606

Mandatory Customer Information

The 3B21D computer is used as the Administrative Module (AM) in various switching system applications and is not provided as a stand-alone product. Therefore, the application documentation is responsible for providing this information.

Refer to the applicable application documentation for "Mandatory Customer Information."

Input/Output Processor Magnetic Tape Unit

15

Contents

Introduction	15-1
Using the Magnetic Tape System	15-3
Operating, Maintaining, and Diagnosing the Magnetic Tape System	15-4
■ Guidelines for Tape Transport Operation	15-4
■ Guidelines for Preventive Maintenance	15-5
■ Guidelines for MTC Diagnostics	15-5
Writing a File from a Disk	15-7

Introduction

The Input/Output Processor (IOP) 9-track Magnetic Tape (MT) system is part of the 3B20D computer and is used to aid the operators in maintaining the overall system. For Small Computer System Interface (SCSI) tape drives, refer to Chapter 19, “SCSI Peripheral System.” The tape system is used to:

- Read field updates/Broadcast Warning Messages (BWMs) to the disk [before the availability of the Software Change Administration and Notification System (SCANS) II.]
- Write files from the disk, such as emergency dumps, log files, off-line dumps, and so forth. (Refer to the “Writing a File from a Disk” section.)
- Provide system backup, Physical Disk-to-Tape (PDT) writer. (Refer to the “Making a Backup Tape Copy of a System Disk” section in Chapter 9.)
- Write the system disk from tape [Load Disk from Tape (LDFT)].
- Release update.

The magnetic tape systems for 3B20D Model 1, 2, and 3 computers contain the following components:

- 3B20D Model 1 Computer
 - Magnetic Tape Controller (MTC) (UN52—1600 bpi or UN134—1600 bpi)
 - Formatter
 - Magnetic tape device.

- 3B20D Model 2 Computer
 - Magnetic tape controller (UN52—1600 bpi)
 - Magnetic tape reader/recorder with integrated format controller (KS-22762).
- 3B20D Model 3 Computer
 - Magnetic tape controller (UN52—1600 bpi or UN145—1600 or 6250 bpi)
 - Magnetic tape reader/recorder with integrated format controller (KS-22762—1600 bpi or KS-23113—1600 or 6250 bpi).

The following sections describe this system, its maintenance, and diagnostics.

Using the Magnetic Tape System

Standard procedures for mounting and dismounting a tape on the 3B20D Model 1 computer are described in 254-301-811, *Routine Tasks*.

Procedures for mounting and dismounting a tape on the 3B20D Model 2 and Model 3 computers are described in 254-302-831, *Routine Tasks (PDS Formats)* and 254-302-841, *Routine Tasks (MML Formats)*.

Consult these information products (IPs) before attempting any read or write activity.

Table 15-1 lists the input commands that control the MT system and describes their use. For more details on any specific input message, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Table 15-1. Magnetic Tape System Input Commands

Command	Description
COPY:TAPE:OUT	Copies disk files to tape. This message copies one or more files to tape along with full or relative pathnames and header information.
COPY:TAPE:IN	Copies files from a magnetic tape containing full or relative pathnames and header information and places them in their respective directories. Options are available to print a list of the files on the tape with their status information.
COPY:TAPE:EMERDMP	Copies data from the emergency dump partition on a disk to a tape.
COPY:BKDISK	Copies the boot disk to tape.

Operating, Maintaining, and Diagnosing the Magnetic Tape System

The following are guidelines for operating, maintaining, and diagnosing the magnetic tape system. Detailed procedures are available in the following IPs:

254-301-811, *Routine Tasks* (for Model 1)

254-302-831, *Routine Tasks (PDS Formats)* (for Model 2 and 3)

254-302-841, *Routine Tasks (MML Formats)* (for Model 2 and 3)

254-301-220, *Magnetic Tape System General Description 3B20D Model 1 Processor*

254-302-220, *Magnetic Tape System General Description 3B20D Model 2 Processor*

254-302-221, *Magnetic Tape System KS-23113 General Description 3B20D Model 2 and Model 3 Computer*

254-303-110, *PDS Input Messages Manual*

254-303-111, *PDS Output Messages Manual*

254-303-112, *MML Input Messages Manual*

254-303-113, *MML Output Messages Manual*

Guidelines for Tape Transport Operation

The following is a list of guidelines that help process data through the MT system:

- Before writing data to a tape, be sure that the tape is large enough to hold the data being written.
- Before using any of the commands in the input message manual, be sure you know the tape size and drive identification.
- Know the pathnames (relative and full) that specify the location to which or from which files are to be copied.
- Verify that the tape drive is on-line.
- If the system does not respond to commands, check the control panel for proper indications, such as the ON-LINE lamp and SELECT INDICATOR. Check if the tape is loaded to or past the load point [Beginning of Tape (BOT)].

Guidelines for Preventive Maintenance

The following are guidelines for preventive maintenance:

- Perform routine maintenance procedures (inspecting and cleaning) after every 8 hours of operation or monthly.
- Visually inspect the tape transport for cleanliness and obvious misadjustment each time it is used.
- Periodically check the operation of the MT device via the test control panel, located under the front panel nameplate, to verify proper operation.
- Before cleaning or adjusting, verify the MT device is off-line.



NOTE:

Adjustments to the MT device should be performed by personnel trained in transport maintenance. Adjustments are only performed if trouble is suspected. Electrical adjustments are preset at the factory and normally should *NOT* require readjustment.

Guidelines for MTC Diagnostics

Table 15-2 lists the input commands that request the MTC and associated IOP diagnostics. For more details on any specific input message, refer to 254-303-110, *PDS Input Messages Manual*, or 254-303-112, *MML Input Messages Manual*.

Table 15-2. MTC Diagnostic Commands

Command	Description
RMV:MT	Removes an MT device from service.
DGN:MTC	Runs diagnostics on the MTC. Several options can be specified to include the MT device.
RMV:MTC	Removes the MTC and any associated magnetic tape drives from service.
RST:MTC	Restores (conditionally or unconditionally) an MTC and magnetic tape transports to service. Conditionally, the unit is restored only if diagnostics report All Tests Pass (ATP); unconditionally, no diagnostic tests are performed. The IOP must be in service before the MTC can be restored.
RST:MT	Restores the specified magnetic tape transport to service. The IOP and MTC must be in service before the MT device can be restored.

Guidelines for running the diagnostics are as follows:

- Keep a 200-foot diagnostic test tape, J1P059AB-1, on hand. This tape is necessary when using demand diagnostics.
- Specify the **TLP** option when requesting diagnostics.
- Perform diagnostics after any maintenance procedure or adjustment.



NOTE:

Restoring the MTC conditionally always processes a request for diagnostics. MTC diagnostic phases test the MTC circuit pack and the control circuits in the IOP. MTC Phase 5 tests the tape formatter and tape transport.

Writing a File from a Disk

To write a file from disk to tape, use the following procedure:

Step	Procedure
1	Place a write ring on a tape large enough to hold all the files to be written.
2	Mount the tape.
3	Verify the tape drive is on-line.
4	If only one file is to be written, enter: In PDS, COPY:TAPE:OUT, TD="/dev/mtxx",FN="/d1/d2/.../f"! In MML, COPY:TAPE:DATA,OUT, TD="/dev/mtxx",FN="/d1/d2/.../f"; Where /d1/d2/.../f is the full pathname of the file to be written. ⇒ NOTE: xx = 08 for 1600-bpi or 00 for 6250-bpi tape drive. The xx value for tape drives is defined in 254-303-110, <i>PDS Input Messages Manual</i>, and 254-303-112, <i>MML Input Messages Manual</i>.
5	If two or more files are to be written, create a file containing, one per line, the full pathnames of the files to be written; enter: In PDS, IN:FILE:APND,FN="/tmp/tlist",LINE n/ "/d1/d2/.../f"! In MML, IN:FILE:DATA, APND,FN="/tmp/tlist",LINE= n/ "/d1/d2/.../f"; Where <i>n</i> is the line at which the file is to be appended. ⇒ NOTE: Before a file can be appended, the file must already exist.

Step	Procedure
	Then enter: In PDS, COPY:TAPE:OUT,TD"/dev/mtxx",FLIST"/d1/d2/.../f"! In MML, COPY:TAPE:DATA,OUT, TD="/dev/mtxx",FLIST="/d1/d2/.../f"; Where /d1/d2/.../f is the pathname of the list file (<i>/tmp/tlist</i> in the previous example).
6	If an abnormal termination occurs and STOPPED is printed, contact your next level of technical assistance.

Contents

Introduction	16-1
■ General Information	16-1
■ SCSDA Requirements	16-4
Device Translation	16-5
Scan Report Dispatching	16-5
Distribute Operation Requests	16-6
Office Alarms	16-6
Craft TTY Interface	16-6
■ ECD Requirements	16-7
■ Interfaces	16-8
SCSDH Message Interface	16-8
Client Process Message Interface	16-9
SCSD ECD Records	16-10
■ Definition of Logical Groups	16-11
■ Logical Addressing	16-14
■ Storing Data in the SCSD Option Block	16-15
■ Listing Logical Groups Defined in ECD	16-17
Capabilities and User Requirements	16-18
■ SCSDA Functions	16-18
■ Client Processes	16-18
■ UN33/UN933 Circuit Pack (SCSDC Board)	16-19
■ SCSD Handler	16-23
■ SCSD Features and Limitations	16-23
SCSDH Interfaces	16-24
Nondeferrable Maintenance Function	16-24
Restore SCSD PC	16-24

Contents

Remove PC	16-25
Operational Functions	16-26
Functional Messages	16-26
Command Response	16-27
Autonomous Report	16-28
SCSD Error Handler	16-29
Software Errors	16-30
■ Interface Requirements	16-30
SCSDH Message Interface	16-30
Client Process Message Interface	16-31
SCSDA ECD Requirements	16-35
■ Power Switch Monitor	16-39
PSM Interfaces	16-46
PSM Initialization	16-46
PSM Messages	16-47
PSM Message Structures	16-48
■ SCSD System and Alarm Requirements and Assignments	16-51
General Requirements	16-51
SCSD Point Assignments	16-52
■ Alarms	16-56
Fan Alarms	16-56
Detailed Design Description	16-57
Power Switch Monitor	16-57
Fan Alarms and PSM Operation Features	16-58
SCSD Commands	16-62
■ Access to Scan and Distribute Points	16-62
Growth and Degrowth Procedures	16-64
■ Preparation and Requirements for Growth and Degrowth Procedures	16-64
■ Growth Procedure for Adding a UN33/UN933 Board	16-65
■ Growth Procedure for Adding Scan Points on a UN33/UN933 Board	16-66
■ Degrowth Procedure for Removing Scan Points on a UN33/UN933 Board	16-68
■ Degrowth Procedure for Removing a UN33/UN933 Board	16-70
■ Procedure for Adding Logical Groups	16-70
■ Procedure for Deleting Logical Groups	16-72

Contents

■ Input/Output Processor Growth and Degrowth	16-74
Growth Requirements	16-75
Backout Procedures	16-75
Verifying SCSD Operation	16-76
■ Verifying Scan Point Status	16-76
■ Verifying SD Point Operation	16-78
■ Troubleshooting the SCSD	16-79

Introduction

This chapter provides information on how to write client processes, communicate with the Scan Point and Signal Distributor Handler (SCSDH), physically connect scan points, develop messages for maintenance personnel, grow and degrow UN33/UN933 boards, and grow and degrow the Input/Output Processor (IOP).

General Information

The Scan Point and Signal Distributor Controller (SCSDC) circuit pack (UN33/UN933) for the 3B20D and 3B21D computers has circuitry to monitor 48 scan points and to control 32 signal distributor points. The UN33 can be used in both 3B20D or 3B21D computers while the UN933 can only be used in 3B21D computers.

The Scan Point and Signal Distributor (SCSD) hardware is supported by software organized into three levels. The SCSDH level of the software forms an interface that handles the software functions for the hardware. The SCSDH is one of the handlers for the IOP driver operating under the control of the IOP driver.

The Scan Point and Signal Distributor Administrator (SCSDA) is the middle level of the software. The SCSDA provides support for the UN33/UN933 (SCSDC) by relaying operation requests to and from the SCSDH and the client processes.

The SCSDA does the following:

- Provides its client processes with scan point information.
- Controls the operation of distributor points.
- Controls the operation and inhibiting of scan points.
- Controls the logical-to-physical scan point address translation.

The SCSD option blocks in the Equipment Configuration Database (ECD) contain the necessary logical-to-physical translation data and scan point inhibit information.

The highest level of the software organization contains a variety of client processes including the Power Switch Monitor (PSM), the spooler alarm process, and the application processes. These are user-level processes and they all communicate with the SCSDA using the common client process messages. Of the client processes, the PSM will be discussed in more detail in the "Interface Requirements" section. The PSM interfaces with the following processes: Maintenance Input Request Administrator (MIRA), Configuration Manager (CONFIG), craft shell, and output spooler. The PSM receives change of device status messages from CONFIG. The PSM must have access to the Unit Control Block (UCB) option blocks (in the ECD) of the SCSDC to process the messages. The PSM also requires ECD access to send messages to MIRA, to output complete messages through the output spooler, and to properly handle requests from devices it monitors.

The software structure is shown in Figure 16-1. The three layers of software are graphically depicted in Figure 16-2.

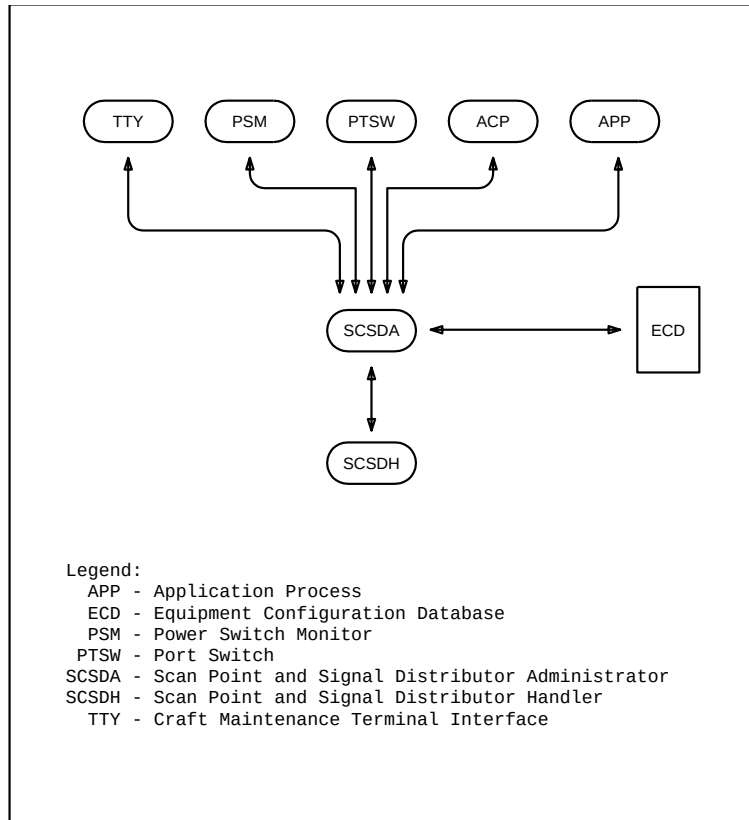


Figure 16-1. Software Structure

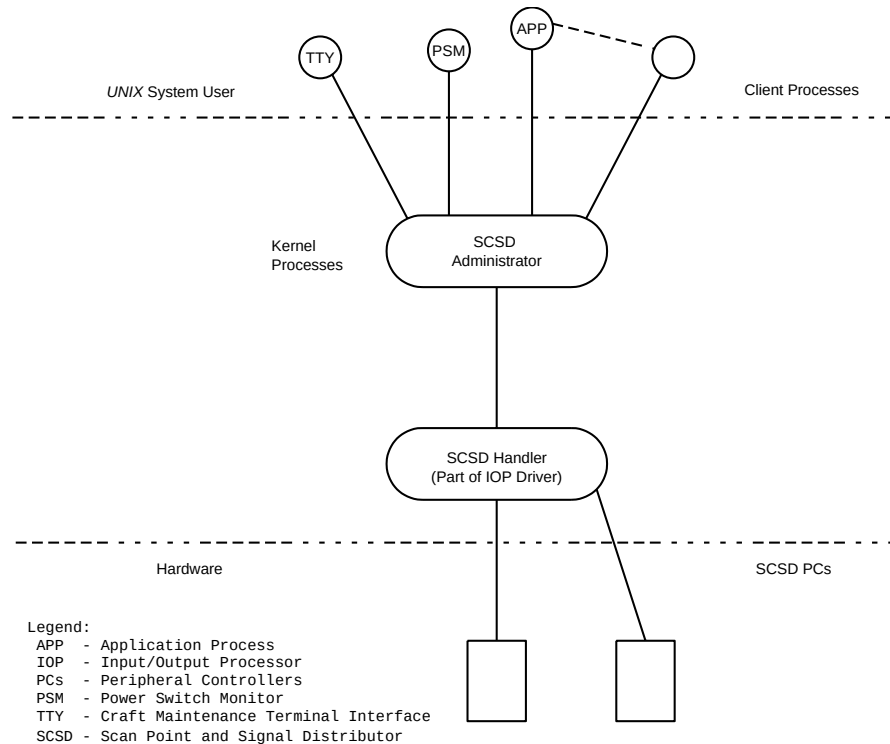


Figure 16-2. Three Levels of Software Interfacing SCSD Hardware

SCSDA Requirements

The SCSDA provides support for the SCSDC which is used for the control and monitoring of power control switches and for other miscellaneous scan and distribute functions. The SCSDA provides the following functions:

- Performs logical-to-physical and physical-to-logical device translation for scan and distribute points.
- Dispatches scan point transition reports autonomously generated by the SCSDC to the proper client processes, answers directed scan requests from the client processes, and relays scan point inhibit requests from the clients to the SCSDH.
- Relays distribute operation requests from the client process to the SCSDH.

- Activates audible alarms via distribute points upon request from the system output spooler.
- Provides a means by which the craft can set scan and distribute points and do scan point inhibits through Teletypewriter (TTY) messages.

Device Translation

Scan and distribute points of an SCSDC are functionally divided into logical groups. Client processes are concerned with these logical groups. For client processes to deal with the SCSDC on a logical basis, the administrator performs logical/physical translation for SCSD points. Logical points are identified by a unique group name and a scan or distribute point number within the group while physical points are identified by a pointer to the SCSDC's *ucb* record and its SCSDC point address. Each logical group is constrained to be entirely on a single SCSDC. Since group names are unique across all SCSDCs, client processes do not have to know which SCSDC a given logical group resides. Thus, this translation maps all physical SCSDCs onto a single logical device which is functionally divided into logical groups.

An exception to the rule of unique group names is the case of duplex groups. Duplex groups are two identical logical group names that contain pairs of scan or distribute points wired in parallel to the same scan source or distribute point destination. Each pair of duplex groups shares a common group name, and the two halves are distinguished by a separate single character duplex identification (dupid) number found in the *scsdbody* form in the dupid field (0 or 1).

Scan Report Dispatching

An SCSDC generates an autonomous scan report upon detecting a scan point state transition. This report is passed to the SCSDA via the SCSDH along with a pointer to the UCB of the reporting SCSDC. The SCSDA determines which point or points have changed state by comparing the scan map supplied by the transition report with a map maintained by the SCSDA. The latter map contains the previous state of the SCSDC, as observed by the SCSDA. The SCSDA generates a message for each logical group containing an uninhibited point which changed state, indicating the old and new states of all scan points in the group. This message is sent to the client process responsible for interpreting these scan points via the process number of the client process contained in the client process message structure of the SCSDA.

Directed scan requests are sent by client processes that want to know the current state of the scan points of an SCSDC. The SCSDA accesses the ECD in order to respond to the request. Client processes may request that individual scan points be inhibited from generating autonomous scan reports, and may read the inhibit status of individual scan points. The SCSDA performs logical-to-physical translation for such requests and relays them to the SCSDH for transmission to the SCSDC.

Distribute Operation Requests

Distribute requests are originated by client processes. There are four distribute operations: set, clear, flash continuously at 1 Hz, and flash for 8 seconds at 1 Hz, then clear. Client processes wishing to perform distribute operations send messages to the SCSDA indicating the desired logical group, the distribute point numbers within the group, and the operations to be performed.

The SCSDA performs logical-to-physical translation and formats a distribute command for each request. These commands are relayed to the SCSDC via the SCSDH.

Flashing of both halves of a duplex pair should be avoided since flash cycles on the two halves are not synchronized. The state of the distribute destination is determined by ORing both halves of a duplex pair, hence the result of a duplex flash operation is indeterminate.

Office Alarms

Office alarms are activated by distribute points on the SCSDC. The exception is when the alarm circuit is not electrically compatible with the SCSDC then the alarms are activated via an interface circuit.

The critical, major, and minor alarms are in a pair of duplex groups of four distribute points, three of which produce the alarm levels. The fourth distribute point is set by system detected alarms to distinguish them from hardware alarms. Operation of the audible alarms is accomplished by setting and clearing the pairs of distribute points.

Craft TTY Interface

Included in the SCSDA software is a utility which allows the craft to read scan points, set distribute points, and set and read inhibits on scan points. The craft may specify points by logical or physical addresses. The physical address allows the SCSD hardware SCSDC to deal with points which have not been assigned logical addresses (that is, points associated with expansion units).

ECD Requirements

This section describes the structure of the ECD.

The ECD is a memory-resident database that contains the attributes and current status of all processor and peripheral hardware known to the operating system. The structure of the ECD reflects the current hardware configuration of the system. The ECD was designed to do the following:

- Eliminate redundant device information.
- Provide a unified approach to the handling and accessing of device information.
- Be flexible enough to adequately represent many types of devices in several configurations.
- Allow read and write access by any process via Low-Level Access (LLA) primitives.
- Provide easy methods of initial and subsequent generation.

The ECD is composed of the following types of forms and records for the SCSDC:

- *mdct* – Contains information necessary to link a logical device to one or more physical devices.
- *scsdbody* – Gives information for one logical group of an SCSD option block.
- *scsdmap* – Contains review-only data from the SCSD option block.
- *scsdopt* – Gives options for the SCSD Peripheral Controller (PC) in the IOP. This information is used by the SCSDA to perform physical-to-logical device translation and to keep the status of the scan points.
- *ucb* – Describes a physical device.

Interfaces

The SCSDH handles the hardware dependencies while the SCSDA receives scan point information and controls distribute points through a message interface with the SCSDH. The SCSDA provides its client processes with scan point information, control over the operation of distribute points, control over inhibiting and allowing of scan points, and logical-to-physical scan point address translation.

SCSDH Message Interface

The SCSDA and the SCSDH communicate through process ports. Messages to the handler are sent to the IOP driver port and relayed by the driver to the handler. The SCSDA is attached to its own port.

Normal operational communication uses four message classes:

- Autonomous scan reports are generated in response to a scan point state transition in the SCSDC. These reports contain a map of the new scan point states.
- Directed scan commands ask for a map of the current scan states; this map is returned in the message response.
- Scan point inhibit commands are used to enable or disable a single specified scan point from generating autonomous reports and to read the setting of a single scan point inhibit.
- Distribute operation commands contain a field indicating which one of four states the point should be set.

Autonomous scan reports are sent by the SCSDH. The other messages are sent by the SCSDA. Autonomous scan and directed scan classes are further divided into A and B subtypes.

The A subtypes contain a map of scan points 0 through 31; the B subtypes contain a map of scan points 32 through 47. This subdivision arises because no more than 32 bits of scan map information may be transmitted in a single IOP autonomous report. All of the message classes contain a pointer to the UCB of the relevant SCSDC.

Initialization of the SCSDA and handler interface involves the use of two additional message classes: PC initialization and PC open. The PC initialization message is sent by the SCSDH with one message being generated for each SCSDC. This message informs the SCSDA that the indicated SCSDC is in a state where it can receive commands but will not generate autonomous reports. Receipt of the response of the SCSDA to this message indicates to the SCSD handler that the SCSDA is alive. When the SCSDA is prepared to receive autonomous reports from an SCSDC, it will send a PC open message for that SCSDC to the SCSDH. In the interval between the PC initialization and PC open

messages, the SCSDA may issue directed scan, distribute operation, and scan point inhibit commands. These commands may be used to initialize the SCSDC before opening it for normal operation. The aforementioned messages are used not only to initialize the SCSDA and handler interface, but also during normal operation to initialize an inactive SCSDC when it is being restored.

Client Process Message Interface

Communication for normal operation between the SCSDA and client processes is based on four message classes which are similar to the four SCSDH and SCSDA operational message classes. These messages differ only in that they may specify SCSD points by either logical or physical addresses. In addition, client processes may transmit multiple distribute operations or scan point inhibit requests in a single message.

Communication between the SCSDA and client processes also utilizes the port of the SCSDA. Requests from client processes are directed to this port. Client processes that are to receive autonomous scan reports must first send an open scan class message to the SCSDA indicating which class of logical scan groups the client is to monitor. The SCSDA retains the client's process number and associates it with the specified scan class. The SCSDA then sends the client a scan group initialization message for each group in the class. These messages indicate the initial states of all scan points in the group. Initialization messages are also sent out for groups contained in an SCSDC undergoing reinitialization during normal operation. When a client receives such a message, it indicates that any distribute points associated with the group are also initialized. An example of a client process is the power switch monitor. (Refer to the "Power Switch Monitor" section.)

SCSD ECD Records

This section describes the logical group information contained in the ECD option block for the SCSD. The *mdct* and *ucb* forms are used to set up the SCSDCs. The *ucb* form for an SCSDC also contains the option block name used by the SCSDC (that is, field 57) and is defined in the *scsdopt* form.

In addition, the option block name is used as a key to the *scsdbody* and *scsdmap* forms. Within the *scsdbody* form, the option name is used to identify and define logical group names and associated scan and distribute points controlled and monitored by the SCSDC.

The *scsdmap* form is a review only form that the SCSDA uses to keep track of all logical groups and their associated scan and distribute points of the SCSDC identified by the option block name. The following is the progression of the ECD forms:

mdct -> *ucb* -> *scsdopt* -> *scsdbody* -> *scsdmap*

Logical group information reflects the way an SCSD board (UN33/UN933) is cabled to scan and signal distribute points on other devices. This information is used by the SCSDA to perform physical-to-logical address translations and to keep track of the status of the scan points. Figure 16-3 is a sample of an *scsdbody* form. This form gives information for one logical group of an SCSD option block. Refer to 254-303-104, *Recent Change and Verify Manual*, for the ECD forms and an explanation of the fields.

SCSD Option Block Body.		scsdbody	(1/1)
			(Recent Change and Verify)
1. option_name:_____	2. scsdname:_____	3. dupid:___	4. sc_class:_____
5. complex_name:_____		6. complex_number:_____	
7. unit_name:_____		8. unit_number:_____	
9. sc_list.sc_entry		11. sd_list.sd_entry	
1)_____		1)_____	
2)_____		2)_____	
3)_____		3)_____	
4)_____		4)_____	
5)_____		5)_____	
6)_____		6)_____	

Figure 16-3. Sample *scsdbody* Form

Definition of Logical Groups

Each SCSDC has 48 scan points and 32 distribute points which can monitor scan points and control distribute points on other devices. Logical group names and logical point numbers are assigned to facilitate the process of monitoring and controlling SCSD points through the SCSDA. All logical groups monitored by a particular process are assigned to the same scan class. Each process monitoring a scan class is notified autonomously when any uninhibited scan point in their scan class changes state.

A logical group name contains a maximum of eight ASCII characters. Each logical group contains a variable number of scan and distribute points which are related in some way. All the points on a unit's power switch belong to one logical group, just as a collection of related alarm points belong in one logical group. A logical scan point is associated with a physical scan point, and a logical distribute point is associated with a physical distribute point. The scan points in a logical group are restricted to the range of 0 through 31 or the range of 32 through 47 because of the manner in which scan point transactions are reported.

All logical groups monitoring power switches are set up in the following way. Logical scan point 0 refers to the physical scan point connected to the X scan point on a power switch, and logical scan point 1 refers to the physical scan point connected to the Y scan point on a power switch. In addition, logical scan point 2 on the 3B21D computer refers to the physical scan point connected to the fuse alarm scan point on the Control Unit 0 (CU 0), CU 1, IOP 0, and IOP 1 power switches. The two distribute points on a power switch are Request In Progress (RQIP) and Out of Service (OOS) which are assigned logical distribute point numbers 0 and 1, respectively. Refer to Figure 16-4 for examples of 3B20D computer SCSD point assignments to logical groups and Figure 16-5 for examples of 3B21D computer SCSD point assignments to logical groups.

LOGICAL GROUP ASSIGNMENTS FOR SCSDC 0				
scsdname:	CU0	IOP1	MHD0	DFC0
scdupid:	0	0	0	0
sc_class:	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num:	2	2	2	2
sd_num:	2	2	2	2
sc_rid:	CU 0's rid	IOP 1's rid	MHD 0's rid	DFC 0's rid
sc_list[0]:	3	9	19	17
sc_list[1]:	2	8	18	16
sd_list[0]:	3	9	19	17
sd_list[1]:	2	8	18	16
scsdname:	PDF0	ACPALARM		
scdupid:	0	0		
sc_class:	PRSW_SCL	NULL_SCL		
sc_num:	2	0		
sd_num:	0	5		
sc_rid:	NULL	NULL		
sc_list[0]:	45			
sc_list[1]:	44			
sd_list[0]:		12		
sd_list[1]:		13		
sd_list[2]:		20		
sd_list[3]:		21		
sd_list[4]:		28		
LOGICAL GROUP ASSIGNMENTS FOR SCSDC 1				
scsdname:	CU1	IOP0	MHD1	DFC1
scdupid:	1	1	1	1
sc_class:	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num:	2	2	2	2
sd_num:	2	2	2	2
sc_rid:	CU 1's rid	IOP 0's rid	MHD 1's rid	DFC 1's rid
sc_list[0]:	11	1	27	25
sc_list[1]:	10	0	26	24
sd_list[0]:	11	1	27	25
sd_list[1]:	10	0	26	24

Figure 16-4. 3B20D Computer SCSD Point Assignments to Logical Groups Examples

LOGICAL GROUP ASSIGNMENTS FOR SCSDC 0				
scsdname:	CU0	IOP1	MHD0	DFC0
scdupid:	0	0	0	0
sc_class:	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num:	3	3	2	2
sd_num:	2	2	2	2
sc_rid:	CU 0's rid	IOP 1's rid	MHD 0's rid	DFC 0's rid
sc_list[0]:	31	23	26	28
sc_list[1]:	30	22	27	29
sc_list[2]:	46	42		
sd_list[0]:	31	23	27	29
sd_list[1]:	30	22	26	28
scsdname:	ACPALARM			
scdupid:	0			
sc_class:	NULL_SCL			
sc_num:	0			
sd_num:	4			
sc_rid:	NULL			
sc_list[0]:				
sc_list[1]:				
sd_list[0]:	12			
sd_list[1]:	13			
sd_list[2]:	14			
sd_list[3]:	15			
LOGICAL GROUP ASSIGNMENTS FOR SCSDC 1				
scsdname:	CU1	IOP0	MHD1	DFC1
scdupid:	1	1	1	1
sc_class:	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num:	3	3	2	2
sd_num:	2	2	2	2
sc_rid:	CU 1's rid	IOP 0's rid	MHD 1's rid	DFC 1's rid
sc_list[0]:	31	23	26	28
sc_list[1]:	30	22	27	29
sc_list[2]:	46	42		
sd_list[0]:	31	23	27	29
sd_list[1]:	30	22	26	28

Figure 16-5. 3B21D Computer SCSD Point Assignments to Logical Groups Examples

Each logical group may be a member of a scan class which is used to determine what processes are monitoring that logical group. Logical groups containing power switch SCSD points belong to the class PRSW_SCL. When a logical group does not belong to a specific scan class, NULL_SCL is given. This is used for groups that will not have scan point changes reported autonomously. Either such groups are not monitored by a process or they contain distribute points but no scan points. An example in some applications is the ACPALARM logical group which contains only distribute points.

The same group of points on a device can be monitored from two different SCSDCs. When logical groups are duplexed in this way, both of these groups have the same logical group name. A group's duplex identification number (DUPID equal to 0 or 1) is used to distinguish between duplex logical groups. Refer to 254-303-104, *Recent Change and Verify Manual*, for the definition of fields concerning logical devices.

Logical Addressing

Processes communicating with the SCSD administrator to monitor scan points and control distribute points use logical names, rather than SCSD unit numbers and physical point numbers, to specify SCSD points. The grouping of related SCSD points into logical units frees a process from having to depend on the hardware for some of its functions. As an example, the process does not need to know how the SCSD points are cabled to the SCSD board (UN33/UN933). A process can specify a scan point or a distribute point by giving a logical group name, a duplex identification (ID), and a logical point number. The SCSDA via its copy of the *scsdmap* forms, translates a logical address to a physical point number (0 through 47 for scan points; 0 through 31 for distribute points) on the proper SCSDC board.

Logical addressing provides more protection than specifying SCSD board numbers and physical point numbers. A process needs only the logical groups which contain scan points it monitors and distribute points it operates. This protects unassigned SCSD points or points in other logical groups from being accessed inadvertently. Access to any SCSD point can be attained by physical address (unit number and physical point number or *ucb* record ID of SCSDC). However, the use of logical addressing provides greater flexibility. Instead of changing processes when SCSD cabling is changed, it is only necessary to update information in the SCSDC *scsdbody* ECD form. This also applies to adding points to logical groups or adding entire logical groups. No processes would have to change since they are notified of any active logical group in the scan class being monitored.

Storing Data in the SCSD Option Block

Figure 16-6 shows the structure of an SCSDC's option block. Information is entered into the option block through an ECD form entry program. Device data is placed in the SCSD option block head. Information regarding logical groups is placed in the *scsdbody* and *scsdmap* forms.

```

extern  SUBSID scopt_rec;
struct  scopt_rec
struct  sc_map
{
    unsigned      sc_e1group:8 ;
    b_switch      sc_equipage:1 ;
    b_switch      sc_inhibit:1 ;
    b_switch      sc_state:1 ;
    unsigned      sc_0fill:5 ;
    unsigned      sc_e1fill:16 ;

    } scpt[48] ;
    unsigned      scsdgrp_num:8 ;
    char          sc_2fill[3] ;
    struct        scsdgrp
    {
        unsigned      sc_num:8 ;
        unsigned      sd_num:8 ;
        EQ_DUPLEX     scdupid:1 ;
        SC_CLASS      sc_class:5 ;
        char          scsdname[a];
        RID           sc_rid;
        byt_unsigned  (*sc_list) ;
        byt_unsigned  (*sd_list) ;

    } (*scsde1grp) ;
};

```

Figure 16-6. Structure of the SCSD Option Block in the ECD

The `sc_map` structure contains information on a scan point and the array `scpt[48]` contains one instance of `sc_map` for each scan point on an SCSDC. The `equpage` bit is set to 1 (1=on=y, 0=off=n) when the scan point is not used to monitor any device. When an SCSD board begins operation, any point which has a set `equpage` bit will be inhibited so its state transitions are not reported. This keeps state changes of unused scan points from being reported. The `sc_inhibit` and `sc_state` bits in the `sc_map` structure are used to keep track of the current inhibit status and state of scan points. These bits are maintained by the SCSDA and are initialized to 0 in the `scsdmap` form. The `sc_lgroup` entry in the `sc_map` structure contains an index into the array `scsdigrp[ ]` to indicate which logical group contains this scan point. If a scan point is not a member of any logical group, a value of -1 is assigned.

The number of logical groups on the SCSDC is stored in `scsdgrp_num`. The `scsdigrp` entry points to an array of `scsdigrp` structures which contain the information that defines each logical group. The `scsdname[ ]` entry is an array of up to eight ASCII characters that serves as the logical group name. A logical group name always begins with a letter and ends with a null character stored after the last character of the name. The `scdupid` contains a duplex identification number of 0 or 1. The `sc_rid` entry contains the record ID of the `ucb` entry for the unit that is monitored by this logical group. When no such ECD entry exists for a unit, a value of NULL is stored.

The scan class to which the group belongs is specified in the `sc_class` entry. The scan class `PRSW_SCL` contains all logical groups that are monitored by the Power Switch Monitor (PSM). There are at least six scan classes (`ALRM_SCL`, `USER1_SCL`, ..., `USER5_SCL`) available for application use and more classes can be added as needed. The `NULL_SCL` class is used to indicate that a logical group's scan point changes are not being monitored by any process. All scan classes are defined in the standard header file `<lla/ecd.h>`.

The number of scan points in a logical group is given in the `sc_num` entry. This is followed by `sc_list`, a pointer to an array containing the physical point number of the scan points in the group. For example, if scan point 0 of a logical group refers to the physical scan point 17, the entry `sc_list[0]` will contain the index 17. Physical scan points on an SCSD PC are numbered from 0 through 47, but the points in a given logical group must be within the range of 0 through 31 or the range of 32 through 47. This restriction is necessary because the SCSD PC or SCSDC reports scan point transitions in groups of 32 (maximum size). The map A scan report contains the states of scan points 0 through 31 and the map B scan report contains the states of scan points 32 through 47. The `sd_list` entry contains the number of distribute points included in this logical group, and each entry in the array included in this logical group. If the distribute point 0 in a group refers to physical distribute point 31, then the entry `sd_list[0]` contains the value 31. Physical distribute points on an SCSDC are numbered from 0 through 31.

Listing Logical Groups Defined in ECD

One way of obtaining a listing of the logical group names defined within the ECD's *scsdbody* form is to use the DBDUMP form. This will generate a dump file. Once the file is created, all of the *scsdbody* forms can be obtained via a **grep scsdbody dumpfile**<CR> command entered on the command line of a *UNIX*® system terminal [nonmaintenance Cathode-Ray Tube (CRT)]. The type of information that will be shown is as follows. (The headings are NOT part of the **grep** command output.)

Form Name	Option Name	Logical Group	DUPID	Scan Class	Unit Name	#	Scan Points	Distributor Points
scsdbody	scopt1	PRSWCU1	1	PRSW	CU	1	11 10	11 10
scsdbody	scopt1	PRSWIOP0	1	PRSW	IOP	0	1 0	1 0
scsdbody	scopt1	PRSWMHD1	1	PRSW	MHD	1	27 26	27 26
scsdbody	scopt1	PRSWDFC1	1	PRSW	DFC	1	25 24	25 24
scsdbody	scopt1	ACPALARM	1	NULL				12 13 20 21 28
scsdbody	scopt1	FANACU1	1	PRSW	CU	1	30 31	30 31
scsdbody	scopt1	PRSWMHD3	1	PRSW	MHD	3	23 22	23 22
scsdbody	scopt0	PRSWCU0	0	PRSW	CU	0	3 2	3 2
scsdbody	scopt0	PRSWIOP1	0	PRSW	IOP	1	9 8	9 8
scsdbody	scopt0	PRSWMHD0	0	PRSW	MHD	0	19 18	19 18
scsdbody	scopt0	PRSWDFC0	0	PRSW	DFC	0	17 16	17 16
scsdbody	scopt0	ACPALARM	0	NULL				12 13 20 21 28
scsdbody	scopt0	PTSWROP	0	PTSW	ROP	0	15 38 39	
scsdbody	scopt0	PTSWMCR	0	PTSW	MTTY	0	11 36 37	
scsdbody	scopt0	PDF0	0	PRSW			45 44	
scsdbody	scopt0	FANACU0	0	PRSW	CU	0	30 31	30 31
scsdbody	scopt0	PRSWMHD2	0	PRSW	MHD	2	23 22	23 22
scsdbody	scopt0	PRSWIOP2	0	PRSW	IOP	2	7 6	7 6

The first column represents the form that was searched for while the second column represents the option block names used by the SCSDCs (in this case, SCSDC 1 and SCSDC 0). The third column represents the logical group names defined in the ECD. The remaining columns represent the DUPID, scan class, unit monitored (if any), and the scan and distribute points associated with each of the logical group names.

Capabilities and User Requirements

This section describes the capabilities and requirements of the SCSDA, client processes, SCSDC, and the SCSDH.

SCSDA Functions

The SCSDA provides the following functions:

- Performs logical-to-physical and physical-to-logical device translation for scan and distribute points.
- Dispatches scan point transition reports autonomously generated by the SCSDC to the proper client processes.
- Answers directed scan requests from the client processes.
- Relays scan point inhibit requests from the clients to the SCSDH.
- Relays distribute operation requests from the client processes to the SCSDH.
- Activates audible alarms via distribute points upon request from the system output spooler.
- Provides through TTY messages a means to read and set scan and distribute points.
- Provides through TTY messages a means to do scan point inhibits.

Client Processes

Normal operational communication between the SCSDA and the client processes are based on four classes of messages that specify SCSD points by either logical or physical addresses. The physical address mode is provided for the benefit of the craft TTY utility. Client processes may also transmit multiple distribute operation or scan point inhibit requests in a single message.

Communication between the SCSDA and client processes utilize the port of the SCSDA. Requests from client processes are directed to this port. Client processes that wish to receive autonomous scan reports must first send an Open Scan Class message to the SCSDA indicating which class of logical scan groups the client wishes to monitor. The SCSDA notes the process number of the client and associates it with the specified scan class. The SCSDA then sends the client a scan group initialization message for each group in the class, indicating the initial states of all scan points in the group. Initialization messages are also sent out for groups contained on an SCSDC undergoing reinitialization during normal operation. Clients that receive a reinitialization message should assume that any distribute points associated with the group must also be initialized.

UN33/UN933 Circuit Pack (SCSDC Board)

The SCSDC consists of circuitry for monitoring 48 scan points and controlling 32 distribute points. Refer to the UN33 block diagram shown in Figure 16-7 and the UN933 block diagram shown in Figure 16-8. A scan point is considered to be a set of open pair leads connected to a monitored current source on the UN33/UN933. The characteristics of the scan and signal distributor points are as follows:

- Scan Points:
 - Each scan point is examined every 48 milliseconds, and the scan map (in the SCSDC board) is updated with any change.
 - A point must remain in the new state for two scans (96 ms) for the change to be reported to the Control Unit (CU).
- Signal Distributor Points:
 - Activated by commands from the CU. Only one point at a time may be activated or deactivated.
 - SD points may be directed by the CU to any one of four states:
 - Operate
 - Clear
 - Flash continuously
 - Flash for 8 seconds, then terminate to the off state.
- Applications/Responses:

The SCSD points are connected to each unit of the 3B20D/3B21D computers. However, there are no SCSD points connected to the 9-track magnetic tape unit of the 3B20D computer. The power switch circuits of the units all use SCSD points and have similar responses. The following are the expected responses from the units:

 - Any manual powerup or powerdown results in a message.
 - Any automatic powerdown results in an alarm and a message.
 - Any nonpowerdown fault results in an alarm and a message.
 - Switching the ROS/RST switch to Request Out of Service (ROS) requests removing the unit from service.
 - The RQIP Light-Emitting Diode (LED) is lit when a request (either a remove or restore) is in progress. If a remove request is denied, the RQIP LED is flashed several times then extinguished. Remove and restore commands made from the unit do not operate the RQIP LED.

- The OOS LED is lit when the unit is out of service.
- Switching the ROS/RST switch to Restore (RST) requests a diagnostic and restore of the unit to service.
- Diagnostics will not be run on powerup if the ROS/RST switch is left in the ROS position.
- Switching the Alarm Cutoff Test (ACO/T) switch to ACO/T and back again retires an alarm associated with the particular powered down unit.



NOTE:

The other functions — Manual Override (MOR), lamp test, and the ROS lamp — do not interface with the SCSD. The power switch on the disk drive does not interface with the SCSD.

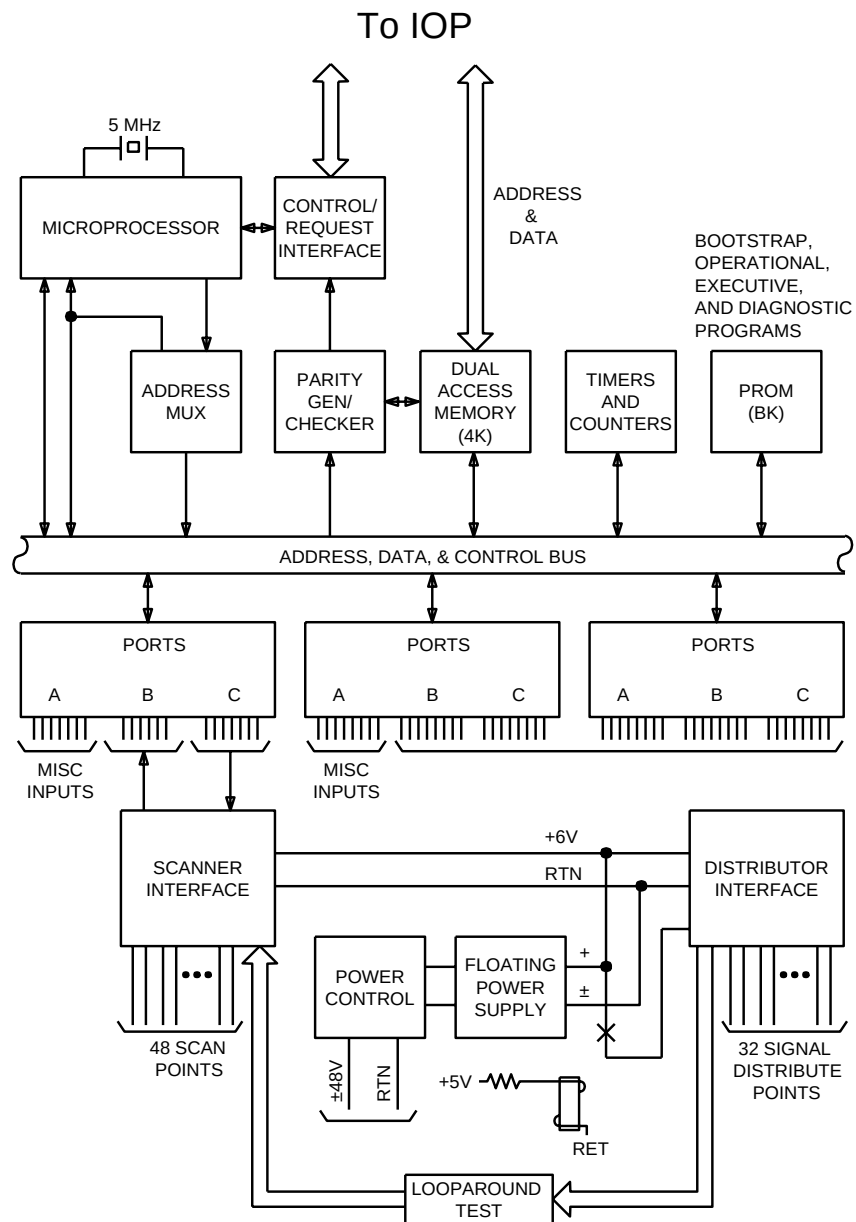


Figure 16-7. UN33 Block Diagram

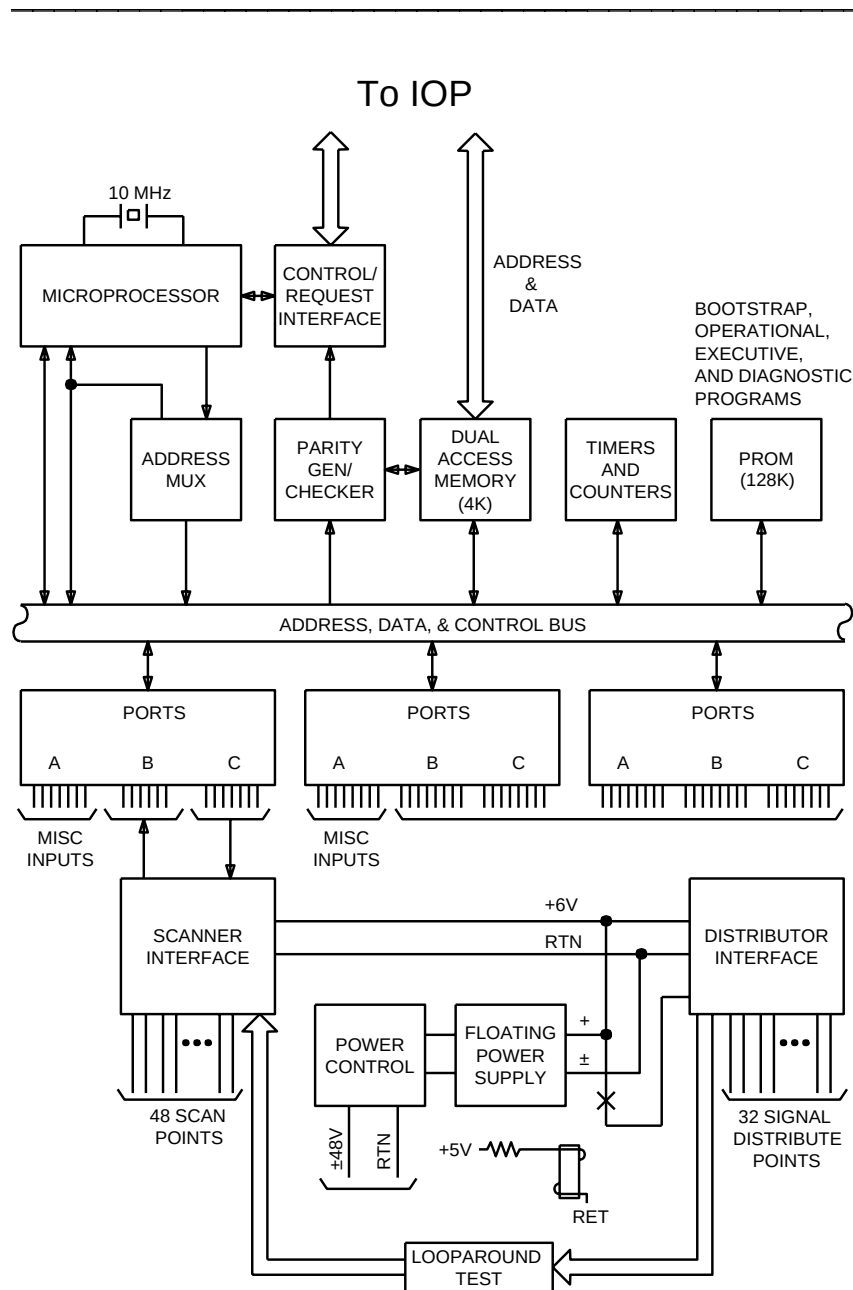


Figure 16-8. UN933 Block Diagram

SCSD Handler

The purpose of the Scan Point and Signal Distributor Handler (SCSDH) is to provide a user/device interface with the SCSD Peripheral Controller (PC), transfer the status of the scan point image (which is maintained by the SCSD controller scan map), operate SD points, and inhibit troubled or spare scan points. The SCSD handler makes up the SCSD system that includes the SCSDA and any client processes that communicate with it. The following lists the SCSD handler attributes:

- Is a part of IOP driver.
- Provides user/device interfaces.
- Isolates rest of the system from details of hardware.
- Functions are initiated by IOP driver.
- Performs nondeferrable functions.
- Provides autonomous report capability.
- Requires no Direct Memory Access (DMA) operation in transmitting or receiving data from or between PC and the administrator.
- Handles hardware and software errors.

SCSD Features and Limitations

The following is a list of the features and limitations of the SCSD system:

- The SCSD system is composed of the SCSDH, SCSDA, and any client processes attached to the SCSDA.
- Each PC is equipped with 48 scan points and 32 distribute points.
- More than one PC can be equipped on each IOP.
- The SCSDH can accommodate any number of PCs attached under same or different IOP.
- No DMA operation is involved in transmitting or receiving the data.
- Each point can be connected in parallel.
- Spare points or the troubled scan points can be inhibited by sending inhibit scan point message.
- The SCSDH communicates only with the administrator.
- The SCSDH can transmit or receive only 32-bit scan point information at a time.
- The SCSDH does not keep the copy of the scan map.

- No retry is made in case of failure or error condition. Instead an error message is reported to the Receive-Only Printer (ROP).
- In case of remove, SCSDH sends SC_DOWN message to administrator.
- In case of restore, SCSDH sends SC_UP message to administrator.

SCSDH Interfaces

The SCSDH to SCSD administrator interface has the following features:

- Messages are used in communication.
- Communication is done via ports.
- The SCSDA process is attached to a port.

The SCSDH to SCSD PC interface has the following features:

- Communication with PC is done through IOP driver and Peripheral Interface Controller (PIC).
- Commands are used in communication.
 - High priority commands which are sent directly to the PC.
 - Base priority commands are queued.

The SCSD PC to SCSDH interface has the following functions:

- Communicates with SCSDH via PIC and IOP driver.
- Sends command responses.
- Generates autonomous base priority reports.

Nondeferrable Maintenance Function

Restore SCSD PC

When the SCSD PC is to be restored during the boot process or the restart condition, the IOP driver passes control to the SCSDH restore function. The restore function is initiated by IOP driver during a boot operation or restart function. The following are the attributes of the restore function:

- Aborts all jobs (in case of restart).
- Sends the PC clear command by calling the IOP driver service routine **pcphyrmv**.
- Sets the device status to the active state.
- Constructs the begin operation command to PC.
- Sends the begin operation command to PC.

- Waits for a response from the PC.
- Generates the SCSD INIT message upon receiving a successful command response.
 - Constructs the SCSD INIT message:
 - Gets the message buffer pointer.
 - Gets the SCSDA process number attached to a port.
 - Inserts the following information in the message:
 - Process number.
 - Type of message.
 - Message status as good.
 - Sends a message to the administrator through a port.
- Waits for acknowledgment from the administrator.
 - If the returned message status is a NULL process number or Message Dead (MSDEAD):
 - Reports to the SCSD error handler.
 - The SCSD error handler reports to the Configuration Manager (CONFIG).
 - Removes the PC from active service.

Remove PC

The PC is removed in case of a deferrable remove request or because of critical errors during normal operation. The SCSDH reports the error to CONFIG. The CONFIG may subsequently report to the IOP driver specifying that the PC should be removed from service. At this point, the user cannot establish communication with the PC until it is restored to service. The remove functions are as follows:

- Abort all pending jobs.
- Send PC clear command by calling IOP driver service routine **pcphyrmv**.
- Set device status to out-of-service state.
- Clear all 32 distribute points.

Operational Functions

The operational functions establish communication with the SCSDH through the IOP driver. The operational functions have the following requirements:

- All operational functions are requested by the administrator.
- The 48 scan point map is grouped into Maps A and B.
- Map A carries 0 through 31 scan point map.
- Map B carries 32 through 47 scan point map.
- Each command can transfer only 32 bit scan point image at a time.

The types of these operational functions as previously outlined are:

- Directed scan map request.
- Inhibit scan point.
- Distribute point operation.
- Open autonomous report, that is, allow scan point report dispatching.

Functional Messages

The administrator sends messages to the SCSDH through the IOP driver. The SCSDH checks the validity of the messages. Communication from the administrator is via functional messages as summarized in the following lists:

- Constructs PC command:
 - Directed scan map.
 - Insert Maps A or B opcode.
- Distributes Point Operation:
 - Insert opcode.
 - Distribute point address.
 - Type of operation.
- Inhibits Scan Point:
 - Insert opcode.
 - Inhibit scan point address.
 - Type of operation.
- Opens Autonomous Reports:
 - Insert opcode.

- Saves message pointer in device scratch area.
- Saves image of command Words 1 and 2 in the message.
- Passes the command to PC using IOP driver service routine PUSH Q1.
- Sets timer for 2 seconds.
- Waits for command response.

Command Response

Once a PC command is executed, the SCSD PC generates a command response including the job completion code (through the PIC) to the IOP driver. The IOP driver then invokes the SCSDH command response function. The command response function interprets the response and generates a status report to the SCSDA which includes errors. The status report is placed in the status word of the message header and an acknowledgment message is returned to the SCSDA through the normal Message Acknowledge (MSACK) system call **MESSINK**. The following is a list of the items in the command response function:

- The PC sends command responses after completing the requested job through PIC to the IOP driver.
- The IOP driver passes the device *ucb* pointer to the SCSDH.
- The SCSDH gets the device scratch area pointer from the *ucb* pointer.
- Recovers timer information.
- Terminates two second timer request.
- Checks the following conditions:
 - Saved command opcode is compared with response opcode.
 - Device status active.
- Checks completion code for errors.
- Reports to the SCSD error handler in case of condition failure or error.
- Checks type of message:
 - Directed scan Map A or Map B.
 - Image of Map A or Map B from command Word 2 is placed in the message.
- Sets proper status in the message to inhibit scan map. In case of read inhibit scan map operation, sets the state of the point in the message.
- Sets proper status in *ms_stat* word of the message.
- Returns the message back to the administrator using system call **MESSINK**.

Autonomous Report

The SCSD PC sends an autonomous report through PIC to the IOP driver; then the IOP driver invokes the SCSDH response function. The response function checks the validity of the autonomous report and generates a message (in the case of a Map A or Map B report) or reports to SCSD error handler (in the case of error report) and passes the information to the administrator. The following actions take place in respect to an autonomous report:

- SCSD PC generates an autonomous report because of change in scan state during an autonomous scan.
- Asynchronous autonomous report is passed to the IOP driver through PIC.
- IOP driver gets the *ucb* pointer and enters the response function.
- Checks conditions before processing the autonomous report:
 - Device status active.
 - Autonomous report opcode.
- Checks type of report:
 - Error report.
 - Checks type of error.
 - Report to SCSD error handler.
- Scans Maps A or B report:
 - Gets message buffer pointer.
 - Gets the administrator process number.
 - Inserts the following information in the message:
 - Process number.
 - Copy of Maps A or B from the second word of the **REPORT** command.
 - Type of message.
 - Status of the message.
 - Message is passed to the administrator through port.
 - SCSDH does not wait for acknowledgment from the administrator.

SCSD Error Handler

When an error is detected, the SCSDH identifies the software error or the completion code error and passes an error number to the error handler. Software errors are detected by the PIC or PC and reported through the response to the command. Upon receiving an error, the error handler recognizes the critical and noncritical type of errors. The critical type of errors are reported to CONFIG and noncritical type of errors are printed on the terminal, ROP, and/or *IODRVLOG* file as a warning signal. The following summarizes the activities of the error handler:

- All errors recognized by the SCSDH are reported to the error handler.
- PIC and PC hardware errors are reported in the response command or the autonomous error report.
- Error conditions are placed in the 12-bit completion code of the command.
- Software errors are reported by the handler.
- Three types of errors are recognized.

— PIC errors:

Displays type of error and completion code.

Reports to CONFIG for further action.

— PC errors (critical):

Displays type of error and completion code.

Reports to CONFIG for further action.

— PC errors (noncritical) — displays type of error and completion code.

— Software errors (critical):

Displays type of error and completion code.

Reports to CONFIG for further action.

— Software errors (noncritical) — displays type of error and completion code.

Given a critical error, the error handler calls the IOP driver service routine *OUTERROR* to print the error. In case of a software error, a proper error number is displayed on the terminal, ROP, and/or *IODRVLOG* file. If the error is detected from the command response, the error number and the contents of the completion code is displayed on the terminal, ROP, and/or *IODRVLOG* file. Then the IOP driver report error function *RPTERR1* is called to report the error to CONFIG. CONFIG may subsequently send a message to the IOP driver specifying that the controller should be restarted or removed from service.

Software Errors

If the error reported by the SCSDH is a noncritical type of error, the IOP driver service routine OUTERERROR is made to print the error. In case of a software error, an error number is displayed on the terminal, ROP, and/or *IODRVLOG* file. If the error is detected from the command response, the error number and the contents of the completion code are displayed on the terminal, ROP, and/or *IODRVLOG* file. The following are software errors:

- Device not active
- Administrator process not active
- Bad message type
- Command Q full
- Time out error
- Scan point address error
- Distribute point address error
- Response command error.

Interface Requirements

SCSDH Message Interface

Messages to the SCSDH are sent to the IOP driver port and relayed to the SCSDH by the IOP driver. The SCSDA is also attached to a port, and the SCSDH directs its messages there.

The IOP driver and the SCSDA are created at boot time or sometime later by the 3B20D/3B21D computer operating system. When the SCSDH is ready to communicate to the SCSDA, it sends the SCSDA a growth message that contains the Record Identification (RID) of the SCSDC being grown by the system.

The SCSDA inserts the RID of the SCSDC, the option block pointer, and the unit number in its device table scratch area. The SCSDA then sends an initialization message (SC_INIT) to the SCSDH. The SC_INIT message notifies the SCSDH that the SCSDA is ready to receive scan point initialization information. One SC_INIT message is sent for each SCSDC in the system. A *ucb* pointer is used to distinguish SC_INIT messages from one another. The SCSDH responds to SC_INIT messages with MSACK messages containing the *ucb* pointer and the entire scan map for each active SCSDC.

After receiving the first MSACK message from the SCSDH, the SCSDA sends the SCSDH a distribute request. This request sets the appropriate distribute point to notify the craft that a phase has taken place. Refer to Table 16-1 for the phase alarm levels. The SCSDA also outputs a phase message to the spooler. Using the equipage map in the SCSD option block, the SCSDA sends one inhibit message for each unequipped scan point on an SCSDC to the SCSDH.

Table 16-1. Phase Alarm Levels

Phase Level	Alarm
1	Minor Alarm (set by the EIH)
2	Major Alarm
3	Major Alarm
4	Critical Alarm

The SCSDH does not start sending autonomous reports until it receives an open message (SC_OPEN) for that SCSDC. An SC_OPEN message is sent only after a client process is available to monitor some scan points on that SCSDC.

The SCSDH notifies the SCSDA when an SCSDC is being taken out of service or when it has been restored or restarted. The client processes are notified by the SCSDA that the SCSDC has been removed or restored. While the SCSDC is out of service, the SCSDA and its client processes must suspend their activities on this SCSDC until it has been restored to service. Once the SCSDC has been restored, the SCSDA and its client processes go through the same initialization procedure as when first created.

The following information is applicable to *UNIX* Real Time-Reliable (RTR) Operating System Release 21.7 and later. When the SCSDC is being taken out of service, the SCSDH sends the SCSDA an SC_DOWN message with an Input/Output (IO) type operation of SC_OOS. Upon receipt, SCSDA sends a distribute operation request to the SCSDH to clear all 32 distribute points.

Client Process Message Interface

Any user process that wants to communicate with the SCSDA uses the message structure *adminhdr*. (Refer to Figure 16-9.) The PSM and the SCSD commands use this client process message format to communicate with the SCSDA. Messages sent by client processes are directed to the port of the SCSDA. They specify an SCSD point by physical-or-logical address. The SCSDA translates a logical address into physical address before sending the request to the SCSDH. The SCSDH can communicate with the SCSDC only by physical scan point addressing.

```

/*
 * SCSD ADMINISTRATOR HEADER FILE
 * This file contains message structures and defines for
 * Administrator - Client Process Communication.
 *
 * The ecd.h header file contains the definition of the b-switch
 * enumerated type used in this structure.
 * Therefore the file ecd.h
 * must be included prior to this file.
 */
/*
 * SCSD Administrator - Client Process Message Structure
 */
Struct adh_body
{
    char          ad_type;          /* type of operation */
    long          ad_rid;           /* phys addr mode: record id for SCSD */
    long          ad-grpid;         /* log addr mode: log group unit rid */
    char          ad_group[9];      /* log addr mode: logical group name */
    char          ad_unit;          /* physical addr in form of int unit */
    char          ad_flags;         /* flags - see below */
    short         ad_class;         /* scan class for open class msg type */
    short         ad_count;         /* # of points in maps or # of requests */
    short         ad_errcd;         /* error code */
    struct adminhdr *ad_chain;      /* admin request link */
    union
    {
        struct statemap
        {
            b_switch  new:1;        /* current state of scan pt */
            b_switch  inhibit:1;    /* inhibit state of point */
            b_switch  old:1;        /* old state of point */
        }
        scan_pt[48];
        struct
        {
            char operation;          /* dist. inhibit opcode */
            char pt_num;             /* scan, dist. point number */
            char result ;            /* return code for operation */
        }
        rqsts[8];
    }
    ad_data
};
struct adminhdr
{
    struct msghdr    ad_hdr;        /* standard message header */
    struct adh_body  ad_body;       /* body of admin-client message */
};

```

Figure 16-9. Administrator Message Structure

Four client process message types (AD_OPEN, AD_INIT, AD_DOWN, and AD_UP) are analogous to the SCSDH and SCSDA message types. A client process can initiate a directed scan request, a distribute operation, and an inhibit request. Autonomous scan reports are sent to a client when a scan point it monitors undergoes a transition.

Multiple distribute operations and inhibit scan point requests can be contained in a single message to the SCSDA. A directed scan request asks for the current state of scan points from the ECD. The request can specify a single scan point by logical or physical address, a logical group, or an entire SCSDC.

Normal operation of a client consists of continuously monitoring scan point transitions and requesting appropriate distribute and inhibit operations. An autonomous scan point report is the mechanism by which a client is notified of a scan point state change.

By sending an open scan class message (AD_OPEN), the client initiates autonomous scan point reporting. An AD_OPEN message contains the name of the scan class the client wishes to monitor.

The SCSDA replies by sending the client an initialization message (AD_INIT) for each logical group on an active SCSDC. The AD_INIT message contains a complete scan point state map for a logical group. AD_INIT messages are sent out to client processes when an SCSDC undergoes a restart or a reinitialization during normal operation.

When a client process receives an AD_INIT message, it initializes any distribute points in that logical group. If a logical group in a scan class resides on an SCSDC that is not active, the SCSDA sends a USRACK or ADMMSG message with the appropriate error codes to the client process notifying it that the SCSDC is unavailable (ER_UNAVL). Figure 16-10 shows the error codes which are contained in the *ad_errcode.h* file. When the SCSDC becomes active, the SCSDA sends the client process an AD_INIT message.

The SCSDA notifies its client processes when an SCSDC is being removed with an AD_DOWN message and when an SCSDC is being restored with an AD_UP message. This informs client processes as to the current state of the SCSDC.

Another SCSDA message type is the AD_CLOSE message. This type is available in *UNIX* RTR Operating System Release 21.3 and later. A client process can use this message type to instruct the SCSDA to terminate it silently. That is, maintenance personnel will not be informed of the termination of the client process via an output report to the ROP. This may be useful for client processes that are terminated by manual craft initializations.

```
/*
 * SCSD ADMINISTRATOR HEADER FILE
 * This file contains error codes used by the SCSD
 * Administrator and Client Processes.
 */
/*
 * Error Codes      (ad_errcode.h)
 */
/* error in ucb pointer */
#define ER_UCBLP 0xe0
/* error in SCSD unit number */
#define ER_UNUM 0xe1
/*error in logical group name */
#define ER_LOGN 0xe2
/* dupid needed to complete operation */
#define ER_DPND 0xe3
/* the SCSD is unavailable */
#define ER_UNAVL 0xe4
/* error in the scan point number specified */
#define ER_PTNUM 0xe5
/*address type not specified */
#define ER_ADDRT 0xe6
/*message to the handler failed */
#define ER_HDLRF 0xe7
/* illegal class */
#define ER_CLASS 0xe8
/* too many clients for scan class, class_list [ ][ ] too small */
#define ER_ENUM 0xe9 /* unknown command type */
#define ER_ADTYPE 0xea
/* incorrect state to process request or response */
#define ER_ADSTAT 0xeb
/* error returned on a lla_rdget call */
#define ER_LLAG 0xed
/* no error */
#define ER_NONE 0xee
/* error illegal operation requested */
#define ER_OPER 0xef
/* error point from database */
#define ER_DBPTNUM 0xec
/* request number per message in error */
#define ER_CNTNUM 0xf0
/* no form for class requested to be monitored */
#define ER_DBCLASS 0xf1
/*ms_stat of return message form handler is FAIL */
#define ER_RHMSF 0xf2
/* the return message from handler io_type unknown */
#define ER_RHIOT 0xf3
/* error returned form lla_uprid() */
#define ER_LLAU 0xf4
```

Figure 16-10. Error Codes for Administrator and Client Processes

SCSDA ECD Requirements

Each SCSDC has a separate minor device chain table (*mdct*) entry in the ECD. The only exception to this is when a pair of SCSDCs have scan or distribute points connected in parallel to the same scan source or distribute destination. These SCSDC points are referred to as duplex pairs. Because they are redundant units, they share a single *mdct* entry in the ECD. The SCSDA requires that the *ucb* option block of each SCSDC contains translation data. (Refer to Figure 16-11 option block layout.) The SCSDA must have access to this information to accomplish the following tasks:

- Translate logical-to-physical SCSD point addresses.
- Initialize the inhibits on scan points.
- Identify duplexed groups and duplexed points.
- Determine scan point transitions by comparing current scan point states with autonomous reports from the SCSDC.
- Determine which client process to notify of a scan point transition.
- Maintain a current scan point status map.
- Maintain a current scan point inhibit map.

```

extern  SUBSID scopt_rec;
struct  scopt_rec
struct  sc_map
{
    unsigned      sc_elgroup:8 ;
    b_switch      sc_equipage:1 ;
    b_switch      sc_inhibit:1 ;
    b_switch      sc_state:1 ;
    unsigned      sc_0fill:5 ;
    unsigned      sc_e1fill:16 ;

    } scpt[48] ;
    unsigned      scsdgrp_num:8 ;
    char          sc_2fill[3] ;
    struct        scsdgrp
    {
        unsigned      sc_num:8 ;
        unsigned      sd_num:8 ;
        EQ_DUPLEX      scdupid:1 ;
        SC_CLASS      sc_class:5 ;
        char          scsdname[9] ;
        RID          sc_rid;
        byt_unsigned   (*sc_list) ;
        byt_unsigned   (*sd_list) ;

    } (*scsdlgrp) ;
} ;

```

Figure 16-11. ECD Option Block Record

When an SCSD command or a client process sends the SCSDA a message containing a logical point number, that address is translated to a physical format before a scan or distribute operation is done. To translate a logical address, the SCSDA searches the *ucb* option block of each SCSDC until the correct logical group name is found. (Refer to Tables 16-2 and 16-3 for a list of the group names.) If the logical group is duplexed, then two groups share a common name and they are distinguished by their dupid. The logical point number in the *sc_list* is then used to chain through the linked list to find the physical point number. With the physical address, the SCSDA can determine the state of a scan point and inhibit status from the data in the *ucb* option block, or request inhibit or distribute operations of the SCSDH.

The translation of a physical address to a logical address requires that the SCSDA use a *ucb* pointer to access the correct *ucb* option block, and then use the physical point number as an index into the scan group list to find the logical group name. This type of translation is done when one or more scan points change state and client processes are notified. The scan class entry in the *ucb* option block is used to determine which clients to notify of a scan point transition.

Table 16-2. 3B20D Computer Logical Group Assignments Examples

scsdname	PRSWCU0	PRSWIOP1	PRSWDFC0	PRSWMHD0	PDF0
scdupid	0	0	0	0	0
sc_class	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num	2	2	2	2	2
sd_num	2	2	2	2	0
sc_list[0]	3	9	17	19	45
sc_list[1]	2	8	16	18	44
sd_list[0]	3	9	17	19	NONE
sd_list[1]	2	8	16	18	NONE
scsdname	FANACU0	PTSWROP	PTSWMCR	ACPALARM	
scdupid	0	0	0	0	
sc_class	PRSW_SCL	PTSW_SCL	PTSW_SCL	NULL_SCL	
sc_num	2	3	3	0	
sd_num	2	0	0	5	
sc_list[0]	31	15	11	NONE	
sc_list[1]	30	38	36	NONE	
sc_list[2]	NONE	39	37	NONE	
sd_list[0]	31	NONE	NONE	12	
sd_list[1]	30	NONE	NONE	13	
sd_list[2]	NONE	NONE	NONE	20	
sd_list[3]	NONE	NONE	NONE	21	
sd_list[4]	NONE	NONE	NONE	28	
scsdname	PRSWCU1	PRSWIOP0	PRSWDFC1	PRSWMHD1	FANACU1
scdupid	1	1	1	1	1
sc_class	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num	2	2	2	2	2
sd_num	2	2	2	2	2
sc_list[0]	11	1	25	27	31
sc_list[1]	10	0	24	26	30
sd_list[0]	11	1	25	27	31
sd_list[1]	10	0	24	26	30

Table 16-3. 3B21D Computer Logical Group Assignments Examples

scsdname	CU0	IOP1	DFC0	MHD0	
scdupid	0	0	0	0	
sc_class	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL	
sc_num	3	3	2	2	
sd_num	2	2	2	2	
sc_list[0]	31	23	28	26	
sc_list[1]	30	22	29	27	
sc_list[2]	46	42	NONE	NONE	
sd_list[0]	31	23	29	27	
sd_list[1]	30	22	28	26	
scsdname	FANACU0	PTSWROP	PTSWMCR	ACPALARM	
scdupid	0	0	0	0	
sc_class	PRSW_SCL	PTSW_SCL	PTSW_SCL	NULL_SCL	
sc_num	1	3	3	0	
sd_num	1	0	0	4	
sc_list[0]	8	43	47	NONE	
sc_list[1]	NONE	41	45	NONE	
sc_list[2]	NONE	40	44	NONE	
sd_list[0]	8	NONE	NONE	12	
sd_list[1]	NONE	NONE	NONE	13	
sd_list[2]	NONE	NONE	NONE	14	
sd_list[3]	NONE	NONE	NONE	15	
scsdname	CU1	IOP0	DFC1	MHD1	FANACU1
scdupid	1	1	1	1	1
sc_class	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL	PRSW_SCL
sc_num	3	3	2	2	1
sd_num	2	2	2	2	1
sc_list[0]	31	23	28	26	8
sc_list[1]	30	22	29	27	NONE
sc_list[2]	46	42	NONE	NONE	NONE
sd_list[0]	31	23	29	27	8
sd_list[1]	30	22	28	26	NONE

Power Switch Monitor

The PSM is a user-level process that monitors scan points associated with the power switches and also monitors the scan points for the fan units. It also monitors the scan points for fuses on the 3B21D computer. The PSM provides an alternative to the IM/OM commands for removing and restoring units which have power switches. That is, a manual request to remove or restore the unit can be made at the power switch by setting the ROS/RST switch. Setting the ROS/RST switch to the ROS position will result in a manual request to remove the unit. Similarly, setting the ROS/RST switch to the RST position will result in a manual request to restore the unit.

The PSM is informed of changes in power switch states via the SCSDA. A group 1 power switch has three scan points and two distribute points which are organized into a logical SCSD group. The PSM uses a transition table to determine the appropriate action to be taken for each scan change.

A scan point change results in one or more of the following actions:

- Generates a remove/restore request and sends the request to MIRA.
- Operates the power switch lamps.
- Sends a message to the output spooler.

The PSM is used to monitor the power switches for various units on the 3B20D/3B21D computers. (Refer to Figure 16-12 for a diagram of a power switch.) The power switch units include the following:

- Control Unit (CU)
- Disk File Controller (DFC)
- Input/Output Processor (IOP)
- Moving Head Disk (MHD)
- Power Distribution Frame (PDF) – only on the 3B20D computer
- Digital Audio Tape (DAT) Unit – only on the 3B21D computer
- Fan Unit (FANALM).

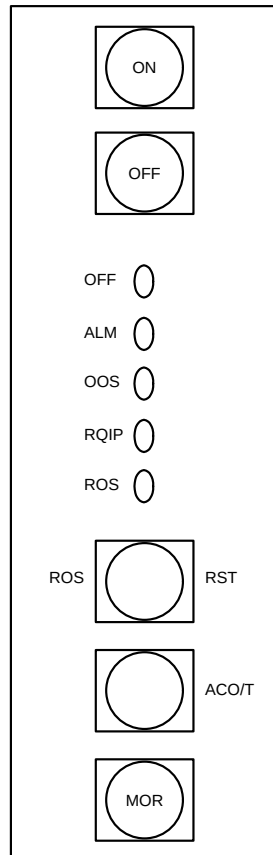


Figure 16-12. Power Switch

Most power switches have two scan points (X and Y) and two distribute points associated with them. On the 3B21D computer, the CU and the IOP 0 and 1 power switches have a third scan point called Z.

- The X scan point is set by the ROS/RST switch on the power switch.
- The Y scan point is set by power faults, such as blown fuses and power out of tolerance.
- The Z scan point is set by fuse alarms within the basic cabinet. It is used to identify one of the reasons for a loss of power on a unit (that is, a blown fuse).

The distribute points are connected to LEDs which are used to alert the craft of the status of a unit. One distribute point is connected to the RQIP light. It is turned on when a request for a remove or restore is issued via the ROS/RST switch. The other distribute point is connected to the OOS light. This point indicates whether the unit is OOS or not.

The PSM uses scan point transitions to determine the appropriate action to take. Table 16-4 has the scan point transitions for the 3B20D computer while Table 16-5 has the scan point transitions for the 3B21D computer. Figure 16-13 is a diagram of the scan point transitions listed in Table 16-4.

Tables 16-4 and 16-5 describe the action taken by the PSM when a scan point transition occurs. The particular action taken is summarized as follows.

fault	The maintenance personnel (via an output message) will be informed of a power fault on a unit. The alarm level will either be MINOR or MAJOR.
fault cleared	The maintenance personnel (via an output message) will be informed of a power fault cleared on a unit.
fuse	Action valid only on the 3B21D computer. The maintenance personnel (via an output message) will be informed of a fuse alarm within the basic or growth cabinets.
fuse cleared	Action valid only on the 3B21D computer. The maintenance personnel (via an output message) will be informed of a fuse alarm cleared within the basic or growth cabinets.
none	No scan point transitioned.
powerdown	The maintenance personnel (via an output message) will be informed of a loss of power to a unit.
powerup	The maintenance personnel (via an output message) will be informed of a restore of power to a unit.
remove (ROS)	The PSM will send a remove request to the MIRA.
restore (RST)	The PSM will send a restore request to the MIRA.

As part of the recovery of a powered down power switch unit, the PSM will perform the following actions when the power switch unit is powered up:

- Send a power restored message to the ROP to inform maintenance personnel.
- Send a restore request to the MIRA if the ROS/RST switch is set to the RST position. The restore request will be for the particular unit and, if one exists, its entire community.

For example, when a DFC is powered up with its ROS/RST switch set to the RST position and the ROS/RST switch of one of its MHDs is set to the ROS position, the PSM will inform maintenance personnel of the power being restored to the DFC via a ROP message. The PSM will also send a request to restore the DFC and its entire community to the MIRA. If all diagnostics pass, the DFC and its community will be restored to service, including the MHD whose ROS/RST switch was set to the ROS position.

Table 16-4. 3B20D Computer Scan Point Transitions and Actions

	Previous X	Previous Y	Current X	Current Y	Action Taken
[0]	0	0	0	0	none
[1]	0	0	0	1	fault
[2]	0	0	1	0	remove (ROS)
[3]	0	0	1	1	powerdown
[4]	0	1	0	0	fault cleared
[5]	0	1	0	1	none
[6]	0	1	1	0	fault cleared
[7]	0	1	1	1	powerdown
[8]	1	0	0	0	restore (RST)
[9]	1	0	0	1	fault
[10]	1	0	1	0	none
[11]	1	0	1	1	powerdown
[12]	1	1	0	0	powerup, restore
[13]	1	1	0	1	powerup, fault
[14]	1	1	1	0	powerup, remove
[15]	1	1	1	1	none

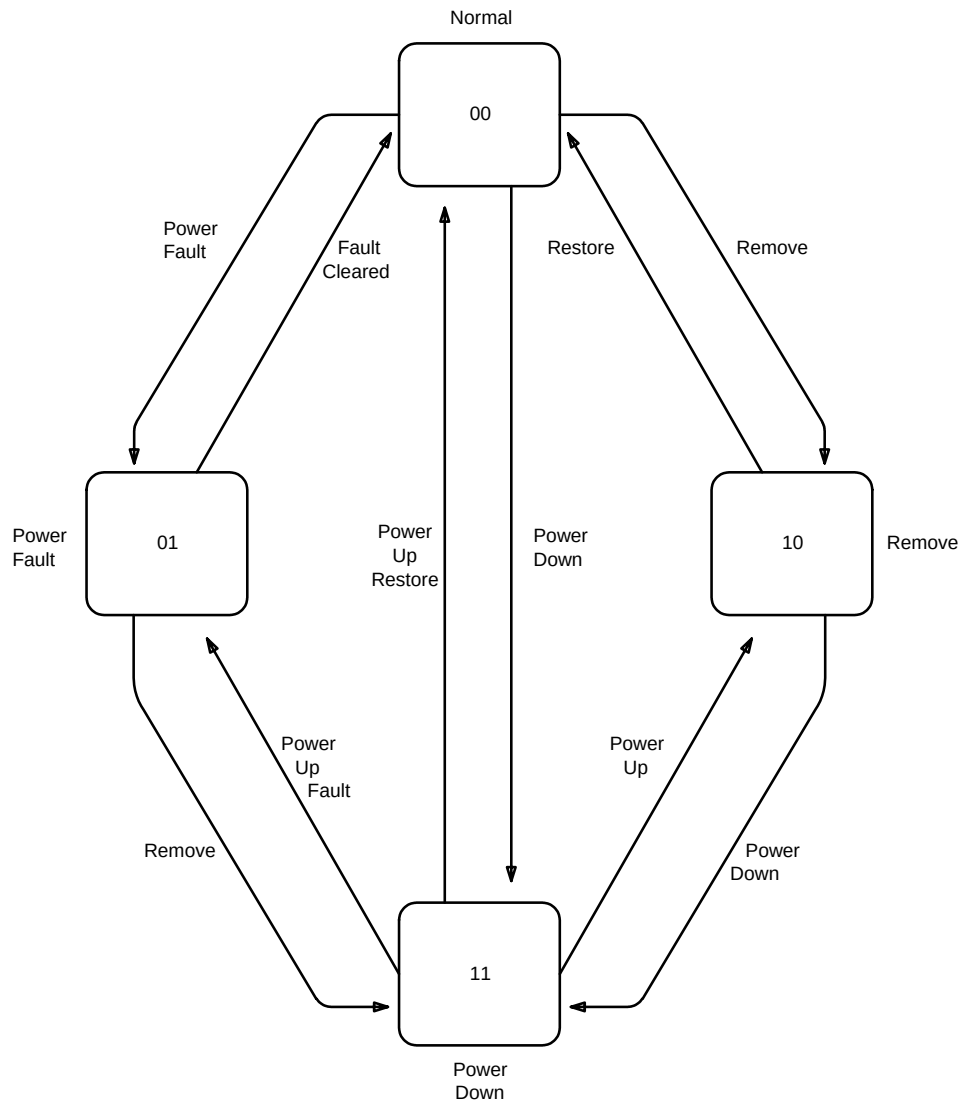


Figure 16-13. 3B20D Computer Scan Point Transition Diagram

Table 16-5. 3B21D Computer Scan Point Transitions and Actions

	Previous X	Previous Y	Previous Z	Current X	Current Y	Current Z	Action Taken
[0]	0	0	0	0	0	0	none
[1]	0	0	0	0	0	1	fuse
[2]	0	0	0	0	1	0	fault
[3]	0	0	0	0	1	1	fault, fuse
[4]	0	0	0	1	0	0	remove (ROS)
[5]	0	0	0	1	0	1	remove, fuse
[6]	0	0	0	1	1	0	powerdown
[7]	0	0	0	1	1	1	powerdown, fuse
[8]	0	0	1	0	0	0	fuse cleared
[9]	0	0	1	0	0	1	none
[10]	0	0	1	0	1	0	fault, fuse cleared
[11]	0	0	1	0	1	1	fault
[12]	0	0	1	1	0	0	remove, fuse cleared
[13]	0	0	1	1	0	1	remove
[14]	0	0	1	1	1	0	powerdown, fuse cleared
[15]	0	0	1	1	1	1	powerdown
[16]	0	1	0	0	0	0	fault cleared
[17]	0	1	0	0	0	1	fuse, fault cleared
[18]	0	1	0	0	1	0	none
[19]	0	1	0	0	1	1	fuse
[20]	0	1	0	1	0	0	fault cleared
[21]	0	1	0	1	0	1	fuse, fault cleared
[22]	0	1	0	1	1	0	powerdown
[23]	0	1	0	1	1	1	powerdown, fuse
[24]	0	1	1	0	0	0	fault cleared, fuse cleared
[25]	0	1	1	0	0	1	fault cleared
[26]	0	1	1	0	1	0	fuse cleared
[27]	0	1	1	0	1	1	none
[28]	0	1	1	1	0	0	fault cleared, fuse cleared
[29]	0	1	1	1	0	1	fault cleared
[30]	0	1	1	1	1	0	powerdown, fuse cleared
[31]	0	1	1	1	1	1	powerdown

Table 16-5. 3B21D Computer Scan Point Transitions and Actions (Contd)

	Previous X	Previous Y	Previous Z	Current X	Current Y	Current Z	Action Taken
[32]	1	0	0	0	0	0	restore (RST)
[33]	1	0	0	0	0	1	restore, fuse
[34]	1	0	0	0	1	0	fault
[35]	1	0	0	0	1	1	fault, fuse
[36]	1	0	0	1	0	0	none
[37]	1	0	0	1	0	1	fuse
[38]	1	0	0	1	1	0	powerdown
[39]	1	0	0	1	1	1	powerdown, fuse
[40]	1	0	1	0	0	0	restore, fuse cleared
[41]	1	0	1	0	0	1	restore
[42]	1	0	1	0	1	0	fault, fuse cleared
[43]	1	0	1	0	1	1	fault
[44]	1	0	1	1	0	0	fuse cleared
[45]	1	0	1	1	0	1	none
[46]	1	0	1	1	1	0	powerdown, fuse cleared
[47]	1	0	1	1	1	1	powerdown
[48]	1	1	0	0	0	0	powerup, restore
[49]	1	1	0	0	0	1	powerup, restore, fuse
[50]	1	1	0	0	1	0	powerup, fault
[51]	1	1	0	0	1	1	powerup, fault, fuse
[52]	1	1	0	1	0	0	powerup, remove
[53]	1	1	0	1	0	1	powerup, remove, fuse
[54]	1	1	0	1	1	0	none
[55]	1	1	0	1	1	1	fuse
[56]	1	1	1	0	0	0	powerup, restore, fuse cleared
[57]	1	1	1	0	0	1	powerup, restore
[58]	1	1	1	0	1	0	powerup, fault, fuse cleared
[59]	1	1	1	0	1	1	powerup, fault
[60]	1	1	1	1	0	0	powerup, remove, fuse cleared
[61]	1	1	1	1	0	1	powerup, remove
[62]	1	1	1	1	1	0	fuse cleared
[63]	1	1	1	1	1	1	none

PSM Interfaces

The PSM interprets scan point transitions of the power switch, removes units from service, restores units to service, and notifies the craft of these events by setting the appropriate LEDs. To accomplish this, the PSM interfaces with the following 3B20D/3B21D computer processes:

- SCSDA – The Scan Point and Signal Distributor Administrator is used to scan the power switch scan points and the fan unit scan points on the 3B20D/3B21D computers. It also controls the distribute points (LEDs) for the power switch and the fan unit.
- Craft Spooler – The Craft Spooler is used to inform craft via messages of powerdowns, powerups, and power faults.
- MIRA – The Maintenance Input Request Administrator is the mechanism by which requests are made to remove a unit from service and restore a unit to service.
- CONFIG – The Configuration Manager tracks the configuration of the system and reports to the PSM when any unit it monitors changes state from Active (ACT) to OOS and vice versa.
- ECD – The Equipment Configuration Database is used to obtain access to the *ucb* for a unit. The *ucb* contains the name, number, and state of the unit needed by the PSM.
- ULARP – The User Level Automatic Restart Process is the mechanism that starts the PSM process and restarts it, if necessary.

PSM Initialization

Initialization consists of checking for the existence of MIRA and the SCSDA and opening the ECD. If any of these operations fail, the PSM exits with the code UFATAL. ULARP will not attempt to restart the PSM if it encounters a UFATAL code.

Each logical group of SCSD scan points has a state associated with it. These states indicate what processes were last communicated with and what response is expected. The states are as follows:

- S_UNAVL – group unavailable
Entry to this state occurs when the PSM cannot communicate with a logical group for one of the reasons listed in Figure 16-10. This state will be updated to S_ACTV to generate autonomous reports.
- S_ACTV – group active
Entry to this state occurs after the SCSDA initialization messages have been received by the PSM. During initialization, group names and scan point states are saved and the appropriate distribute points are set. When the PSM is in the active state, it is waiting for a request.

- **S_DIR** – waiting for acknowledgment of a directed scan request
Entry to this state occurs when a scan point on a power switch changes state. The SCSDA detects these scan point transitions and sends them to the PSM. The PSM saves the old and new states of the scan points and sends a directed scan to the SCSDA. The reason a directed scan is sent is that under certain conditions two scan points can change simultaneously and the scanner may report only one of them. By sending a directed scan request, the PSM will be informed of all scan point changes.
- **S_MIRA** – waiting for an acknowledgment from MIRA
Entry to this state occurs when a remove or restore request has been sent to the PSM. After the request has been sent and an acknowledgment message has been received from MIRA, the RQIP LED on the power switch is lit by the PSM to indicate to the craft that the request is in progress. Transition to the active state (**S_ACTV**) occurs when a success or failure message has been received from MIRA. If the request failed, the RQIP LED will flash for 8 seconds to indicate to the craft that the request failed.
- **S_MIRA_AUTO** – waiting for an acknowledgment from MIRA while processing an autonomous report.
Entry to this state occurs when the PSM receives an autonomous scan report while in the **S_MIRA** state. This state allows autonomous reports to be processed while an acknowledgment message from MIRA is waiting to be received.

PSM Messages

The PSM sends and receives messages. The PSM sends the following messages:

- Remove requests to the MIRA.
- Restore requests to the MIRA.
- Directed scan requests to the administrator.
- Distribute operation requests to the administrator.

There are three classes of messages that the PSM receives:

- SCSDA messages are of two types:
 - INIT messages tell the PSM the initial states of the power switch scan points. This type of message is received only after the PSM has been brought up.
 - Autonomous reports inform the PSM that a scan point for a power switch has changed state.

- Acknowledgment messages are received from the SCSDA and MIRA. These messages are as follows:
 - SCSDA Acknowledgment Messages – The SCSDA sends acknowledgment messages to the PSM for distribute requests and directed scan requests. Acknowledgment messages for distribute requests indicate whether the request failed or succeeded. Acknowledgment messages for directed scan requests contain the status information for the scan points in question.
 - MIRA Acknowledgment Messages – MIRA sends acknowledgment messages to the PSM to indicate that a remove or restore request has been initiated. This message would then cause the PSM to light the RQIP LED.
- CONFIG messages are messages sent when any unit is actually removed or restored. The OOS LED is set On or Off to reflect the current status of that unit. The PSM ignores these messages for units that it is not monitoring.

PSM Message Structures

Communication with other processes is done via messages. The following is a description of the message structures used by the PSM to communicate with other processes:

- **SCSDA Message Structure.**

The SCSDA has a message structure called *adh_body*. Elements of this structure include information about the logical group name, RID, group state, and the scan point states.
- **MIRA Structure.**

MIRA has a message structure called *dgnrpy*. This message allows the PSM to send remove and restore requests for a specific unit.
- **CONFIG Structure.**

CONFIG sends a message to the PSM when any unit is taken out of service or restored to service. (PSM ignores units which it is not monitoring.) The fields used are the RID of the unit and the status of the OOS LED.
- **PSM Data Structure.**

The PSM keeps track of all of its power switches in a static data area called *group*, which is an array of structures overlaid by the *psmstat* structure definition. This data structure is used to tie together all the requests, messages, and changes of state that occur at the power switches. Figure 16-14 shows these data structures.

```

struct psmstat
{
    char          groupnm[9];    /* name of group */
    char          dupid;         /* duplex ID of this group */
    char          state;         /* state this group is in */
    char          fill1;
    RID           rid;           /* RID of unit scanned by this group */
    RID           scsd_rid;       /* RID of SCSDC unit monitoring this group */
    int           scsd_unit;      /* SCSDC unit number */
    int           iop_unit;       /* IOP unit number */
    struct eqid   eqid;          /* name of the unit being scanned */
    int           pt_count;       /* point in this group */
    short         to_key;         /* time out key */
    short         sequence;       /* book keeping to know what operation is doing */
    union
    {
        struct
        {
            b_switch sc_fill:    28;    /* filler */
            b_switch sc_oldx:    1;    /* old value of sc pt x */
            b_switch sc_oldy:    1;    /* old value of sc pt y */
            b_switch sc_newx:    1;    /* new value of sc pt x */
            b_switch sc_newy:    1;    /* new value of sc pt y */
        }
        sc_xy;
        struct
        {
            b_switch sc_fill:    26;    /* filler */
            b_switch sc_oldx:    1;    /* old value of sc pt x */
            b_switch sc_oldy:    1;    /* old value of sc pt y */
            b_switch sc_oldz:    1;    /* old value of sc pt z */
            b_switch sc_newx:    1;    /* new value of sc pt x */
            b_switch sc_newy:    1;    /* new value of sc pt y */
            b_switch sc_newz:    1;    /* new value of sc pt z */
        } sc_xyz;
        struct
        {
            b_switch sc_fill:    24;    /* filler */
            b_switch oldinhxy:   2;    /* inhibit states */
            b_switch newinhxy:   2;    /* new inhibit state */
            b_switch oldscxy:    2;    /* old scan states */
            b_switch newscxy:    2;    /* new scan states */
        } sc_st1;
        struct
        {
            b_switch sc_fill:    20;    /* filler */
            b_switch oldinhxyz:   3;    /* inhibit states */
            b_switch newinhxyz:   3;    /* new inhibit state */
            b_switch oldscxyz:    3;    /* old scan states */
            b_switch newscxyz:    3;    /* new scan states */
        }
        sc_st1_21;
    }
}

```

Figure 16-14. PSM Data Structure (Sheet 1 of 2)

```

    struct
    {
        b_switch sc_fill:    24;    /* filler */
        b_switch inh_trans;  4;    /* old and new inhibit transitions */
        b_switch sc_trans:   4;    /* old and new scan transitions */
    }
    sc_st2;
    struct
    {
        b_switch sc_fill:    20;    /* filler */
        b_switch inh_trans;  6;    /* old and new inhibit transitions */
        b_switch sc_trans:   6;    /* old and new scan transitions */
    }
    sc_st2_21;
    struct
    {
        b_switch sc_fill:    24;    /* filler */
        b_switch inholdx:    1;
        b_switch inholdy:    1;
        b_switch inhnewx:    1;
        b_switch inhnewy:    1;
        b_switch scoldx:     1;
        b_switch scoldy:     1;
        b_switch scnewx:     1;
        b_switch scnewy:     1;
    }
    sc_st3;
    struct
    {
        b_switch sc_fill:    20;    /* filler */
        b_switch inholdx:    1;
        b_switch inholdy:    1;
        b_switch inholdz:    1;
        b_switch inhnewx:    1;
        b_switch inhnewy:    1;
        b_switch inhnewz:    1;
        b_switch scoldx:     1;
        b_switch scoldy:     1;
        b_switch scoldz:     1;
        b_switch scnewx:     1;
        b_switch scnewy:     1;
        b_switch scnewz:     1;
    }
    sc_st3_21;
    int    action;    /* action dictated by sc change */
}
sc_state;
};
struct psmstat group [MAXGRP];    /* actual data used by power switch monitor */

```

Figure 16-14. PSM Data Structure (Sheet 2 of 2)

SCSD System and Alarm Requirements and Assignments

This section consolidates the scan and signal distributor and alarm requirements for the 3B20D and 3B21D computers. The scan and signal distributor points for the minimum configuration are fixed and preassigned in order to populate a *UNIX* RTR operating system ECD.

- The 3B20D computer has fixed scan and signal distributor point pin assignments for two control units (CU 0 and CU 1), IOP 0, IOP 1, DFC 0, DFC 1, MHD 0, MHD 1, MTTY, ROP, cooling units, software generated alarms, and the power distribution frame.
- The 3B21D computer has fixed scan and signal distributor point pin assignments for two control units (CU 0 and CU 1), IOPs 0-3, DFC 0, DFC 1, MHDs 0-5, MTTY, ROP, cooling unit, software generated alarms, and 1 DAT unit.

General Requirements

The following are basic requirements dictated by the 3B20D/3B21D computer operating system and office engineering constraints:

- SCSDCs should not be duplicated.
- SCSD points should not be duplicated.
- The following scan points are monitored from the power switch:
 - Scan point X – request out of service
 - Scan point Y – major or minor power fault
 - Scan point Z – fuse alarm (3B21D computer only).
- Two available relay contacts in the power distribution frame are used for scan points SAMJ (major) and SBMN (minor).
- Two signal distribute points are dedicated to each power switch (group 1 only).
 - SD point RQIP – request in progress
 - SD point OOS – out of service.
- Signal distribute points are reserved for software alarm functions, some of which may be specified by the applications. The following are signal distribute points:
 - Crit – critical alarm
 - Mj – major alarm
 - Mn – minor alarm
 - Sys – software triggered alarm (3B20D computer only)
 - aret – alarm retire.

SCSD Point Assignments

On the 3B20D computer the scan and signal distributor circuit pack, UN33, contains 48 scan points and 32 signal distribute points. The number of SCSD points required on the two base UN33 circuit packs for a basic 3B20D computer complex consisting of two CUs, two IOPs, two DFCs, two MHDs, one MTTY, one ROP, cooling units, and the power distribution frame is 28 scan points and 20 Signal Distributor (SD) points. In addition, ten SD points (five per base UN33 circuit pack) are assigned for the software generated alarms.

On the 3B21D computer the scan and signal distributor circuit pack, UN33/UN933, contains 48 scan points and 32 signal distribute points. The number of SCSD points required on the two base UN33/UN933 circuit packs for a basic 3B21D computer complex consisting of two CUs, four IOPs, two DFCs, six MHDs, one MTTY, one ROP, a cooling unit, and one DAT unit is 42 scan points and 32 SD points. In addition, eight SD points (four per base UN33/UN933 circuit pack) are assigned for the software generated alarms.

The following rules are used for assigning the scan and signal distributor points:

- Minimum of one SCSD circuit pack per IOP 0 and IOP 1.
- Duplicate SCSD points (if required) must be assigned the same points on the duplicate controllers.
- Scan groups, that is SCSD points from one power switch, must be assigned to the same SCSDC.
- SCSDC 0 is equipped in IOP 0, PC slot 02, and monitors the 3B20D/3B21D computer equipment with even numbers; for example, CU 0, DFC 0, MHD 0, MHD 2, and so forth. SCSDC 1 is equipped in IOP 1, PC slot 02, and monitors the 3B20D/3B21D computer equipment with odd numbers. The exceptions to this rule are IOP 0 and IOP 1 which are monitored from SCSDC 1 and SCSDC 0, respectively.
- Applications may assign their equipment to the remaining SCSD points and/or add additional SCSD packs, if needed.

Table 16-6 lists the scan and signal distributor points on SCSD controller 0 and SCSD controller 1 (that is, SCSDC 0 and 1) of the 3B20D computer. Tables 16-7 and 16-8 list the 3B21D computer scan and signal distributor points on SCSDC 0 and 1 for both large and small configurations. The tables show what points are assigned to a particular unit and their scan and/or signal distributor functions. SC and SD point assignments for an application are shown in *italic* type in these tables. The *italicized* table entries are NOT part of the 42 SC and 40 SD points required for the basic 3B21D computer complex.

Table 16-6. 3B20D Computer Scan Point and Signal Distributor Point Assignments

SCSD Point	SCSDC 0			SCSDC 1		
	Unit	SC Function	SD Function	Unit	SC Function	SD Function
00		*	*	IOP 0	Power	OOS
01		*	*	IOP 0	ROS/RST	RQIP
02	CU 0	Power	OOS		*	*
03	CU 0	ROS/RST	RQIP		*	*
04	MHD 4	Power	OOS	MHD 5	Power	OOS
05	MHD 4	ROS/RST	RQIP	MHD 5	ROS/RST	RQIP
06	IOP 2	Power	OOS		*	*
07	IOP 2	ROS/RST	RQIP		*	*
08	IOP 1	Power	OOS		*	*
09	IOP 1	ROS/RST	RQIP		*	*
10		*	*	CU 1	Power	OOS
11	MTTY	PTSW	*	CU 1	ROS/RST	RQIP
12	Alarm	*	Critical	Alarm	*	Critical
13	Alarm	*	Major	Alarm	*	Major
14		*	*	IOP 3	Power	OOS
15	ROP 0	PTSW	*	IOP 3	ROS/RST	RQIP
16	DFC 0	Power	OOS		*	*
17	DFC 0	ROS/RST	RQIP		*	*
18	MHD 0	Power	OOS		*	*
19	MHD 0	ROS/RST	RQIP		*	*
20	Alarm	*	Minor	Alarm	*	Minor
21	Alarm	*	System	Alarm	*	System
22	MHD 2	Power	OOS	MHD 3	Power	OOS
23	MHD 2	ROS/RST	RQIP	MHD 3	ROS/RST	RQIP
24		*	*	DFC 1	Power	OOS
25		*	*	DFC 1	ROS/RST	RQIP
26		*	*	MHD 1	Power	OOS
27		*	*	MHD 1	ROS/RST	RQIP
28	Alarm	*	Retire	Alarm	*	Retire
29		*	*		*	*
30	FANALM 0	*	*	FANALM 1	*	*
31	FANALM 0	*	*	FANALM 1	*	*
32		*	*		*	*
33		*	*		*	*
34		*	*		*	*
35		*	*		*	*
36	MTTY	PTSW	*		*	*
37	MTTY	PTSW	*		*	*
38	ROP 0	PTSW	*		*	*
39	ROP 0	PTSW	*		*	*
40		*	*		*	*
41		*	*		*	*
42		*	*		*	*
43		*	*		*	*
44	PDF 0	*	*		*	*
45	PDF 0	*	*		*	*
46		*	*		*	*
47		*	*		*	*

* Reserved or inaccessible.

Table 16-7. 3B21D Computer Scan Point and Signal Distributor Point Assignments — Large Office

SC/SD Point	SCSDC 0			SCSDC 1		
	Unit	SC Function	SD Function	Unit	SC Function	SD Function
00	3B Growth	FAN	Reset	3B Growth	FAN	Reset
01		—	—		—	—
02		*	*		*	*
03		*	*		*	*
04		*	*		*	*
05		*	*		*	*
06		*	*		*	*
07		*	*		*	*
08	3B FAN	FANALM 0	RESET	3B FAN	FANALM 1	RESET
09		—	—		—	—
10	Misc FAN	FANALM 0	Reset	Misc FAN	FANALM 1	Reset
11		—	—		—	—
12	Alarm	—	Critical	Alarm	—	Critical
13	Alarm	—	Major	Alarm	—	Major
14	Alarm	—	Minor	Alarm	—	Minor
15	Alarm	—	Retire	Alarm	—	Retire
16	Tape-1	ROS/RST	OOS		+	+
17	Tape-1	Power	RQIP		+	+
18	MHD 4	ROS/RST	OOS	MHD 5	ROS/RST	OOS
19	MHD 4	Power	RQIP	MHD 5	Power	RQIP
20	IOP 2	Power	OOS	IOP 3	Power	OOS
21	IOP 2	ROS/RST	RQIP	IOP 3	ROS/RST	RQIP
22	IOP 1	Power	OOS	IOP 0	Power	OOS
23	IOP 1	ROS/RST	RQIP	IOP 0	ROS/RST	RQIP
24	MHD 2	ROS/RST	OOS	MHD 3	ROS/RST	OOS
25	MHD 2	Power	RQIP	MHD 3	Power	RQIP
26	MHD 0	ROS/RST	OOS	MHD 1	ROS/RST	OOS
27	MHD 0	Power	RQIP	MHD 1	Power	RQIP
28	DFC 0	ROS/RST	OOS	DFC 1	ROS/RST	OOS
29	DFC 0	Power	RQIP	DFC 1	Power	RQIP
30	CU 0	Power	OOS	CU 1	Power	OOS
31	CU 0	ROS/RST	RQIP	CU 1	ROS/RST	RQIP
32		—	—		—	—
33	3B Growth	FA	—	3B Growth	FA	—
34		*	—		*	—
35		*	—		*	—
36		*	—		*	—
37		*	—		*	—
38	Misc FA 1	FA1	—	Misc FA 1	FA1	—
39	Misc FA 0	FA0	—	Misc FA 0	FA0	—
40		+	—	ROP	PTSW	—
41		+	—	ROP	PTSW	—
42	IOP 1	Fuse	—	IOP 0	Fuse	—
43		—	—	ROP	PTSW	—
44		+	—	MTTY	PTSW	—
45		+	—	MTTY	PTSW	—
46	CU 0	Fuse	—	CU 1	Fuse	—
47		—	—	MTTY	PTSW	—

* Reserved — Inaccessible + Not assigned

Table 16-8. 3B21D Computer Scan Point and Signal Distributor Point Assignments — Small Office

SC/SD Point	SCSDC 0			SCSDC 1		
	Unit	SC Function	SD Function	Unit	SC Function	SD Function
00	CMU 0	Power	OOS	CMU 0	Power	OOS
01	CMU 0	ROS/RST	RQIP	CMU 0	ROS/RST	RQIP
02	CMU 0	W??	DGN	CMU 0	W??	DGN
03	CMU 0	V??		CMU 0	V??	
03	NCLK 0		OOS	NCLK 0		OOS
04	CMU 1	Power	OOS	CMU 1	Power	OOS
05	CMU 1	ROS/RST	RQIP	CMU 1	ROS/RST	RQIP
06	CMU 1	W??	DGN	CMU 1	W??	DGN
07	CMU 1	V??		CMU 1	V??	
07	NCLK 1		OOS	NCLK 1		OOS
08	3B FAN	FANALM 0	RESET	3B FAN	FANALM 1	RESET
09		—	—		—	—
10	Misc FAN	FANALM 0	RST	Misc FAN	FANALM 1	RST
11		—	—		—	—
12	Alarm	—	Critical	Alarm	—	Critical
13	Alarm	—	Major	Alarm	—	Major
14	Alarm	—	Minor	Alarm	—	Minor
15	Alarm	—	Retire	Alarm	—	Retire
16		+	+	TAPE-1	ROS/RST	OOS
17		+	+	TAPE-1	Power	RQIP
18	MHD 4	ROS/RST	OOS	MHD 5	ROS/RST	OOS
19	MHD 4	Power	RQIP	MHD 5	Power	RQIP
20	IOP 2	Power	OOS	IOP 3	Power	OOS
21	IOP 2	ROS/RST	RQIP	IOP 3	ROS/RST	RQIP
22	IOP 1	Power	OOS	IOP 0	Power	OOS
23	IOP 1	ROS/RST	RQIP	IOP 0	ROS/RST	RQIP
24	MHD 2	ROS/RST	OOS	MHD 3	ROS/RST	OOS
25	MHD 2	Power	RQIP	MHD 3	Power	RQIP
26	MHD 0	ROS/RST	OOS	MHD 1	ROS/RST	OOS
27	MHD 0	Power	RQIP	MHD 1	Power	RQIP
28	DFC 0	ROS/RST	OOS	DFC 1	ROS/RST	OOS
29	DFC 0	Power	RQIP	DFC 1	Power	RQIP
30	CU 0	Power	OOS	CU 1	Power	OOS
31	CU 0	ROS/RST	RQIP	CU 1	ROS/RST	RQIP
32	NCLK 0	Power		NCLK 0	Power	
33	NCLK 0	ROS/RST		NCLK 0	ROS/RST	
34	NCLK 1	Power		NCLK 1	Power	
35	NCLK 1	ROS/RST		NCLK 1	ROS/RST	
36	ESM Alarm	Critical		ESM Alarm	Critical	
37	ESM Power	Major		ESM Power	Major	
38	Misc FA 1	FA1		Misc FA 1	FA1	
39	Misc FA 0	FA0		Misc FA 0	FA0	
40		+		ROP	PTSW	
41		+		ROP	PTSW	
42	IOP 1	Fuse		IOP 0	Fuse	
43		—		ROP	PTSW	
44		+		MTTY	PTSW	
45		+		MTTY	PTSW	
46	CU 0	Fuse		CU 1	Fuse	
47		—		MTTY	PTSW	

— Inaccessible + Not assigned

Alarms

The hardware generated alarms, major and minor, are triggered by contact closures on the power switch. Cabling can be made such that all the major and minor contacts can be connected in parallel to the office alarm circuit. Documentation for interconnections is shown in SD-4C054-01 and ED-4C208.

Fan Alarms

The fan alarms are designed to work for the fan circuitry described in ED-4C387-30, *Common Systems Specification for 3B20D Computer Processor Cooling Unit*, for the 3B20D computer and in SD-5D168-01, *6 FAN BIDIRECTIONAL FAN UNIT*, for the 3B21D computer. The PSM provides a fan alarm scheme for the 3B20D/3B21D computers that encompasses the following features:

- For all releases except *UNIX RTR Operating System Release 6.4*, a major office alarm is sounded in the event of multiple fan failures and a minor office alarm is sounded in the event of a single fan failure. For Release 6.4, a critical office alarm is sounded in the event of multiple fan failures and a major office alarm is sounded in the event of a single fan failure. In all releases, the use of four scan points per duplex machine (two scan points per alarm) is required. However, on the 3B21D computer, a critical office alarm is sounded for all fan failures. In all 3B21D computer releases, the use of a single scan point per duplex machine is required.
- The scan points must latch so as not to cause the appearance of transients to the software.
- The alarm is triggered via software. No hardware connection to the office alarm grid is provided.
- The ability to retire the fan alarm indicators from the Maintenance Teletypewriter (MTTY) via the **PF-4** key which retires all alarms.
- The ability to reset the fan alarm via the **ORD** command.
- The ability to inhibit the fan alarms for 30 minutes from the MTTY. This allows maintenance work on fans without triggering alarms.
- The ability to "ALLOW" the fan alarms before "INHIBIT" times out. A warning message will be printed on the MTTY if the "INHIBIT" times out before the "ALLOW" command is issued.
- A new alarm is triggered every 30 minutes after the first alarm if no action is taken and the failure persists.
- For *UNIX RTR Operating System Release 21.13* and later, a visual indication of the status of the CU fan alarm scan point(s) will be provided on the Common Processor Display Page (CPDP). This visual indication can be turned off via the *ularp* record for the PSM. Field 14 (proc_args) can be set to "NOFAN" to turn this capability off.

Detailed Design Description

The SCSDA detects scan point changes and informs the PSM. The PSM sets the appropriate office alarms via distribute points.

Audible alarms and output messages are generated via software determined by the scan point transitions set by the fan circuitry. The Priority of Action (POA) field on the spooler message is set according to scan point state transitions. This POA is then interpolated by the alarm process which, in turn, operates the audible alarms via SCSDA messages. LED indicators on the fan unit are not controlled by software. The fan circuitry lights these LEDs to identify the fan that is alarming in the fan tray. The alarm state on the fan unit can be retired remotely via a craft command. This will only retire the alarm on the unit itself and will not clear the office audible alarms. (The alarms are silenced temporarily.) After a 10-second circuit delay, if the failure persists, the alarm will be triggered by the fan unit again.

Power Switch Monitor

If a fan failure occurs on a CU frame, the fan alarm circuitry will cause scan points to be set. The SCSDA detects these transitions and notifies the PSM. The PSM then starts a timer to reset the fan alarms if no action is taken by the craft and the fan failure still persists after 30 minutes. The PSM will also print out a fan alarm message (REPT FAN SINGLE FAILURE CU 0-1 or REPT FAN MULTIPLE FAILURE CU 0-1 or REPT FAN ALARM CU BASIC CABINET). Fan alarms will be reset via a distribute message sent to the SCSDA.

If the craft within 30 minutes decides to repair the fans, they first inhibit the fan alarm scan points using the **INH:SCSD** command via the MTTY. This is done so that the SCSDA will ignore scan point transitions for the fan alarms in that frame and will not send the PSM scan point information for the fan alarms.

The PSM will be notified of an inhibit via a message from the **INH** command and will start a 30-minute inhibit timer. This inhibit timer prevents the craft from inadvertently suppressing the fan alarms for long periods of time.

If the fans are repaired before the 30-minute inhibit timer expires, the craft would then remove the inhibit command with the **ALW:SCSD** command. This allows the SCSDA again to recognize scan point transitions for the fan alarms and informs the PSM.

If the craft does not fix the fans and the 30-minute inhibit timer expires, the PSM will remove the inhibit on the fan alarm scan points via an **ALW** type message, which is sent to the SCSDA. The PSM will then print a message alerting the craft that the scan points are no longer inhibited (REPT FAN INHIBIT CLEARED CU 0-1 or REPT FAN INHIBIT CLEARED CU BASIC CABINET) and cancel the 30-minute reset timer. Since there has been no change in the fan alarm scan points, the alarms will be reset via a distribute message sent to the SCSDA.

For Releases 21.13 and later, the CPDP will have CU fan alarm box(es) to provide a visual status of the CU fan alarm scan point(s). All CU fan failures will result in the appropriate CU fan alarm box to be highlighted red. All inhibits of the CU fan alarm scan point(s) will result in the appropriate CU fan alarm box to be cleared with a text string of "INH" in it. Normal CU fan operation will result in the appropriate CU fan alarm box to be cleared.

In order for the PSM to monitor scan points under a particular group name, the groups are identified by their logical group prefix. The logical group names are prefixed with either FANA for a fan alarm group or PDF for a Power Distribution Frame group.

Fan Alarms and PSM Operation Features

The fan alarms and the PSM have the following operating features:

- **Basic Operation.**

The PSM waits for autonomous reports of scan point transitions which it monitors. The action taken on this report is determined by a scan point transition table. (The transition table consists of the various combinations of the new and old saved scan point states.)

The PSM monitors scan points for the fan alarms and ensures that the fan scan points are not inhibited for longer than 30 minutes without craft intervention.

The inhibit timer and the requirement that a new alarm be generated every 30 seconds while a failure persists requires that a timing chain be implemented in the PSM.

- **Internal Timing Chain.**

A user process has only a single timer that it can request. A timing chain is maintained because more than one logical group could be alarmed.

- **PSM Interface.**

The functions that the PSM interfaces to the timing chain are *reqtotsk* and *deltoreq*. These functions are used to request a time-out entry on the chain and to cancel a time-out request.

- **Timing Chain Implementation.**

The timing chain consists of two queues, an in-progress and an available queue.

- The in-progress queue is a double linked list for ease of deletion.
- The available queue is a singly linked list as entries are only taken off the top of this chain.

The timing chain is ordered in increasing time-out value, that is, earliest first. The functions that maintain the timing chain are *inittime*, *addtask*, *schtask*, *deltask*, and *time_ent*.

The time-out value is posted via the **alarm()** system call. The entry into PSM by the system is defined by the **signal()** system call. Because the time-out event is asynchronous, an interruption while manipulating the timing queue could leave the queues in an inconsistent state; therefore, the system time-out will be canceled before manipulating the timing queue. These functions on their return reissues the time-out request.

- *inittime* is used to initialize the timing chain.
- *addtask* is called by *reqtotsk*. (*reqtotsk* gets an entry to put on the timing chain.) *addtask* will put the time-out request on the chain. *schtask* is then called to post the head time-out task.
- *schtask* will use the system call **alarm()** to request the system time-out value.
- *deltask* will remove an entry from the list and place it back on the available queue.
- *time_ent* is the entry called by the system when a time-out has occurred. The timing chain is traversed and all expired tasks will be serviced. If the timing chain is not empty, *schtask* is called to request the next time-out event.

▪ **Autonomous Report.**

The following outlines the autonomous report:

- *gotauto* will determine if the transition is for a fan scan point and call *hld_fanalr* (via FANA prefix on a logical group name).
- *hld_fanalr* will determine if the transition is for an inhibit, allow, or an auto report. An inhibit transition table using the old and new inhibit states will determine the action to take.
- If the report is for an inhibit, timers are canceled, a 30-minute timer with a delayed time-out function of *send_alw* is requested, and the group is set to S_INH.
- If the report is for an allow, and no inhibits are present, determine if the group state is either S_ALW or S_INH. If it is either one of these states, return. The inhibit time-out will do any necessary processing. Otherwise, set the group state to S_INH and set an inhibit delay time-out function of *send_alw*. If no other inhibits are present, cancel timers and call *reset_fan*.
- If this is an autonomous report test and if the group state is either S_DIR or S_DIS, this auto report was generated by a PSM operation or PSM has already started processing by an earlier report. In either case, this auto report is ignored. If the report is to be processed — cancel timers. (A directed scan is issued, and the group state is set to S_DIR.)

- **Directed Scan Request.**

The *gotdir* request will call *fanalarm* if the operation is for a fan scan request. *fanalarm* will determine, using a scan state transition table, the output message to alert craft of the transition that has taken place. If an alarm was generated, a 30-minute timer is set with the time-out function *reset_fan*. The group state is set to S_ACTV.

- **Allow Fan Scan Points.**

- The function *send_alw* is entered on an inhibit time-out.
- *send_alw* will send an allow message to the SCSDA. In addition, it will output a message alerting craft that the inhibits are being removed and setting the group state to S_ALW.
- The acknowledgment will be handled by the function *recv_alw*. If the allow operation did not succeed, *err_proc* is called and will output an alarmed message indicating the error. The alarm severity is determined by the saved state of the fan scan points. If the group state is still S_ALW, a 30-minute timer is requested with the delay function *send_alw*. Otherwise, if the operation succeeded, output a message to the ROP indicating the inhibits are cleared and returned. Autonomous processing will complete the operation.

- **Reset Fan Alarms.**

- The *reset_fan* function is entered (either manually or automatically) on an alarm time-out or an allow completion.
- The *reset_fan* function will issue a distribute message to the SCSDA. This will pulse the distribute points associated with this group to retire the fan alarm. The group state is set to S_DIS.
- On the acknowledgment of the distribute operation, *gotdist* will determine if this is for a fan alarm operation. If the operation failed, *oper_fail* is called. If the group state is still S_DIS, request a 30-minute timer with a delayed function of *reset_fan*. Otherwise, a one second time-out with a delayed function of *ignor_rest* is set. This delay is used to ignore the transitions of resetting the fan scan points.
- *ignor_rest* will set a 15-second time-out with the delayed function *send_dir*. This delay is set to do a directed scan if another alarm is not generated. Otherwise, if an alarm is regenerated, this timer will be canceled. The group state is set to S_ACT to enable processing of auto reports. If a timer could not be acquired, *dir_fan* will be called.

- **External Errors.**

Two error conditions could occur:

- The SCSDC could be brought back into service.
- A chattering-fan alarm-scan point could be detected.

In the case of a chattering-scan point, SCSDA will send an AD_INHIBIT message to the PSM. This message is treated by the PSM as if an inhibit were placed on a fan scan point and *hld_fanalr* is called. The group state when the SCSDC is brought back into service determines what actions will take place:

- S_INH – The AD_INIT message, will be ignored. Upon reinitialization, the inhibits are replaced. Although the SCSDC could have been removed and restored, it has no affect on this operation.
- S_ALW or S_DIS – Timers will be canceled, and the requested delayed operation is executed.
- S_DIR, S_UNAVL, or S_ACTV – The group state is set to S_ACTV to process this report as an auto report.

- **Internal Errors.**

The function *err_proc* is called to process errors from a request by PSM. A message is printed to the ROP indicating the failure along with the logical group name and its current state. If the error is from the SCSDA, reporting the SCSDC is OOS, the group state will not be changed. An example is the ER_UNAVL error or any error such that the SCSDA could not complete because of errors or circumstances it encountered. The group state will be changed to S_UNAVL for errors that were generated because of an invalid request; for example, ER_PTNUM.

SCSD Commands

Access to Scan and Distribute Points

The SCSD commands allow access to scan and distribute points from a craft terminal. These commands enable the craft to read scan point states, set distribute points and to enable, disable, or read inhibits on scan points. The address of a scan or distribute point can be given in logical form which consists of a logical group name and a point number within that group. When a logical group is duplexed, a dupid of 0 or 1 is needed to distinguish the two duplex groups from each other. Any point can also be specified by a unique physical address consisting of an SCSDC number and a point number. This allows the craft to access scan and distribute points which have not yet been assigned to logical groups.

When an SCSD command is input from a craft terminal, the craft shell parses the command and performs some error checking. The correct SCSD command program is then searched for and executed. With information passed to the SCSD command program from the shell, the SCSD command program is able to carry out the action requested from the craft terminal. The SCSD command program communicates with the SCSDA using the client process message interface.

The output command (**OP:SCSD**) makes it possible to read scan points individually, by logical groups, or to read all scan points for an SCSDC at once. Refer to 254-303-111, *PDS Output Messages Manual*, or 254-303-113, *MML Output Messages Manual*, for more information on the message parameters. The inhibit status of each scan point is also displayed by the **OP** command. When displaying the status of a duplexed logical group, information is given on both the dupid 0 and the dupid 1 group.

The inhibit command (**INH:SCSD**) makes it possible for the craft to inhibit a scan point. Refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*, for information on the parameters of the **INH:SCSD** input message. The allow command (**ALW:SCSD**) makes it possible for the craft to disable the inhibit. This message can also be found in 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*. Both the **INH** and **ALW** commands can use physical or logical addressing.

Using the order command (**ORD:SCSD**), the craft can perform a distribute operation on a distribute point. Refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*, for information on the parameters of the **ORD:SCSD** input message. There are four possible distribute operations: set, clear, flash for 8 seconds, and flash and clear continuously. If the distribute point is in a duplexed logical group, a dupid can be given to specify which distribute point of the duplexed pair is operated on.

This is useful for separately testing each distribute point in a duplex pair. If a dupid is not given for a flash operation, then the SCSDA will choose one point of the duplex pair and do the flash operation on that point only. If a dupid is not given for a set or clear operation, then the operation will always be done in duplex over both points of the pair.

The reset fan command (**CLR:FANALM**) will set and clear the fan alarm distribute points. (**RESET:FANALM** is used in Generic 2.) This command is useful for testing whether a fan alarm still exists or not. Refer to 254-303-110, *PDS Input Messages Manual*, and 254-303-112, *MML Input Messages Manual*, for parameter information on the input messages.

Growth and Degrowth Procedures

This section describes the procedures for growing and degrowing UN33/UN933 boards and the associated building and power equipment alarm scan points. This section also contains procedures for adding and deleting logical groups on the UN33/UN933 board. Each alarm scan point is a hardware sensor probe on the UN33/UN933 board that detects an alarm condition. The UN33/UN933 boards are installed in slots of the IOP. The growth and degrowth procedures are step-by-step instructions for modifying the ECD via the Recent Change and Verify (RC/V) subsystem.

There are two key RC/V forms used in the procedures: the *ucb* and the *scsdbody* form. The goal of the procedures is to give the craft the capability to configure the building and power equipment alarms more closely to the actual states of the equipment on the switch. This should eliminate the case in which the ECD shows a nonexistent building and/or power alarm unit as equipped. When such a condition occurs, a number of error messages are produced on the ROP which might be confusing to the craft.

Preparation and Requirements for Growth and Degrowth Procedures

It is essential that the information about the growth/degrowth unit be made available prior to the start of the procedure. The following information should be available:

- IOP number on which the UN33/UN933 boards will be added or removed.
- Slot number in the IOP.
- UN33/UN933 board number.
- Scan point logical group name.
- Scan point number.
- Distributor point number.

The information needed can be found in the scan point assignments layout. The layout varies from site to site; therefore, it is necessary to consult the office data for an individual location.

The steps in the procedures are to be followed exactly; any changes should be documented after the completion of each procedure regardless of the outcome to ensure better future maintenance. If a system boot occurs, initiated by the system or craft, before the completion of the ECD disk update, the craft must redo all steps in that procedure. If errors were encountered when updating the ECD, note the problem and call for assistance according to the proper escalation procedures.

The **DGN:SCSDC** diagnostic should be run to make sure the UN33/UN933 boards and the switch are stable before starting the growth/degrowth procedures. Note that for the degrowing procedures, an alarm condition on the switch may not be really cleared. This is in spite of the state of the associated UN33/UN933 board and/or the alarmed scan point that is shown to be unequipped. The craft is warned that any alarm condition should be investigated before the degrowing procedure for that alarm unit may take place.

Growth Procedure for Adding a UN33/UN933 Board

The procedure for adding a UN33/UN933 board is not in this IP. Refer to the following IPs to perform the growth procedure for adding a UN33/UN933 board (SCSDC):

- If you are adding a UN33 to a 3B20D Model 2 or Model 3 computer, the procedures are found in 254-302-837, *Growth and Retrofit Tasks*.
- If you are adding a UN933 to a 3B21D computer, the procedures are found in 254-303-100, *Growth/Retrofit Tasks*.

After installing a UN33/UN933 board, scan points can be added by performing the "Growth Procedure for Adding Scan Points on a UN33/UN933 Board" section. The procedure to add scan points does not provide every single detail and assumes that you are familiar with RC/V sessions. If you are not familiar with RC/V sessions, please refer to 254-303-104, *Recent Change and Verify Manual*, for an example session and more information.

Growth Procedure for Adding Scan Points on a UN33/UN933 Board

Review the sections “Preparation and Requirements for Growth and Degrowth Procedures” and “Growth Procedure for Adding a UN33/UN933 Board” in this chapter before beginning this procedure. The UN33/UN933 board on which the scan points will be added should be installed in the IOP prior to the start of this growth procedure.

Change ECD to reflect the addition of the growth unit by performing the following procedure:

Step	Procedure
1	At the RC/V terminal, gain craft restricted command privileges. Enter log:on and then password.
2	Enter: rcv:menu;rcvecd,db "incore",nreview!
3	Perform Steps (a) through (h) to update scan points information. (a) Execute the <i>trbegin</i> form. (b) On the “form selection” page, enter scsdbody for the form name. (c) On the “database operation” page, enter u for update mode setting. (d) Display the <i>scsdbody</i> form associated with the growth unit. Enter scopt# in field 1 where # is the UN33/UN933 board number. Enter the scan point logical group name in field 2. ⇒ NOTE: The RC/V subsystem should display the remaining data on the current <i>scsdbody</i> form. If not, abort the growth procedure by entering “!” and go to Step (g). (e) Change field 9 (sc_list.sc_entry) on the <i>scsdbody</i> form. Enter c at the bottom of the screen. Enter 9 when asked for field. Enter the appropriate subfield number in field 9 when asked for row number. Enter the scan point number in the subfield.

Step	Procedure
	⇒ NOTE: Repeat the last two operations in Step (e) until all scan points in the current logical group are defined. If the scan points that need to be defined are in more than one logical group, continue with the current logical group until (and including) Step (h), and then restart the procedure with another logical group beginning with Step 3. (f) Update the <i>scsdbody</i> form. Enter u at the bottom of the screen. "FORM UPDATED" flashes once in the upper right corner of the screen when the form has completed its update. (g) Return to the "form selection" page. Enter < in field 1. (h) Execute the trend form to complete the RC/V transaction. If errors were encountered, go to Step 5. If no error were encountered, continue to Step 4.
4	Update ECD disk image. Execute the activate form.
5	Exit <i>UNIX</i> RTR RC/V subsystem. Enter < on the "form selection" page.
	⇒ NOTE: The ECD update is complete. Proceed to Step 8 if errors were encountered during the update.
6	Find out the current Process Identification Number (PID) of the alarm subsystem.
7	Reinitialize the alarm subsystem by restarting the process. At the RC/V terminal, enter: exc:envir:uproc.fn bin/kill,args PID ! Where <i>PID</i> is the number noted in Step 6.
8	Relinquish the restricted command privileges. Enter log:off
9	Note the changes in the operation logbook.

Degrowth Procedure for Removing Scan Points on a UN33/UN933 Board

Review the section "Preparation and Requirements for Growth and Degrowth Procedures" before beginning this procedure. The UN33/UN933 board on which the scan points will be removed should still be in place in the IOP.

Change ECD to reflect the addition of the degrowth unit by performing the following steps:

Step	Procedure
1	At the RC/V terminal, gain craft restricted command privileges. Enter log:on and then enter password.
2	Enter: rcv:menu;rcvecd,db "incore",nreview!
3	Perform Steps (a) through (h) to update scan point information. (a) Execute the trbegin form. (b) On the "form selection" page, enter scsdbody for the form name. (c) On the "database operation" page, enter u for update mode setting. (d) Display the <i>scsdbody</i> form associated with the growth unit. Enter scpt# in field 1 where # is the UN33/UN933 board number. Enter the scan point logical group name in field 2. ⇒ NOTE: The RC/V subsystem should display the remaining data on the current <i>scsdbody</i> form. If not, abort the growth procedure by entering "!" and go to Step (g). (e) Change field 9 (sc_list.sc_entry) on the <i>scsdbody</i> form. Enter s at the bottom of the screen. Enter 9 when asked for field. Enter the appropriate subfield number in field 9 when asked for row number. Enter the old scan point number when asked for old data (form:). Press the RETURN key when asked for new data (to:).

Step	Procedure

⇒ **NOTE:**

Repeat the last three operations in Step (e) until all scan points in the current logical group are defined. If more than one scan point logical group needs to be updated, continue with the current logical group until (and including) Step (h), and then restart the procedure with another logical group beginning with Step 3.

- (f) Update the *scsdbody* form. Enter **u** at the bottom of the screen. "FORM UPDATED" flashes once in the upper right corner of the screen when the form has completed its update.
- (g) Return to the "form selection" page. Enter **<** at field 1.
- (h) Execute the **trend** form to complete the RC/V transaction. If errors were encountered, go to Step 5. If no errors were encountered, continue to Step 4.

4 Update ECD disk image. Execute the **activate** form.

5 Exit *UNIX* RTR RC/V subsystem. Enter **<** on the "form selection" page.

⇒ **NOTE:**

The ECD update is complete. Proceed to Step 8 if errors were encountered during the update.

6 Find out the current PID of the alarm subsystem. At the MTTY, display the 150 process status page. Note the PID of the process with the process name.

7 Reinitialize alarm subsystem by restarting the process. At the RC/V terminal, enter:

exc:envir:uproc.fn bin/kill,args PID!

Where *PID* is the number noted in Step 6.

8 Relinquish the restricted command privileges. Enter **log:off**

Step	Procedure
9	If the board is to be removed, refer to the appropriate degrowth procedure for removing a UN33/UN933 board in 254-302-837, <i>Growth and Retrofit Tasks</i> , or 254-303-100, <i>Growth/Retrofit Tasks</i> .
10	Note the changes in the operation logbook.

Degrowth Procedure for Removing a UN33/UN933 Board

Refer to the following IPs to perform the degrowth procedure for removing a UN33/UN933 board (SCSDC).

- If you are removing a UN33 board from a 3B20D Model 2 or Model 3 computer, refer to 254-302-837, *Growth and Retrofit Tasks*.
- If you are removing a UN933 board from a 3B21D computer, refer to 254-303-100, *Growth/Retrofit Tasks*.

Procedure for Adding Logical Groups

Review the section "Preparation and Requirements for Growth and Degrowth Procedures" in this chapter before beginning this procedure. The UN33/UN933 board on which the new logical group will be added should be in place in the IOP. If not, refer to the appropriate growth procedure for adding a UN33/UN933 board in 254-302-837, *Growth and Retrofit Tasks*, or 254-303-100, *Growth/Retrofit Tasks*.

Insert a new logical group into ECD by performing the following procedure:

Step	Procedure
1	At the RC/V terminal, gain craft restricted command privileges. Enter log:on and then enter password.
2	Enter rcv:menu;rcvecd,db "incore",nreview!
3	Perform Steps (a) through (l) to enter SCSD points information. (a) Execute the trbegin form.

Step	Procedure
	(b) On the "form selection" page, enter scsdbody for the form name. (c) On the "database operation" page, enter i for update mode setting. (d) Insert the new logical group. Enter the option name of the UN33/UN933 board that will be monitoring this logical group in field 1. Enter the name of the new logical group in field 2. (e) Enter the duplex identification number (0 or 1) in field 3. (f) Enter the scan class to which this logical group belongs in field 4. (g) Enter the unit name and number (if any) in fields 7 and 8, respectively. ⇒ NOTE: Required for devices that will be monitored by this UN33/UN933 board. (h) Enter the scan points (if any) that will be monitored by this group in field 9 (sc_list.sc_entry) subfields. (i) Enter the distributor points (if any) that will be monitored by this group in field 11 (sd_list.sd_entry) subfields. (j) Insert the scsdbody form. Enter i at the bottom of the screen. "FORM INSERTED" flashes once in the upper right corner of the screen when the form has completed its insertion. Repeat Steps (d) through (j) to insert other new logical groups. (k) Return to the "form selection" page. Enter < in field 1. (l) Execute the trend form to complete the RC/V transaction. If errors were encountered, go to Step 5. If no errors were encountered, continue to Step 4.
4	Update ECD disk image. Execute the activate form.
5	Exit <i>UNIX</i> RTR RC/V subsystem. Enter < on the "form selection" page.
	⇒ NOTE: The ECD insert is complete. Proceed to Step 7 if errors were encountered during the insertion.

Step	Procedure
6	Manually remove and restore the UN33/UN933 board (SCSDC) that will be monitoring the new logical group.
7	Relinquish the restricted command privileges. Enter log:off
8	Note the changes in the operation logbook.

Procedure for Deleting Logical Groups

Review the section "Preparation and Requirements for Growth and Degrowth Procedures" before beginning this procedure. The UN33/UN933 board on which the logical group will be deleted should be in place in the IOP.

Delete from ECD a logical group by performing the following procedure:

Step	Procedure
1	At the RC/V terminal, gain craft restricted command privileges. Enter log:on and then enter password.
2	Enter rcv:menu;rcvecd,db "incore",nreview!
3	Perform Steps (a) through (g) to delete SCSD points information. (a) Execute the trbegin form. (b) On the "form selection" page, enter scsdbody for the form name. (c) On the "database operation" page, enter d for delete mode setting. (d) Display the <i>scsdbody</i> form associated with the logical group to be deleted. Enter the option name of the UN33/UN933 board that will be monitoring this logical group in field 1. Enter the name of the logical group in field 2.

⇒ NOTE:
The RC/V subsystem should display the remaining data on the current *scsdbody* form. If not, abort the delete procedure by entering "!" and go to Step (f).

Step	Procedure
	(e) Delete the <i>scsdbody</i> form. Enter d at the bottom of the screen. "FORM DELETED" flashes once in the upper right corner of the screen when the form has completed its deletion. Repeat Steps (d) and (e) to delete other logical groups. (f) Return to the "form selection" page. Enter < in field 1. (g) Execute the trend form to complete the RC/V transaction. If errors were encountered, go to Step 5. If no errors were encountered, continue to Step 4.
4	Update ECD disk image. Execute the activate form.
5	Exit <i>UNIX</i> RTR RC/V subsystem. Enter < on the "form selection" page.
	⇒ NOTE: The ECD insert is complete. Proceed to Step 7 if errors were encountered during the deletion.
6	Manually remove and restore the UN33/UN933 board (SCSDC) that monitored the deleted logical group.
7	Relinquish the restricted command privileges. Enter log:off
8	If the board is to be removed, refer to the appropriate degrowth procedure for removing a UN33/UN933 board in 254-302-837, <i>Growth and Retrofit Tasks</i> , or 254-303-100, <i>Growth/Retrofit Tasks</i> .
9	Note the changes in the operation logbook.

Input/Output Processor Growth and Degrowth

The IOP interfaces the 3B20D/3B21D computers with as many as four groups, each group containing four peripheral devices for a total of 16. The allocation is two groups each containing four devices on the basic unit (J1C130BA) and two groups each containing four devices on the optional growth unit (J1C130BB). Both units are mounted on the peripheral control frame (J1C136AF) along with the required fan unit (J1C129AF).

The IOP growth and degrowth procedures for the 3B20D Model 2 or Model 3 computer can be found in the 254-302-837, *Growth and Retrofit Tasks*. The IOP growth and degrowth procedures for the 3B21D computer can be found in the 254-303-100, *Growth/Retrofit Tasks*.

If a growth unit is being added to an existing basic unit, the present fan position must be lowered to accept the optional growth IOP. This requires the basic unit to be removed from service and the power removed for the necessary wiring changes. After the wiring is complete, the *chopt* record should be reviewed to verify that the fields relating to the additional unit are set. These fields are usually set to the maximum values in the standard ECD shipped to all offices.

The IOP is connected to the 3B20D/3B21D computer via a Dual Serial Channel (DSCH) associated with a Direct Memory Access Controller (DMAC) in the DMA Input/Output (IO) unit.

Growth Requirements

The general growth procedures requirements relating to the growth activity in an operational 3B20D/3B21D computers environment can be found in the 254-302-837, *Growth and Retrofit*.

All information concerning the logical locations in the database should be obtained using *dbinfo* which is described in 254-303-104, *Recent Change and Verify Manual*. Copies of the forms to be updated and/or inserted should be filled in before beginning the growth procedures.

Backout Procedures

Backout procedures are those actions taken to return the 3B20D/3B21D computers to the original operating configuration before the growth was attempted. Reversing the steps in descending order suffices as a backout procedure. It is assumed that backout procedures will not be executed after the backup disk has been updated, rather normal troubleshooting techniques should be used. Therefore, the craft and customer must be certain that the IOP is not causing any system degradation prior to updating the backup disk copy of the ECD.

Verifying SCSD Operation

To verify the operation of either the scan points or the distributor points, manual action is required. To verify SCSD operations, follow the procedures in the succeeding paragraphs.

Verifying Scan Point Status

Operation of the scanner portion of the SCSD is controlled by the following messages:

- **OP:SCSD** — Reports the inhibit status and the state of the scan points on the SCSD.
- **INH:SCSD** — Inhibits the reporting of transitions of a scan point on the SCSD.
- **ALW:SCSD** — Allows the reporting of transitions of a scan point on the SCSD.

Obtain physical scan point assignments from schematic drawings or office records. The physical and logical assignments of scan points are recorded in the ECD and can be accessed via the RC/V mechanism. To determine logical and physical scan point assignments using the RC/V mechanism, use the following procedure.

Step	Procedure
1	At the recent change terminal, enter: In PDS, RCV:MENU:RCVECD,DB "root",NREVIEW! In MML, RCV:MENU:DATA,RCVECD,DB="root",NREVIEW; The "UNIX RTR RCV (ODIN)-DATA ENTRY" page is displayed.
2	Enter form name klist.
3	Enter in block number 1, a temporary filename; for example, /tmp/scsd.
4	Enter n in all blocks except <i>scsdbody</i>.
5	At <i>scsdbody</i>, enter y.
6	In the field at the lower right corner, enter e.
7	At the maintenance terminal, enter: In PDS, DUMP:FILE:ALL, FN"/tmp/scsd" In MML, DUMP:FILE:DATA,ALL, FN="/tmp/scsd"; The contents of the <i>/tmp/scsd</i> file are printed. A list of the SCSD option and logical names make up the contents.
8	Enter a <i>form name</i>, for example, scsdbody.
9	Enter an <i>option name</i>, for example, scsdopt0.
10	Enter an <i>scsdname</i> (logical group name). The remainder of the form is populated. Column 9 contains scan point assignments, and column 11 contains SD point assignments.

Verifying SD Point Operation

To verify the operation of an SD point, the selected point must be identified by either physical or logical assignment. The method of identification is the same as that used to identify scan point assignments.

Once identified, the SD points can be placed in any of the following states using the **ORD:SCSD** input message:

- OPERATE — set
- RELEASE — clear
- STEP — flash for 8 seconds
- REPEAT — flash continuously.

To verify operation of the selected SD point, use the **ORD:SCSD** message to flash the point while observing the state of associated lamps and indicators.

Typical SD Point Verification Example:

Verify the operation of the SD points on MHD 0.

Step	Procedure
1	Identify SD point assignment from the office records. For example, the 3B20D computer scan point and signal distributor point assignments shown in Table 16-6 for the MHD 0 SD points are assigned as follows: ▪ MHD 0 — Lead OOS is member number 0, SD point 18. ▪ MHD 0 — Lead RQIP is member number 0, SD point 19.
2	Using information from the previous step, enter the command: In PDS, ORD: SCSD; UNIT 0: PT 18, OPR STEP! In MML, ORD: SCSD; UNIT=0: DATA, PT=18, OPR=STEP;
3	At MHD 0, observe the OOS lamp. If the SD point is operating, the lamp will flash for 8 seconds.
4	Repeat using the data associated with lead RQIP.

Troubleshooting the SCSD

You should have access to SCSD assignments and be familiar with using the **OP:SCSD** and **ORD:SCSD** commands.

The SCSD diagnostics test the circuitry on the pack but do not loop through the outside connections. For problems with single scan points, the point should be manually operated. The transition of the point can be verified with the **OP:SCSD** command. A problem with a single signal distributor point can be investigated using the **ORD:SCSD** command. The operation of all lamps can be tested with the ACO/T switch on the power switch (TN2, TN3, TN5, and TN6).

Spare Circuit Packs

17

Contents

Introduction	17-1
General Guidelines	17-1

Introduction

Achieving high reliability in the 3B20D/3B21D computers requires that each office have available a stock of operational spare circuit packs. The recommended number of spares the office should stock are a minimum of one spare circuit pack per code for every 3B20D computer and every 3B21D computer.

Another option for maintaining spare stock is the clustering of spares if spare quantities are based on pack failure rates and repair intervals.

General Guidelines

Observe the following guidelines for testing spare circuit packs:

- Ensure that circuit packs are handled and stored in compliance with 032-173-301, *Testing, Replacing, Handling, Storing, and Shipping Circuit Packs and Semiconductor Devices*.
- Ensure that your office has the recommended spare inventory.
- Establish a circuit pack inventory and testing log (Figure 17-1) to “track” all circuit packs including those used for troubleshooting.

[illegible]

Figure 17-1. Circuit Pack Inventory and Testing Log

Contents

Introduction	18-1
Field Test Set	18-1
■ Description	18-2
GRASP/EGRASP	18-3
■ GRASP/EGRASP Capabilities	18-3
■ EGRASP Capabilities	18-4
Software Debugging Tools	18-4

Introduction

The following sections describe the available 3B20D/3B21D computer maintenance tools.

- Field Test Set (FTS)
- Generic Access Package (GRASP)/Enhanced Generic Access Package (EGRASP)
- Software Debugging Tools.

Field Test Set

For *UNIX*® Real-Time Reliable (RTR) Operating System Release 1 only, the 3B20D/3B21D computers FTS is a troubleshooting tool that monitors problems in program execution in the Control Unit (CU) at an office without disrupting service. The office maintenance person will be expected to perform the following:

- Connect the FTS to the CU.
- Boot the FTS.
- Input programs to the FTS.
- Output data from the FTS.

Description

The FTS is equipped for local or remote mode operation. In the remote mode, the programs are loaded and data is output from a remote site.

The FTS monitors the CU by tracing data flow information between the CU and main store and keeps a trace of system calls, processes, and instructions.

The following items are included with the FTS:

- Data set
- Five circuit boards
 - 8086 circuit board, MC4067, TN686
 - Two 512K Random Access Memory (RAM) circuit boards, TN685
 - Input/Output (IO) interface board, UN62
 - Dual Utility Circuit (DUC), UN61.
- Eight-foot ribbon cable
- Four cassette tapes
 - FTS diagnostics
 - FTS scratch tape
 - DUC software
 - *UNIX* system software.

In addition, the following items must be available:

- Cathode-Ray Tube (CRT) terminal dedicated to use with the FTS
- AC power available near the FTS location
- Modular phone (line) for remote use
- Acoustic coupler for remote use
- Two-way map between process name and unit identification
- Source listings labeled by process name (not required for remote mode)
- Function/address listings by process name (not required for remote mode).

GRASP/EGRASP

The GRASP in the computer is a single-user utility system and is a subsystem of the generic release of the operating system software. GRASP allows software maintenance personnel to observe the behavior of the operating system software in an operational environment. GRASP aids in the analysis of a software fault and should be considered a means of observing the flow of system software to isolate software bugs. It is intended to be used to gather information on a known problem rather than to detect or correct problems.

The EGRASP (effective with *UNIX* RTR Operating System Release 6) is a resident software package which provides on-line real-time software debugging capabilities. EGRASP supports an interface to the UN21, UN61, UN615, and UN379 utility circuits to provide trace and matching functions. In addition, EGRASP contains all of the functionality of GRASP. Thus, it provides the capability to place multiple breakpoints, read and write memory and registers, and dump the contents of memory and machine registers. The utility circuit packs monitor operations between the computer CU and the computer main memory.

GRASP/EGRASP is controlled by input from the maintenance terminal or from the switching control center. GRASP/EGRASP output appears on the maintenance terminal or read-only printer. Although GRASP/EGRASP can be used by the office maintenance person, it should be used in consultation with your next level of technical support. Improper use of GRASP/EGRASP can result in program mutilation or excessive utilization of system resources.

GRASP/EGRASP Capabilities

GRASP/EGRASP capabilities include the following:

- Data Transfer Functions — COPY, DUMP, and LOAD provide a means to move, print, and write data to addressable system locations.
- Breakpoints — WHEN provides the capability to execute instructions and perform utility functions when the program flow reaches or matches a specified condition.
- Breakpoint Manipulation Commands — These commands allow the GRASP/EGRASP user to create, save, view, and remove breakpoints.
- Override Commands — These functions enable the user to override GRASP/EGRASP default time limits.
- Transfer Trace — The primary function of the transfer trace is to indicate the flow of execution around a target event; for example, branch instruction.

EGRASP Capabilities

For *UNIX* RTR Operating System Release 6 only, EGRASP capabilities include the following:

- Data History Trace — Records accesses to or from memory.
- Simultaneous Data History and Transfer Trace — Records any operation that reads or writes to memory and any program transfer and context switch.
- Function Calls and Returns — Records call and return instructions.
- Traces Function Calls with Parameters Passed to Functions — Records the program address calling the function, the address of the function being called, and the parameters of the associated function passed on the stack.
- Data History and Function with Parameter Trace — Records information associated with both a data history and a trace of function calls with parameters passed.
- Matching — Matching on one or more of the following conditions: Maintenance Reset Function (MRF), a stop and switch, and external input.

Software Debugging Tools

Refer to 254-303-107, *Software Troubleshooting Guide*, for general information on how to use the following tools provided for debugging software:

- **browse** Command
- **ibrowse** Command
- **cx** Command
- Generic Access Package
- **idump** Command.

Also 254-303-107, *Software Troubleshooting Guide*, contains Operating System Trap (OST) service routines for kernel and supervisor processes, hexadecimal offset charts, register layouts, and an American Standard Code for Information Interchange (ASCII) to hexadecimal conversion chart.

Contents

Introduction	19-1
Firmware vs. Pumpcode	19-2
Verify the Integrity of Pumpcode	19-4
Installing New Pumpcode	19-5
SCSI Disk Drives	19-7
SCSI Tape Drives	19-9
■ Drive Identification	19-9
■ Removing from or Restoring to Service	19-10
■ Diagnosing	19-10
■ Removing from or Inserting into the System	19-11
■ Cleaning	19-13
Cleaning KS-23909, L10 9-Track Tape Drives	19-13
Cleaning KS-23909, L21 9-Track Tape Drives	19-16
Cleaning UN376 (NCR006-3503341), UN376C (NCR006-3300608/Comcode 407545243), or UN376E (KS-24367, L1/Comcode 407771260) DAT Drives	19-19
■ Setting Device ID	19-21
Setting Device IDs for KS-23909, L10 9-Track Tapes	19-21
Setting Device IDs for KS-23909, L21 9-Track Tapes	19-24
Setting Device IDs for DAT Drives	19-25
■ Write Protection	19-27
9-Track Drive Write Protection	19-27
DAT Drive Write Protection	19-27
■ Tape Capacity/Tape Density	19-28

Contents

9-Track Drive Tape Capacity	19-28
9-Track Drive Tape Density	19-28
DAT Drive Tape Capacity	19-29
DAT Drive Tape Density	19-29
■ Loading/Unloading 9-Track and DAT Tapes	19-30
Loading or Unloading 9-Track Tapes	19-30
Loading the KS-23909, L10 9-Track Tape Reel (Auto Load)	19-30
Loading (Lacing) the KS-23909, L10 9-Track Tape by Hand (Manual)	19-31
Removing the KS-23909, L10 9-Track Tape Reel with Power On	19-32
Removing the KS-23909, L10 9-Track Tape Reel During a Power Failure	19-33
Loading the KS-23909, L21 9-Track Tape Reel	19-34
Unloading the KS-23909, L21 9-Track Tape Reel	19-36
Manually Rewinding the KS-23909, L21 9-Track Tape Reel	19-36
Loading or Unloading DAT Drives (NCR006-3503341, NCR006-3300608/Comcode 407545243, or KS-24367, L1/Comcode 407771260)	19-37
Loading the DAT Cartridge Tape/Cassette	19-37
Ejecting the DAT Cassette	19-38
■ DAT Drive Cartridge Tapes and Status	
Indicators	19-39
NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT Drive Status Indicators	19-40
NCR006-3503341 SCSI DAT Drive Status Indicators	19-40

Introduction

This section presents information which is helpful in maintaining the Small Computer System Interface (SCSI) peripheral system. A typical SCSI Disk File Controller (DFC) consists of a Duplex Dual Serial Bus Selector (DDSBS), a SCSI Host Adapter (HA), two SCSI buses, and several SCSI devices. The SCSI DFC interfaces the 3B20D/3B21D computers with the SCSI devices; it translates the requests from the 3B20D/3B21D computers into SCSI commands and sends the translated request to the SCSI device's embedded controller.

The SCSI peripheral system can coexist with the Storage Module Drive (SMD) system on the 3B20D computer. However, the SMD and SCSI disk drives cannot coexist under the same DFC.

Firmware vs. Pumpcode

The SCSI HA can be controlled by either firmware or pumpcode.

- Firmware is the compiled or assembled program/data that resides in nonvolatile memory [usually Electronic Programmable Read-Only Memory (EPROM), Programmable Read-Only Memory (PROM), or Read-Only Memory (ROM)] on the SCSI HA circuit board. The information embodied in the firmware is immediately available on powerup of the circuit board.
- Pumpcode resides on a disk and is loaded (for example, pumped) into the volatile memory [or Random Access Memory (RAM)] on the circuit board.

The objectives of the firmware and of the pumpcode in decreasing order of priority, are as follows:

- Firmware
 - Robustness
 - Maintainability
 - Performance.
- Pumpcode
 - Performance
 - Robustness
 - Maintainability.

The firmware establishes control of the SCSI HA under the following situations:

- When the DFC is powered up or diagnosed
- When the DFC is in the Out of Service (OOS) major state
- When the DFC is reset by the DFC driver
- When the DFC fails to download the pumpcode.

The pumpcode is downloaded to the SCSI DFC under the following situations:

- Within 5 minutes after a system boot
- After a successful restore of the DFC from OOS to active state
- As part of Phase 1 recovery for the DFC
- When a user manually requests the download.

The pumpcode will begin to run on the SCSI HA after a successful download. There are output messages to indicate the start, in-progress, and completion of the pump procedure.

While the pumpcode is in operation, if a regular (nonmaintenance) Input/Output (IO) job for a disk fails, the IO job will be retried several times. On the 3B21D computer, if a regular (nonmaintenance) IO job for a tape device fails, the IO job will not be retried. If the IO job still fails, the pumpcode will be removed and the job will be retried with firmware. A successful completion of the same job by the firmware indicates that there could be a problem with the pumpcode (for example, pumpcode is corrupted); the pumpcode hence will not be downloaded again unless by manual request. This strategy intends to avoid thrashing by repeatedly downloading the bad pumpcode. It also helps to avoid unnecessary removal of a disk. If the same job fails when retried by firmware, the disk will go through the normal recovery by the DFC driver (for example, disk is removed from service), and the pumpcode will be downloaded again.

In order for the 3B20D/3B21D computers DFC driver to download the pumpcode, there must be a pumpfile residing on disk. The pumpfile must be in the directory */dfc* and must be a single extent contiguous file. The name of the pumpfile has the format *xyz*, where *x* is the pack code (for example, *u294*), *y* is the version number, and *z* is the issue number of the firmware; for example, *u29471* is the pumpcode corresponding to firmware Version 7 and Issue 1. However, the pumpfile used for a manual pump request can be a regular file and also can be in any directory in any file system on the 3B20D/3B21D computers.

Verify the Integrity of Pumpcode

This section presents a procedure to verify the integrity of the SCSI pumpcode. This procedure could be used occasionally to make sure that the pumpcode which resides on the disk is not corrupted.

Step	Procedure
1	Verify that the pumpfile is valid. Enter: In PDS, LOAD:DFC a;PUMP:FN "/dfc/xyz",CHECKSUM! In MML, LOAD:DFC=a:PUMP:FN="/dfc/xyz":DATA,CHECKSUM; Where: - a = DFC number. - x = Pack code. - y = Version number. - z = Issue number of the firmware.  NOTE: The pumpfile must exist in the directory /dfc since /dfc is the directory specified in the command line. Wait for completion message.
2	If the pumpcode is good, the completion output message is as follows: LOAD DFC a PUMP CHECKSUM COMPLETED VALID PUMPCODE If the pumpcode is corrupted or invalid, the following message will be output: LOAD DFC a PUMP CHECKSUM COMPLETED INVALID PUMPCODE
3	Contact your technical assistance organization immediately if the pumpcode is corrupted or invalid.

Installing New Pumpcode

This section presents a procedure to install a new pumpcode for a SCSI DFC.

Step	Procedure
1	Create a temporary directory called <i>/newpump</i> in the root file system. Install the new pumpfile; for example, reading from tape into directory <i>/newpump</i> . Record the name of the new pumpfile (for example, <i>xyz</i>), where <i>x</i> is the pack code (for example, <i>u294</i>), <i>y</i> is the version number, and <i>z</i> is the issue number of the firmware.
2	Verify that the version number of the firmware is the same as that of the pumpcode. Enter: In PDS, OP:DFC a;INFO! In MML, OP:DFC=a:INFO; Where: <i>a</i> = DFC number. Wait for the output message. Compare the first digit of the version number of the firmware and new pumpcode. If they are not the same, DO NOT CONTINUE . Otherwise, continue to Step 3.
3	Verify that the new pumpfile is valid. Enter: In PDS, LOAD:DFC a;PUMP:FN "/newpump/xyz",CHECKSUM! In MML, LOAD:DFC=a:PUMP:FN="/newpump/xyz":\ DATA,CHECKSUM; Where: <i>a</i> = DFC number. <i>x</i> = Pack code. <i>y</i> = Version number. <i>z</i> = Issue number of the firmware. Wait for completion message. If the CHECKSUM is valid, continue to Step 4. Otherwise, DO NOT CONTINUE .

Step	Procedure
4	Request a DFC pump to download the new pumpcode. Enter: In PDS, LOAD:DFC a;PUMP:FN "/newpump/xyz"! In MML, LOAD:DFC=a:PUMP:FN="/newpump/xyz"; Where: - a = DFC number. - x = Pack code. - y = Version number. - z = Issue number of the firmware. Wait for completion message. If the pump completes successfully, continue to Step 5. Otherwise, DO NOT CONTINUE to download the new pumpcode but try to download the old pumpcode.
5	Soak the new pumpcode for at least 24 hours.
6	If there is no problem with the new pumpcode, continue to Step 7. Otherwise, reload the old (official) pumpcode. Enter: In PDS, LOAD:DFC a;PUMP! In MML, LOAD:DFC=a:PUMP; Where: a = DFC number. Wait for the completion message. Contact your technical assistance organization immediately about the problem with the new pumpcode. DO NOT CONTINUE.
7	Copy the file <i>/dfc/xyz</i>, which is the current pumpfile, to <i>/dfc/xyz.ofc</i>. Where x, y, and z are as previously indicated. ⇒ NOTE: Keep this pumpfile as the backup for around 1 week before removing it permanently. If there is any problem with the new pumpcode during this week, the backup pumpcode can be downloaded again.
8	Copy the pumpfile <i>xyz</i> residing in directory <i>/newpump</i> to the same pumpfile name in directory <i>/dfc</i>.
9	Remove the pumpfile in the directory <i>/newpump</i> and the directory itself.

SCSI Disk Drives

This section provides information about characteristics of the 3B20D/3B21D computer SCSI disk drives. The following is a list of items that are particular to SCSI disk drives:

- A SCSI disk drive typically takes about 1 minute after power up to be fully accessible.
- There is a switch to set the device identification (ID) of the drive. The ID switch should only be set to a value between 0 and 6. No two drives can have the same device ID on the same SCSI bus. The device ID number **7** is reserved for and used by the SCSI HA on the SCSI bus. The SCSI disk drive must be powered down before changing the ID switch.
- A SCSI disk is connected to a Small Computer System Interface Bus (SBUS) under a SCSI HA. The SCSI bus must be removed from service before disconnecting or connecting the SCSI disk drive. There are two SCSI buses for each SCSI HA. On the 3B20D computer, the SCSI bus can support up to four SCSI disk drives. On the 3B21D computer, the SCSI bus can support up to seven SCSI disk drives. The SCSI bus can be removed/restored without affecting the operation of the other SCSI bus on the same SCSI HA.
- The **VFY:MHD** command which is used to verify the disk drive will only provide logical block number(s) for the defective disk block(s). Likewise, the **LOAD:MHD:DEFECT** command will only accept a list of defects in logical block number format (typically the output by the **VFY:MHD** command) for the SCSI disk drive instead of physical format as for the SMD disk drive.
- The **LOAD:MHD;DEFECT** command is used to reallocate defective blocks on the SCSI disk drive. The disk should be reformatted after bad blocks are reallocated. Refer to the "Adding to the Defect Table of a SCSI Disk Drive" section of Chapter 10 for the procedure. After a successful reallocation of the defect(s), all previous information about the defect(s) or bad block(s) become invalid. As an example, logical block number 3, if reallocated, moves from one physical block (the defective one) to another physical block. It is possible for (new) logical block number 3 to become defective and need to be reallocated.
- Multiple **VFY:MHD** commands are allowed to run concurrently for different SCSI disk drives on the same DFC or same SBUS. In the SMD DFC, only one **VFY:MHD** command at a time is allowed to run. This is because the SMD firmware does not support enough buffers to hold the verify failure data for multiple disk drives. The SCSI firmware/pumpcode does.
- Disk formatting is done completely by the embedded controller in the SCSI disk drive; for the SMD DFC, the firmware actually does the formatting on a disk drive.

- There are two types of the defect management lists for a SCSI disk drive:
 - *Primary Defect List* (or P list) is the list of defects originally supplied by the manufacturer of the device and are considered as permanent flaws. This list is not subject to additions.
 - *Grown Defect List* (or G list) consists of defects introduced as the result of the successful completion of the **LOAD:MHD;DEFECT** command. This list does not include the Primary Defect List. The Grown Defect List can only be grown and cannot be erased. Therefore, great care should be taken when adding defects into the Grown Defect List (via **LOAD:MHD;DEFECT** command).

The G and P defect lists can be listed using the **DUMP:MHD;DEFECT** command. However, the defect lists are shown only in the physical format (Cylinder/Head/Sector). The physical format of the defect list is not useful for defect management, it only provides general information to the user; the user can at least know the number of defects for each list.

- If any disk drive SMD or SCSI is powered down while in operation, especially when there is a write operation in progress, there is a risk that the drive could sustain an uncorrectable data error. Therefore, a disk should only be powered down after removing it from service and no write activity is being performed on it.
- The “diagnostic area” is a special area on the disk used only for diagnostic purposes and resides outside the user accessible area. For the SCSI disk, 256 blocks are used for diagnostic purposes. For the SMD disk drives, one cylinder is used by the DFC diagnostic firmware for this purpose.

SCSI Tape Drives

This section provides information about characteristics of the 3B21D computer SCSI tape drives only, their operations, and maintenance. The following subsections are arranged to guide the user from the off-line status of the device to the ready-to-operate status and then through visual diagnostic indications in case of malfunction.

The 3B21D computer supports the following two types of SCSI tape drives:

- A vertically mounted 9-track SCSI tape drive which uses an industry standard ½-inch, reel-to-reel magnetic tape. Use of 9-track media that complies to American National Standards Institute (ANSI) Std.3.40-1983 (or later issue) or ECMA 62 specification is recommended.
- A circuit pack (UN376) mounted Digital Audio Tape (DAT) Unit which uses a 4-millimeter wide tape cassette that conforms to industry Digital Data Storage (DDS) standard. Tape media cartridges designated with the DDS logo are required to be used in the DAT unit. DAT tapes designated with the DDS logo meet requirements of ECMA-139 and ANSI X3.206 standards.

Drive Identification

The 9-track tape drives can be identified in the following two ways:

- A label with the KS-number is located on the rear of the unit.
- Through software by entering the following command for Program Documentation Standard (PDS) and Man-Machine Language (MML):

OP:MT x;INFO

Where x is the tape drive number; for example, **MT 2**.

The output on the maintenance terminal screen will display the KS-number and product revision number of the tape drive under the **PRODUCT IDENTIFICATION** field.

The DAT drive can be identified through software as previously described for the 9-track tape drive.

Removing from or Restoring to Service

To remove a tape drive from service, execute the following command for PDS and MML:

RMV:MT x

Where x is the tape drive number; for example, **MT 2**.

The status of the tape drive should change from Active (ACT) to OOS on the corresponding DFC Display Page on the Maintenance Terminal (MTTY) indicating that the drive is now out of service.

To restore the tape drive to service only if diagnostics report All Tests Pass (ATP), execute the following command for PDS and MML:

RST:MT x

To restore the tape drive to service unconditionally (no diagnostic tests are performed), execute the following command for PDS and MML:

RST:MT x;UCL

The status of the tape drive should change from OOS to ACT on the corresponding DFC Display Page on the MTTY indicating that the drive is now back in service.

Diagnosing

Refer to Chapter 6, "Diagnostics," for information on performing diagnostics on SCSI tape drives.

Removing from or Inserting into the System

A SCSI tape drive is connected to a SCSI bus. The SCSI bus must be removed from service before disconnecting and connecting the SCSI tape drive or before removing or inserting the UN376 circuit pack.

A SCSI bus can be removed or restored without affecting the operation of the other SCSI bus on the same SCSI HA.



CAUTION:

If it is necessary to remove from service or power down a tape drive while writing to it, the tape operation should be considered to have failed.

To remove a tape drive from the system, use the following procedure:

Step	Procedure
1	Remove the tape drive from service, and execute the following command for PDS and MML: RMV:MT x Where x is the tape drive number; for example, MT 2 .
2	Remove from service the SBUS to which the tape drive is connected, and execute the following command for PDS and MML: RMV:SBUS y Where y is the SBUS number; for example, SBUS 1 .
3	Run the drive diagnostics on the 9-track tape drive from its front control panel. (Refer to 9-track tape drive vendors' manuals; KS-23909, L40 for KS-23909, L10 tape drive, and KS-23909, L30 for KS-23909, L21 tape drive.)  CAUTION: <i>The 9-track tape drive should be removed from service (OOS), before drive self-diagnostics are initiated from its front control panel.</i>
4	If necessary, remove a 9-track tape drive electrically from service by turning off the power on-off switch on the drive. Remove a DAT drive electrically and mechanically out of service by pulling out the circuit pack "latch switch" on the corresponding UN376 circuit pack. (Refer to 254-303-105, <i>Hardware Reference Manual</i> .)

To restore a tape drive into the system, use the following procedure:

Step	Procedure
1	Restore a 9-track tape drive electrically to service by turning on the power on-off switch on the drive. Restore a DAT drive mechanically and electrically to service by inserting the UN376 circuit pack (which the DAT drive is on) into its corresponding slot in the 3B21D computer frame and pushing the circuit pack "latch switch" in.
2	Restore the SBUS (that the tape drive is connected to) to service, and execute the following command for PDS and MML: RST:SBUS y Where y is the SBUS number; for example, SBUS 1 .
3	Restore the tape drive to service: ▪ Conditionally. (The restore operation will be successful only if all the diagnostic tests are ATP.) Execute the following command for PDS and MML: RST:MT x Where x is the tape drive number; for example, MT 2. ▪ Unconditionally. (No diagnostic tests are performed.) Execute the following command for PDS and MML: RST:MT x;UCL

Cleaning

Clean tape path, tape heads, and tapes reduce read/write errors, shorten read/write times, and lengthen tape life. Routine cleaning of the tape drive significantly increases its performance.

A tape sheds binder and oxide during normal tape operations. Usually, debris from a tape can be removed by periodic cleaning. However, some combinations of humidity, temperature, tape tension, tape speed, and chemical composition of the binder in certain brands of tape may cause a deposit to be formed on the head. These deposits cannot be removed by normal cleaning procedures. Sometimes these deposits cannot even be seen. If allowed to accumulate, the head of a 9-track drive or the DAT drive itself will have to be replaced.

The recommended tape path cleaning interval is every 8 hours. However, if the error messages/status indications (such as dual density 1600/6250 display with or without UNKNOWN displayed on the KS-23909, L21 drive display panel) begin to occur regularly, the tape path should be cleaned more regularly. If frequent cleaning does not improve reliability, the tape media should be checked for wear, tear, and dirt. Old and worn tapes should be copied immediately and then discarded.

Cleaning KS-23909, L10 9-Track Tape Drives

Table 19-1 indicates a cleaning frequency and a suitable cleaner for various parts of the KS-23909, L10 9-track tape drive.

Table 19-1. Cleaning Schedules and Equipment

Assembly	Cleaning Frequency	Suitable Cleaner
Heads, tape cleaner, fixed guides, rollers	Daily if in continuous tape motion	Texpads (IPA-moistened pads), non-CFC equivalent with lint-free cloth, or foam swabs
Tacho roller	As stated above	As stated above
Painted surfaces	As required	Amberclens antistatic foaming cleaner or non-CFC equivalent
Loading door window	As required	Ambersil glass cleaner or non-CFC equivalent with antistatic cloth

To clean the KS-23909, L10 9-track tape drive path, use the general guidelines in the following procedure:

 NOTE:

Refer to 254-303-101, *Routine Maintenance Tasks*, for the procedure to clean KS-23909, L10 tape drives.

Step	Procedure
1	To gain access to the tape path, pull the handle on the loading door to hinge it open. The tape path components are shown in Figure 19-1.
2	Using the recommended cleaning pad, carefully wipe the surfaces of the head to remove all traces of dirt and oxide. In extreme cases, it may be necessary to use a scrubbing action to remove hard oxide deposits.  CAUTION: <i>The edges of the sapphire cleaner are extremely sharp. Observe great care when cleaning. Avoid inhaling an excess of fumes from the cleaning fluids.</i>
3	Using the recommended cleaning pad, wipe the tape cleaner, particularly the front edge where debris may have accumulated.
4	Inspect the remaining tape path components and, if necessary, wipe clean the tape contact surfaces of the roller guides, tension arm roller, and the surface of the tachometer roller.  CAUTION: <i>Do not use alcohol-based fluid products on plastic or painted parts. Do not allow cleaning fluid to enter pregreased roller bearings.</i>
5	Clean any dirt or oxide which may have gathered around the take-up and supply reel areas.
6	Close the loading door.

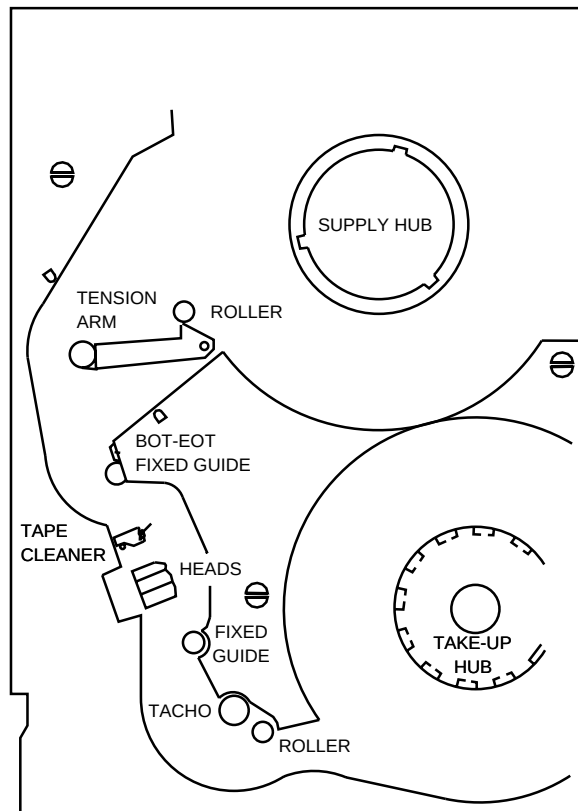


Figure 19-1. Tape Path Components

For cleaning other areas, use the following procedure:

Step	Procedure
1	Use an antistatic cleaner and a cloth on the tape path cover.
2	Use a cloth moistened with soapy water to wipe the fascia and painted surfaces.

Cleaning KS-23909, L21 9-Track Tape Drives

Table 19-2 indicates the cleaning criteria for the KS-23909, L21 9-track tape drive.

Table 19-2. Cleaning Schedule Guidelines

Cleaning Level	Cleaning Criteria
MINIMUM	Clean the tape path thoroughly EVERY 8 HOURS if: ▪ Less than ten reels are used in 8 hours. ▪ You see no particles on the tape head after each reel of tape. ▪ You do not suspect abnormal dust (from increased traffic or vacuuming) in the room where the drive is located.
NORMAL	Clean the tape path thoroughly EVERY 1 TO 2 HOURS of continuous running if: ▪ More than ten reels are used in 8 hours. ▪ You see no particles on the tape head after each reel of tape. ▪ You do not suspect abnormal dust in the room where the drive is located.
HEAVY	Clean the tape thoroughly AFTER EACH REEL of tape if: ▪ Particles appear on the tape head after each reel of tape. ▪ You are reading interchange tapes from outside your clean room. ▪ You are using new or seldom-used tapes (new tapes usually contain debris, generated during their manufacture, from the slitting process).
SPECIAL	Clean the tape path ANYTIME you suspect abnormal dust in the room where the drive is located. This condition may be caused by custodial activity, equipment moves, or supply delivery. Also, clean the tape path if the drive has not been used for several days.

Use the following materials to clean the tape path:

- **Cleaning Solvent**

Use only high-quality electronic-grade isopropyl alcohol of at least 90 percent concentration. The isopropyl mixture must consist of alcohol and distilled water only.

- **Nonabrasive, Lint-Free Cloths, and/or Swabs**

Follow these precautions:

- Do not use cleaner solutions that contain lubricants. Lubricants leave a deposit on the tape head and impair performance.
- Do not use alcohol cleaning solutions on the rubber gripping fingers on the take-up reel.
- Do not use aerosol cleaners. The spray is difficult to control and often contains metallic particles that can damage the tape head.
- Do not use soap and water on the tape path. Soap leaves a thick film, and water may damage electronic parts.
- Discard used cloths and swabs after use. Even if they appear clean, they are contaminated.
- Do not use facial tissues. Although they may seem effective, they leave highly abrasive lint in the tape path.

To clean the KS-23909, L21 9-track tape drive path, use the general guidelines in the following procedure:

 NOTE:

Refer to 254-303-101, *Routine Maintenance Tasks*, for the procedure to clean KS-23909, L21 tape drives.

Step	Procedure
1	Pour a small amount of solvent into a clean container, such as a small UNWAXED paper cup, or if the cleaner comes in a squeeze bottle, squeeze a small amount on a lint-free cloth or swab.  CAUTION: <i>Alcohol dissolves wax. If you use a waxed cup, the wax transfers to the tape path. DO NOT dip cloths or swabs into the cleaner container or touch the cloths or swabs to the lip of the open container during pouring. This will contaminate the solvent.</i>

Step	Procedure
2	Use a lint-free swab or cloth and apply gentle pressure in one direction to clean the following surfaces. Refer to Figure 19-2 for the locations to be cleaned (Points 1 through 8). ▪ Read/Write/Erase Head (if swab appears dirty, repeat with a new swab) (Point 1) ▪ Tape Cleaner Block (if swab appears dirty, repeat with a new swab) (Point 2) ▪ Buffer Arm Movable Roller (Point 3) ▪ Buffer Arm Fixed Guide (Point 4) ▪ Speed Encoder (Point 5) ▪ Inner diameter (rubber) of the take-up reel (this specially treated surface, if kept clean, greatly eases the manual threading action required to attach the tape to the take-up reel) (Point 6).
3	Use a lint-free wipe to brush out debris below the take-up reel and speed encoder (from Point 7 to Point 7 at the bottom of the figure). Brush off the tape-in path sensor to the right of the buffer arm (the other Point 7 next to Point 3 and 4).
4	Periodically check and wipe off rubber grip pads on the supply reel hub (Point 8). Use a dry or damp swab or cloth. DO NOT USE ISOPROPYL ALCOHOL TO CLEAN THESE FINGERS.

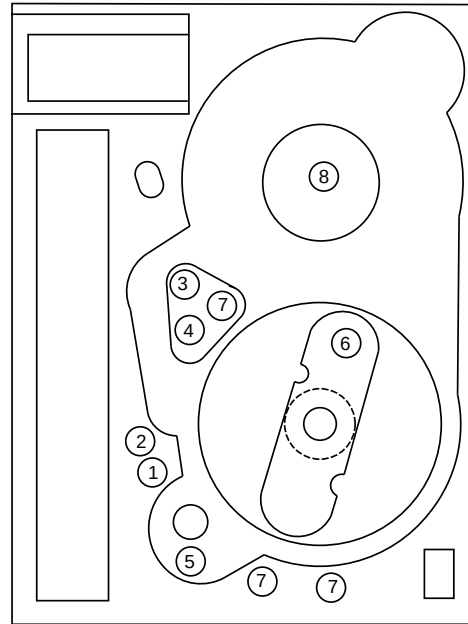


Figure 19-2. Cleaning Points

**Cleaning UN376 (NCR006-3503341),
UN376C (NCR006-3300608/Comcode 407545243),
or
UN376E (KS-24367, L1/Comcode 407771260)
DAT Drives**

The tape heads should be cleaned after every 25 hours of operation. The NCR006-3503341 DAT tape heads should also be cleaned if “caution” is signaled by the Cassette Indicator (green) and Drive Indicator (alternately flashing green and amber).

The NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 tape heads should also be cleaned if “caution” is signaled by the Cassette Indicator (any state) and Drive Indicator (pulse amber). Cleaning is also required for the NCR006-3300608 and KS-24367, L1 when the Cassette Indicator is green and the Drive Indicator alternately flashes green and amber. Refer to Figure 19-3 for Cassette Indicator and Drive Indicator locations.

A “cleaning” cassette (Hewlett-Packard* P/N 92283K or equivalent) is used to manually clean the drive. To clean the drive, insert the “cleaning” cassette in the drive. The drive will automatically load the cassette, clean the heads, and eject the cassette when the operation is complete. The date should be recorded on the “cleaning” cassette label to maintain a history of use. After 25 uses, discard the “cleaning” cassette.



CAUTION:

Cleaning cassettes are considerably more abrasive to the recording heads of the DAT drive than standard data cassettes. Usage should be kept within recommended limits.

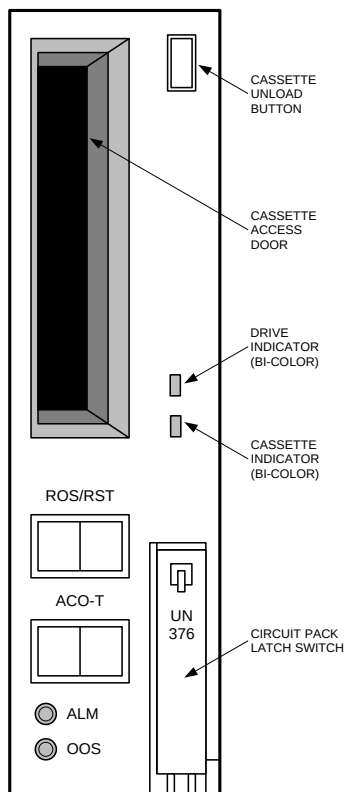


Figure 19-3. UN376 (NCR006-3503341), UN376C (NCR006-3300608/Comcode 407545243), and UN376E (KS-24367, L1/Comcode 407771260) SCSI DAT Drives — Front View

*

Hewlett-Packard is a registered trademark of Hewlett-Packard Company.

Setting Device ID

The device ID setting procedures for the SCSI tape drives are as follows. Device IDs should only be set to values between 0 and 6. The device ID number 7 is reserved for and used by the SCSI HA on the SCSI bus. No two devices can have the same device ID on the same SCSI bus.

Tape drive device IDs should be set numerically higher than those of the disk drives on the same SCSI bus to give higher priority to slower tape jobs.

Device IDs of the 9-track SCSI tape drives are set via the front panel controls of the drives with the following procedure.

Setting Device IDs for KS-23909, L10 9-Track Tapes

The initial system display is Ld Point.

Step	Procedure
1	Press the RESET key. Display: BOT.
2	Press the DIAG key. Display: Test 00.
3	Press the TENS and UNITS keys to display. Display: Test 67.
4	Press the RESET key. Display: Opt 01.
5	Press the TENS and UNITS keys to display. Display: Opt 15.
6	Press the DIAG key. Display: Bi 3E . (06 = Option Byte 15's default setting for SCSI ID 6 is "3E").

➤ NOTE:

The SCSI ID setting is determined by option byte 15, bits 2, 1, and 0. Refer to Table 19-3 for clarification of byte 15 option bit settings.

Step	Procedure
7	Press the TENS and UNITS keys to set ID to desired number. Display: Bi 3D . (04 = new ID setting).
8	Press the RESET key. Display: Ha lt 67.
9	Press the DIAG key. Display: BOT.
10	<i>POWER CYCLE.</i>
	 NOTE: Option 5 must be set to 0.

Table 19-3. Description of Configuration Option Byte 15 Bit Settings

Option Byte No.*	Bit No.	Interpretation
15	7	1 = File mark status reported read, space, or write 0 = File mark status reported read or space only (Lucent Technologies setting = 0)
	6	1 = Tape halted and "check condition" upon read into EOT (sticker) 0 = EOT ignored when reading (Lucent Technologies setting = 0)
	5	1 = Irrecoverable write errors finish with head on BOT side of attempt 0 = Irrecoverable write errors finish with head on EOT side of attempt (Lucent Technologies setting = 1)
	4	Code compatibility (with bits 2, 1, and 0 of byte 14) (Lucent Technologies setting = 1)
	3	1 = Buffered write EOM flagged following buffer dump in EW area 0 = Buffered write EOM flagged when physical EOT detected (Lucent Technologies setting = 1)
	2	SCSI ID Address Setting: 1 = SCSI address bit set (MSB) 0 = not set (Lucent Technologies default setting = 1)
	1	1 = SCSI address bit set 0 = not set (Lucent Technologies default setting = 1)
	0	1 = SCSI address bit set (LSB) 0 = not set (Lucent Technologies default setting = 1)

* Byte 15 default setting for SCSI ID 6 is "3E."

BOT = Beginning of Tape

EOM = End of Medium

EOT = End of Tape

EW = Early Warning

LSB = Least Significant Bit

MSB = Most Significant Bit

Setting Device IDs for KS-23909, L21 9-Track Tapes

The initial system display is ONLINE and BOT.

Step	Procedure
1	Press the ONLINE key to take the drive off-line. Display: BOT.
2	Press the OPTION key. Display: TEST *.
3	Press the NEXT key until the ID * appears in the display. Display: ID *.
4	Press the ENTER key. Display: ID # (current ID number).
5	Press the NEXT key or PREV key until desired ID number appears in display. Display: ID # (desired ID number).
6	Press the ENTER key. Display: SET<ID #> (stays on for 20 seconds), then ID *.
7	Press the OPTION or RESET key to exit the Option Mode. Display: BOT.
8	<i>POWER CYCLE.</i>

Setting Device IDs for DAT Drives

The device ID of the DAT drive is set using the three SCSI device ID switches located on the UN376 circuit pack. Refer to Figure 19-4, Figure 19-5, and Table 19-4 for the UN376 circuit pack ID switch locations and switch settings.

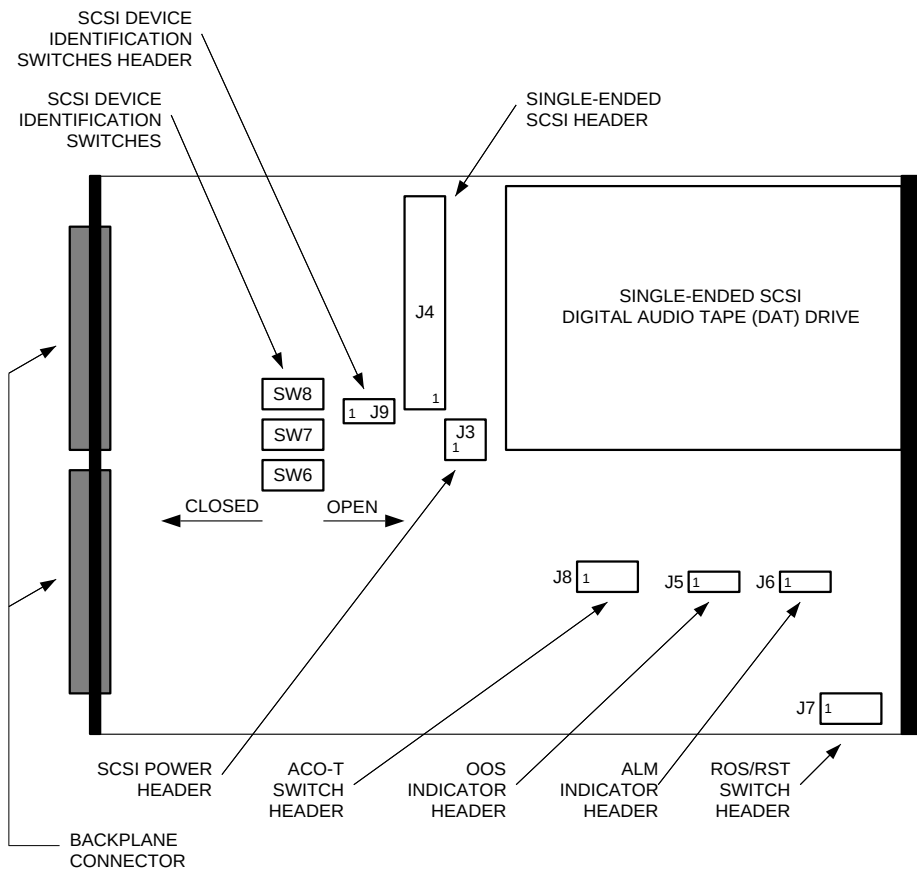


Figure 19-4. UN376 and UN376B Series 1 and 2 Circuit Pack Layout

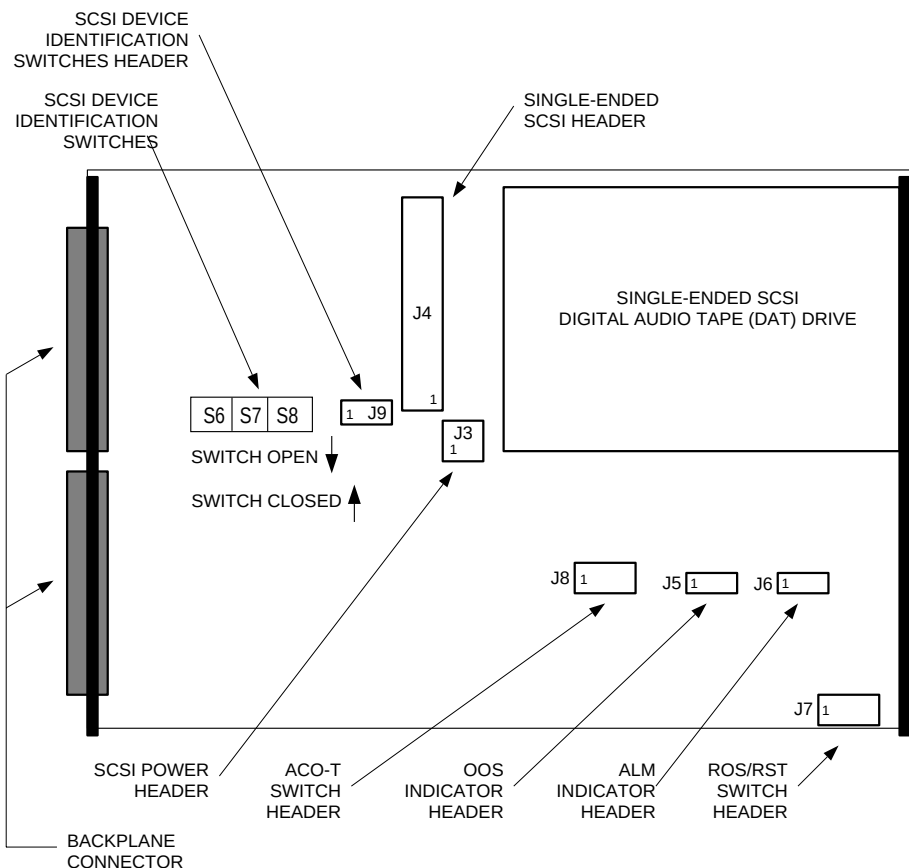


Figure 19-5. UN376B Series 3 and Later Circuit Pack Layout

Table 19-4. DAT SCSI Device Identification Switch Settings

SCSI ID*	SW8	SW7	SW6
0	Open (0)	Open (0)	Open (0)
1	Open (0)	Open (0)	Closed (1)
2	Open (0)	Closed (1)	Open (0)
3	Open (0)	Closed (1)	Closed (1)
4	Closed (1)	Open (0)	Open (0)
5	Closed (1)	Open (0)	Closed (1)
6	Closed (1)	Closed (1)	Open (0)

* SCSI device ID number 7 is not allowed.

Write Protection

9-Track Drive Write Protection

To enable the write operations to the 9-track drive media, “write-enable” rings are installed on the back side of a tape reel. Data can be written to the media only when this ring is in place. To protect data from being overwritten, the “write-enable” ring should be removed. Then, data can be read from the tape without the ability to write data to the tape.

DAT Drive Write Protection

The DAT drive cassette tape has a “write-protect” feature to prevent recording over data on tape. Protection is invoked by sliding the “write-protect tab” located on the rear of the cassette to open the window (Figure 19-6). A sensor in the drive detects the position of the tab and will not allow writing to the tape if the tab is in the “open” position. A cassette tape is write-enabled when the hole is closed.

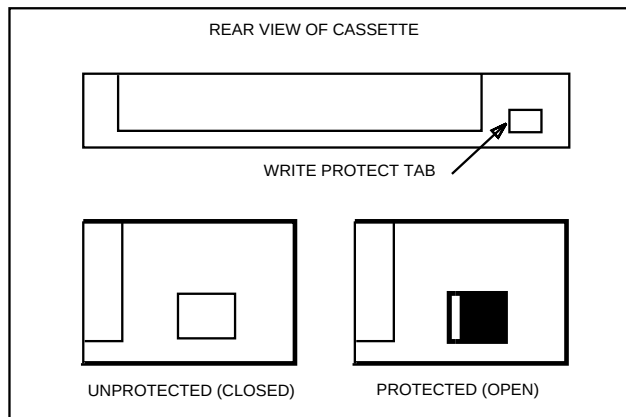


Figure 19-6. Cassette Write Protect

Tape Capacity/Tape Density

9-Track Drive Tape Capacity

The 9-track tape drives accept 6.25-, 7-, 8.5-, and 10.5-inch tape reels of 0.5-inch magnetic tape. Table 19-5 summarizes the data storage capacities available with the various sizes of reels and recording methods.

Table 19-5. KS-23909 9-Track Tape Capacity

Reel Size (inches)	Reel Capacity (feet)	Approximate Storage Capacity*	
		1600 BPI (64 kByte Blocking)	6250 BPI (256 kByte Blocking)
10.50	2400	45 MBs	178 MBs
8.50	1200	22 MBs	89 MBs
7.00	600	11 MBs	44 MBs
6.25	300	5 MBs	22 MBs

* A megabyte (MB) is 2^{20} bytes.

9-Track Drive Tape Density

The 3B20D/3B21D computers SCSI subsystem supports the following two 9-track tape densities:

- 1600 Bytes Per Inch (BPI) with phase-encoded recording.
- 6250 BPI with group-coded recording.

The particular density with which the tape is going to be read from, or written to, is specified in the tape drive's special device filename. (Refer to Chapter 20, "Magnetic Tape Special Device Filenames.")

DAT Drive Tape Capacity

The DAT drive cassette sizes and their storage capacities are given in Table 19-6.

Table 19-6. DAT Cartridge Tape Capacity

Cartridge Size	Mode	Capacity*
30 meters	noncompressed	650 MB
60 meters	noncompressed	1300 MB
90 meters	noncompressed	2000 MB

* A megabyte (MB) is 2^{20} bytes.

⇒ NOTE:

DAT Cartridge Tapes with 120 meter cartridge size are not supported.

DAT Drive Tape Density

The DAT drives currently can support only one tape density. However, the DFC driver is designed to work with the DAT drive special filenames either with high or low density. (Refer to Chapter 20, "Magnetic Tape Special Device Filenames.")

Loading/Unloading 9-Track and DAT Tapes

Loading or Unloading 9-Track Tapes



CAUTION:

During the loading/unloading operations, care should be exercised when handling the tape reels. If reels are gripped in any way that presses the flanges together, there is a possibility of damaging the edge of the tape.

The hub is the strongest, least flexible portion of the reel. The tape reel should always be held by the hub or as close to the hub as possible.

Loading the KS-23909, L10 9-Track Tape Reel (Auto Load)

To load the KS-23909, L10 9-track tape when there is no tape reel in the loading chamber, use the following procedure; otherwise, use the "Removing the KS-23909, L10 9-Track Tape Reel with Power On" procedure.

Should the tape loading malfunction, refer to the vendor's manual (KS-23909, L10) for possible display indications.

Step	Procedure
1	When the drive indicates OK (flashes momentarily) at the end of a power-on sequence, open the loading door to place the reel on the supply hub. Wait for the OK in case an internal fault prohibits further use. The display now should be TO LOAD TAPE PRESS LD/ONL.
2	Ensure that the write-enable ring is either fitted to permit writing or removed to prevent overwriting of existing data.
3	Ensure the tape end is free. For new reels, remove the adhesive strip and/or rubber block which constrains the free end to the tape back.
4	Place the tape reel on the self-centering supply hub with the "write enable" ring towards the casting.
5	Close the loading door and press LD/ONL . The display should indicate LOCATING, quickly followed by LOADING. If DOOR is indicated, check that the loading door is securely closed. The drive firmware automatically checks that the reel is not inverted, threads the tape to the take-up spool, and searches for BOT without any further operator action. The load sequence is complete when LD point is indicated.

Loading (Lacing) the KS-23909, L10 9-Track Tape by Hand (Manual)

Since the KS-23909, L10 tape drive is designed for fully automatic threading, it is not normally necessary or desirable to manually lace the tape except under fault conditions, such as a failed loading fan. To manually thread the tape drive, use the following procedure:

Step	Procedure
1	Set the power switch to OFF.
2	Open the loading door, and place the tape reel on the supply hub. Push the hub to operate the clamp.
3	Lace the tape through to the take-up hub. Wind on two or more turns until the tape is pulled through when the take-up reel is rotated.
4	Close the loading door, and set the power on. The loading firmware senses that tape is already laced and proceeds to tension the tape and search for BOT without further operator action.

Removing the KS-23909, L10 9-Track Tape Reel with Power On

To unload the KS-23909, L10 9-track tape, use the following procedure:

⇒ NOTE 1:

If the unthreading sequence is commenced inadvertently, it can be cancelled by first pressing the **RESET** key and then pressing the **LD/ONL** key to resume threading. (In order to resume, the tape will probably move to BOT.)

⇒ NOTE 2:

For tape protection, place the reel of tape in a dust-proof cover and store in a safe place.

Step	Procedure
1	To avoid loss of data, check that the 3B20D/3B21D Host computer is not accessing the drive.
2	Set the drive off-line by pressing RESET . If in diagnostic mode, press DIAG control once to return to off-line status (that is, the DIAG legend not illuminated).
3	If OFFLINE is displayed, press RESET while RWD/UNL is pressed. The tape rewinds (with REW/UNLD indicated) to BOT.
4	When the display indicates READY , remove the tape reel by opening the loading door and pressing the center portion of the supply hub. (Refer to Steps 3 through 5 of the "Removing the KS-23909, L10 9-Track Tape Reel During a Power Failure" section.)

Removing the KS-23909, L10 9-Track Tape Reel During a Power Failure

If power to the drive is cut off and it is essential that the reel of tape is recovered without waiting for power to be restored, the following procedure enables the user to do so. The user should be aware that the time required to rewind the tape increases with the amount of the tape on the take-up (lower) spool.

Step	Procedure
1	Set the power switch to OFF. This is essential for safety reasons.
2	Open the loading door, and rotate the supply reel counterclockwise until all the tape has been rewound.
3	Press the hub button.
4	Lift off the unclamped supply reel.
5	Close the loading door.

Loading the KS-23909, L21 9-Track Tape Reel

To load the tape reel, use the following procedure:

Step	Procedure
1	Make sure that the drive is OFFLINE. The door cannot be opened unless the drive is OFFLINE. If the drive is ONLINE, press the ONLINE key once so that ONLINE is no longer displayed.
2	Open the tape drive door.
3	Check for tape "write-enable" capability, as desired.
4	Mount the tape (with the free end to the right) on the top (supply) hub by pushing it over the rollers.
5	Thread the tape through the tape head mechanism, and wind the tape around the lower (take-up) hub (Figure 19-7).
6	Close the tape drive door. Closing the door initiates the next step in the tape loading sequence. LOADING will appear in the display. The following will be displayed in sequence: a. Density of the tape (1600, 6250, BLANK, or UNKNOWN) b. BOT (Beginning of Tape) c. ONLINE — the drive is now ONLINE and the door is locked. Opening the door can be done only after the drive is taken OFFLINE.

⇒ NOTE:

TAPE LOAD FAILURE: If the tape fails to load and NO BOT is displayed, refer to the vendor's manual (KS-23909, L21).

When the drive goes ONLINE, the displays and indicators are as follows:

- The ONLINE indicator remains displayed.
- The TAPE DENSITY INDICATOR remains displayed and shows the current selection for write density. (This will be overwritten by the DFC driver software when the device is addressed by the driver.)
- The WRITE-ENABLE STATUS INDICATOR remains displayed and shows the write enable/disable status of the tape.
- The message in the display generally corresponds to the command the 3B20D/3B21D Host computer is currently sending to the tape drive.

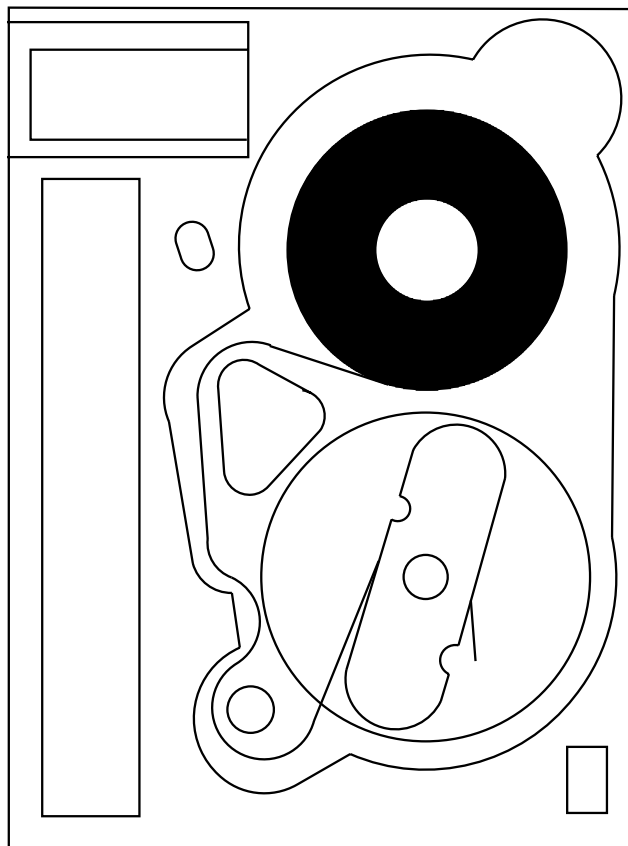


Figure 19-7. Tape Load Diagram

Unloading the KS-23909, L21 9-Track Tape Reel

To unload the tape reel, use the following procedure:

Step	Procedure
1	Take the drive OFFLINE by pressing the ONLINE key. When the drive is OFFLINE, ONLINE will no longer be displayed.
2	Press the REW/UNL key. The drive rewinds the tape. When this activity is completed, NO TAPE is displayed. ⇒ NOTE: It is important that the rewind completes and that NO TAPE is displayed before the door is opened. If not, the supply hub fingers may remain in the locked position and manual unlocking will be required. To manually retract the supply hub fingers, push and hold the tape LOCK/UNLOCK lever and turn the supply hub counterclockwise to retract the hub fingers.
3	When the rewind activity is completed, the front door can be opened and the tape can be removed from the supply hub.

Manually Rewinding the KS-23909, L21 9-Track Tape Reel

To manually rewind the tape reel, use the following procedure:

Step	Procedure
1	While pushing up on the supply hub lock actuator, rotate the supply hub in a counterclockwise direction, approximately one-quarter turn. This will release the rollers that hold the reel on the hub.
2	Manually rewind the tape onto the supply hub, as required.
3	Remove the tape reel from the supply hub.

**Loading or Unloading DAT Drives
(NCR006-3503341,
NCR006-3300608/Comcode 407545243, or
KS-24367, L1/Comcode 407771260)**

The DAT drive cassette is loaded in the drive by inserting the cassette with the label facing up in the loading slot on the UN376 circuit pack front panel. (Refer to 254-303-105, *Hardware Reference Manual*.) If the drive has power applied, the cassette will be drawn into the mechanism and positioned for use.



CAUTION:

Do not force the cassette into the drive. The cassette should be inserted gently into the drive. Very little force is required to cause the drive to grab the cassette, after which the drive will pull the cassette in. Excessive force in insertion of the cassette can damage the drive mechanism components.

The cassette is unloaded by pressing the **EJECT** button on the front panel of the UN376 circuit pack.

Loading the DAT Cartridge Tape/Cassette

Insert the cartridge tape (cassette) into the drive until the drive takes hold of the cartridge. The cartridge tape loading sequence is as follows:

Step	Procedure
1	The drive threads the tape and rewinds the tape to the Beginning of Media (BOM). The tape is then moved to the Beginning of Partition (BOP) for Partition 0. The reference area is then checked to determine the tape format (DDS, audio, and so forth).
2	If the tape is blank, the drive leaves the tape at the BOP for Partition 0 and waits for the next command.

Step	Procedure
3	NCR006-3503341 DAT — If the error rate is high, CAUTION is signaled on the front panel of the UN376 circuit pack by lighting the Cassette Indicator (bottom) green and alternately flashing the Drive Indicator (top) green and amber.
4	NCR006-3300608 and KS-24367, L1 DAT — If the error rate is high, CAUTION is signaled on the front panel of the UN376C/UN376E circuit pack by pulsing the Drive Indicator (top) amber. The Cassette Indicator (bottom) may be in any state while the Drive Indicator pulse is amber.
5	The system area on the tape is then accessed and the tape log is read into the drive.
6	The drive rewinds to the BOP and goes on-line.
	 NOTE: If the NCR006-3503341 DAT drive is power-cycled while a cassette is loaded, the drive performs a load sequence and goes on-line.

Ejecting the DAT Cassette

The cassette tape can be removed from the drive by pressing the  button. The unload sequence is as follows:

Step	Procedure
1	The tape is rewound to the BOP for Partition 0.
2	If the tape is write-enabled, a copy of the tape log held in the drive RAM is written back to the tape.
3	The tape is rewound to the BOM, unthreaded from the drive mechanism, and ejected.

DAT Drive Cartridge Tapes and Status Indicators

The DAT unit uses DDS grade 4 millimeter wide tape in 90-, 60-, and 30-meter cartridges (cassettes). The 120M DDS-2 cartridges are not supported by either the NCR006-3503341, the NCR006-3300608/Comcode 407545243, or the KS-24367, L1/Comcode 407771260 DAT units and cannot be used. Use of 90M cartridge is recommended.

The recommended maximum use for a tape is 2000 passes over any particular area of the tape. This includes retries. The recommended maximum number of times a tape should be loaded into a drive is 100. Data should be transferred to a new tape and the old tape discarded if either of these “useful life” thresholds are reached.

Worn tapes and tapes at or near end of “useful life” may cause excessive read-after-write error recovery and high error correction counts to occur. The following are indicators that a tape is worn or has exceeded one of the “useful life” thresholds:

NCR006-3503341 DAT

- The “Media Wear Caution” signal when displayed indicates that the tape is worn or cleaning may be needed. “Media Wear Caution” is signaled by the Cassette Indicator Light Emitting Diode (LED) (green) and Drive Indicator LED (alternately flashing green and amber). (Refer to Table 19-7 for DAT Drive Indicators.)

NCR006-3300608/Comcode 407545243 or KS-24367, L1/Comcode 407771260 DAT,

- The “Cleaning Needed” signal LED does not extinguish after cleaning. “Cleaning Needed” is signaled by the Cassette Indicator LED (any color of green that could be steady, flashing, or pulsing) and Drive Indicator LED (pulse amber). (Refer to Table 19-7 for DAT Drive Indicators.)
- The “Cleaning Needed” signal LED extinguishes after cleaning but reappears following the next tape access. “Cleaning Needed” is signaled by the Cassette Indicator LED (any color of green) and Drive Indicator LED (pulse amber). (Refer to Table 19-7 for DAT Drive Indicators.)
- The “Caution” signal is displayed. “Caution” is signaled by the Cassette Indicator LED (green) and Drive Indicator LED (alternately flashing green and amber). (Refer to Table 19-7 for DAT Drive Indicators.)

**NCR006-3300608/Comcode 407545243 and
KS-24367, L1/Comcode 407771260
DAT Drive Status Indicators**

Table 19-7 shows the Drive Indicators for the NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT units.

Table 19-7. NCR006-3300608/Comcode 407545243 and KS-24367, L1/Comcode 407771260 DAT Drive Indicators

Cassette Indicator	Drive Indicator	State	Description
Off	Off	Not Applicable	(1) Power is off. (2) Power is on and the cassette is not present.
Pulse Green	Off	Read/Write	Cassette is in the process of unloading or loading, or Self-Test is in progress.
Green	Off	Read/Write	Cassette is loaded and is on-line.
Flash Green	Off	Read/Write	Cassette is loaded and is being accessed.
Pulse Green	Off	Write-Protect	Cassette is write-protected and is in the process of unloading or loading.
Green	Off	Write-Protect	Cassette is write-protected, loaded, and on-line.
Flash Green	Off	Write-Protect	Cassette is write-protected, loaded, and being accessed. (Read only.)
Green	Alternately Flashing Green and Amber	Error	Caution Signal. If triggered by the wellness check, insert a cleaning cartridge and repeat the test with a fresh tape. If the Caution Signal persists, fail the drive. If the Caution Signal is cleared when a fresh tape is used, the original tape is faulty. Discard the original faulty tape.
Any	Pulse Amber	Error	Cleaning is needed or cassette is near end of useful life.
Any	Amber	Error	Hard fault.
Flash Green	Off	Cleaning	Cleaning cartridge is process of loading or unloading, or cleaning.

Legend: Flash = $\frac{1}{4}$ second on, $\frac{1}{4}$ second off
Pulse = $\frac{1}{2}$ second, $\frac{1}{2}$ second off

NCR006-3503341 SCSI DAT Drive Status Indicators

Table 19-8 shows the Drive Indicators for the NCR006-3503341, L1 DAT.

Table 19-8. NCR006-3503341, L1 3.5-Inch SCSI DAT Drive Indicators

Cassette Indicator	Drive Indicator	State	Description
Off	Off	Not Applicable	Power is off, or power is on and the cassette is not present.
Flashing Green	Flashing Green	Read/Write	Cassette is in the process of unloading or loading.
Green	Green	Read/Write	Cassette is loading and is on-line.
Green	Flashing Green	Read/Write	Cassette is loaded and is being accessed.
Green	Off	Read/Write	Cassette is loaded and is off-line. Note that taking the UN376 circuit pack OOS does NOT take the DAT drive on the circuit pack off-line.
Flashing Amber	Flashing Green	Write-Protect	Cassette is write-protected and is in the process of unloading or loading.
Amber	Green	Write-Protect	Cassette is write-protected, loaded, and on-line.
Amber	Flashing Green	Write-Protect	Cassette is write-protected, loaded, and being accessed.
Amber	Off	Write-Protect	Cassette is write-protected, loaded, and off-line.
Green	Alternately Flashing Green and Amber	Error	Media Wear Caution Signal. An excessive number of read-after-writes or an excessive number of errors corrected by third-level error correction have been detected.
Amber	Amber	Error	High Humidity Warning. All commands are aborted and the tape is unthreaded to prevent damage to the tape and head. This state is also used to indicate no terminating resistor on the SCSI bus.
Flashing Amber	Flashing Amber	Error	Normal self-test diagnostic sequence.
Flashing Amber	Amber	Error	Self-test diagnostic failure.

Magnetic Tape Special Device Filenames

20

Contents

Introduction	20-1
Tape Device <i>logdev</i> Records	20-1
IOP Magnetic Tape <i>logdev</i> Records (3B20D Computer Only)	20-2
SCSI/IOP Backward Compatible <i>logdev</i> Records	20-4
SCSI <i>logdev</i> Records	20-5

Introduction

The following sections describe the magnetic tape special device filenames:

- Tape Device *logdev* Records
- IOP Magnetic Tape *logdev* Records (3B20D computer only)
- SCSI/IOP Backward Compatible *logdev* Records
- SCSI *logdev* Records.

Tape Device *logdev* Records

The *logdev* records are used to associate logical device names with real (physical) devices. A user specifies a logical device name to access the tape device hardware. For tape units, the *logdev* record also specifies parameters to be used during the tape operation. A particular *logdev* record is identified by its name, specified in the *name* field of the *logdev* form.

IOP Magnetic Tape *logdev* Records (3B20D Computer Only)

The *logdev* record name for the current 3B20D computer Input/Output Processor (IOP) tape units is of the following form:

/dev/mtnc

Where:

- n* Specifies the IOP 9-track tape unit number.
- c* Specifies characteristics (tape density and rewind on close).

The *logdev* record *partition* field controls the options of the IOP tape. Figure 20-1 shows the bit field layout of the current IOP tape partition number.

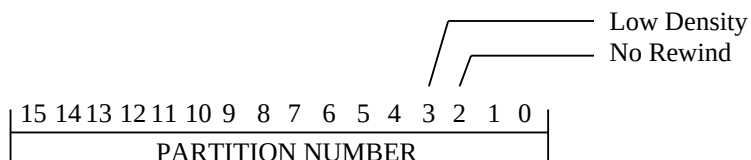


Figure 20-1. IOP Tape Device Partition Number Descriptor

Table 20-1 describes the existing IOP 9-track operating parameters selectable by a given logical device name. Note that the tape parameters are specified in the *partition* field of the *logdev* record and is shown in decimal and binary forms.

Table 20-1. IOP 9-Track Operating Parameters

Logical Name	Unit	Partition	Binary	Parameters
/dev/mt00	0	0	0000	Hi Density (6250 bpi), Rewind
/dev/mt04	0	4	0100	Hi Density (6250 bpi), No rewind
/dev/mt08	0	8	1000	Low Density (1600 bpi), Rewind
/dev/mt0C	0	12	1100	Low Density (1600 bpi), No rewind
/dev/mt1C	1	12	1100	Low Density (1600 bpi), No rewind

Note that the IOP 9-track tape implementation only supports variable size tape records and that the IOP tape hardware (latest version) limits tape records to a maximum size of 6144 bytes. An earlier version of IOP tape hardware limited tape records to a maximum size of 2048 bytes. These constraints mean that the

largest size record that can be written to or read from tape is 6 K bytes using the latest version of the IOP tape hardware. Using earlier IOP tape hardware, the upper limit is 2 K bytes.

Tapes written at the maximum record size on the later IOP tape drive (6 K max) cannot be read on the earlier IOP tape drive (2 K max). What this means is that the current logical device names do not limit the tape record size and do not prevent writing backward incompatible tapes. Since backward compatibility is not guaranteed among various vintages of the IOP 9-track tape units, it will not be guaranteed by this feature either. The Small Computer System Interface (SCSI) 9-track tape drives will be able to write tapes that will not be readable on the IOP tape drives.

SCSI/IOP Backward Compatible *logdev* Records

By expanding the number of control fields within the *partition* field of the *logdev* form, a logical device record can be defined that suffices for both the IOP and SCSI tape devices. This redefinition of the tape device *partition* field is shown in Figure 20-2.

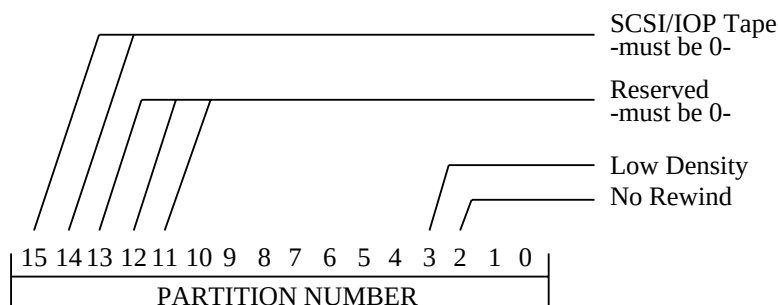


Figure 20-2. SCSI/IOP Tape Device Partition Number Descriptor

This layout of the *partition* field becomes effective with the SCSI tape units. In this layout, bits 15 and 14 identify how bits 10 through 0 are interpreted. Bits 13, 12, and 11 are reserved for future use.

This form of the *partition* field is an extension of the IOP only *logdev* record and specifies the same operating characteristics as the previously shown IOP *logdev* record. The *logdev* name will be the same as that used for the IOP only record.

To restate the IOP and SCSI/IOP *logdev* characteristics:

- No rewind/rewind
- Low density/high density
- Variable length record sizes.

The IOP tape hardware limits the maximum record size to 6 K bytes (2 K bytes on earlier hardware). The SCSI tape hardware can support record sizes up to the 3B20D computer limit of 128 K bytes (1 memory segment).

For both the 2 K and 6 K bytes versions, it is possible to write tapes on the SCSI 9-track tape units that can be read by the IOP 9-track tape units; it is also possible to read tapes on the SCSI 9-track tape units that were written by the IOP 9-track tape units. By choosing the appropriate SCSI/IOP logical device name, tape operating characteristics can be selected that will be compatible between SCSI and IOP 9-track units except for record length. The user (user

process) must limit its Input/Output (IO) requests to the tape drive to the maximum size supported by the IOP tape hardware with which it is to be compatible.

SCSI *logdev* Records

⇒ NOTE:

This section applies to both SCSI 9-track and SCSI Digital Audio Tape (DAT) units.

Because of the compatibility requirements concerning IOP 9-track tape, most of the discussion centers around the 9-track implementation.

New operating characteristics, supported by the SCSI tape units, cannot be specified using the IOP (IOP/SCSI) *logdev* partition number layout and will require a different layout of the *partition* field in the *logdev* records. Table 20-2 describes the new SCSI tape *logdev* information. The *logdev* name for the SCSI tape units is of the following form:

ldev/mt n s d b [r]

Where:

- n* Specifies the SCSI tape unit number.
- s* Specifies how the tape blocking factor will be handled.
 - f* Indicates tape records will be fixed size (when writing tape, records will be padded with 0x00 by the DFC, as necessary).
 - v* Indicates tape records will be variable size.
- d* Specifies the tape writing density.
 - l* Low density (1600 bpi).
 - h* High density (6250 bpi).

The DAT drives do not support multiple density writing. Therefore, the density specification will be ignored when the tape drive is a DAT unit.

- b* Specifies the tape blocking factor (record size).
 - 0* Maximum record size is limited only by the system/tape hardware (invalid if record sizes are fixed).
 - non-0* Record size at multiples of (512 * *b*) bytes.
Minimum record size if record sizes are fixed.
Maximum record size if record sizes are variable.

r Specifies the rewind/no rewind upon file close option.

Opened for writing:

- null Indicates that upon device file close, two file marks will be written on tape and the tape rewound.
- n* Indicates that upon device file close, two file marks will be written on tape and the tape positioned in-between the two file marks.

Opened for reading:

- null Indicates that upon device file close, tape will be rewound to Beginning of Tape (BOT).
- n* Indicates that upon device file close, the tape will be forward spaced to the End of Tape (EOT) side of the nearest file mark, unless already there by virtue of reading the tape prior to the device close.

IMPLEMENTOR'S NOTE:

This means that an OPEN and CLOSE sequence with no reads will advance the tape by 1 file mark.

The *logdev* record *partition* field will control the options of the SCSI tape. Figure 20-3 shows the bit field layout of the SCSI tape partition number. Table 20-2 shows typical *logdev* values.

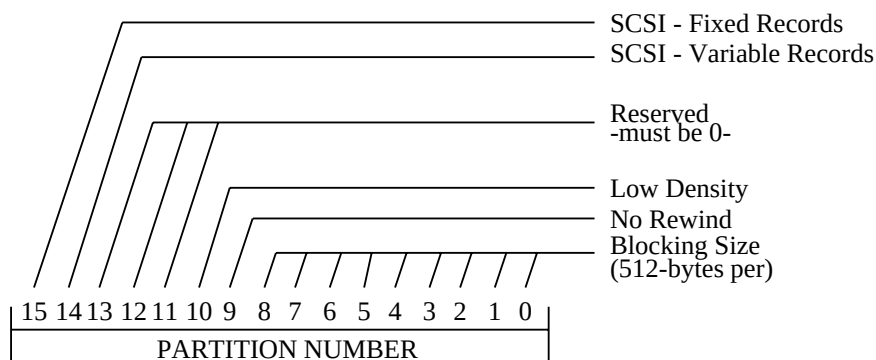


Figure 20-3. SCSI Tape Device Partition Number Descriptor

Table 20-2. Typical *logdev* Values

Device Name	Device	Binary Partition Number		Options
<i>/dev/mtnsdbr</i>	n	ss00 0drb bbbb bbbb	HEX	
<i>/dev/mt0vl0</i>	0	0100 0100 0000 0000	4400	SCSI, 1600 bpi, rewind, variable record size
<i>/dev/mt0vh0</i>	0	0100 0000 0000 0000	4000	SCSI, 6250 bpi, rewind, variable record size
<i>/dev/mt0fh4n</i>	0	1000 0010 0000 0100	8204	SCSI, 6250 bpi, no rewind, 2 K fixed record size
<i>/dev/mt3fh4n</i>	3	1000 0010 0000 0100	8204	SCSI, 6250 bpi, no rewind, 2 K fixed record size
<i>/dev/mt3fh4</i>	3	1000 0000 0000 0100	8004	SCSI, 6250 bpi, rewind, 2 K fixed record size
<i>/dev/mt0fh256</i>	0	1000 0001 0000 0000	8100	SCSI, 6250 bpi, rewind, 128 K fixed record size
<i>/dev/mt0fl12</i>	0	1000 0100 0000 1100	840C	SCSI, 1600 bpi, rewind, 6 K fixed record size
<i>/dev/mt0vl4</i>	0	0100 0100 0000 0100	4404	SCSI, 1600 bpi, rewind, 2 K-max variable record size
<i>/dev/mt0vl4n</i>	0	0100 0110 0000 0100	4604	SCSI, 1600 bpi, no rewind, 2 K-max variable record size
<i>/dev/mt0vl12</i>	0	0100 0100 0000 1100	440C	SCSI, 1600 bpi, rewind, 6 K-max variable record size
<i>/dev/mt0vl12n</i>	0	0100 0110 0000 1100	460C	SCSI, 1600 bpi, no rewind, 6 K-max variable record size
<i>/dev/mt0vh12</i>	0	0100 0000 0000 1100	400C	SCSI, 6250 bpi, rewind, 6 K-max variable record size
<i>/dev/mt0vh12n</i>	0	0100 0010 0000 1100	420C	SCSI, 6250 bpi, no rewind, 6 K-max variable record size
<i>/dev/mt00</i>	0	0000 0000 0000 0000	0000	SCSI/IOP, 6250 bpi, rewind, variable record size
<i>/dev/mt04</i>	0	0000 0000 0000 0100	0004	SCSI/IOP, 6250 bpi, no rewind, variable record size
<i>/dev/mt08</i>	0	0000 0000 0000 1000	0008	SCSI/IOP, 1600 bpi, rewind, variable record size
<i>/dev/mt0c</i>	0	0000 0000 0000 1100	000C	SCSI/IOP, 1600 bpi, no rewind, variable record size

The following list gives examples of how to read the *logdev* values shown in Table 20-2. Application device filenames may be different than those shown in the examples.

- In order to create a 1600-bpi 9-track tape (with rewind), compatible with the IOP 9-track tape unit, a user must either specify */dev/mt0vl12* or specify */dev/mt08* and not attempt to write more than 6 K bytes in any one tape write.
- In order to create a 6250-bpi 9-track tape (with rewind), compatible with the IOP 9-track tape unit, a user must either specify */dev/mt0vh12* or specify */dev/mt00* and not attempt to write more than 6 K bytes in any one tape write.
- Both SCSI 9-track and SCSI DAT units will work with the IOP tape unit *logdev* names. The SCSI 9-track unit will, in this instance, write/read an IOP compatible tape only if each and every write/read job specifies 6 K bytes or less.
- When the *logdev* name is */dev/mt0vl12* or */dev/mt08*, the SCSI DAT unit will write/read a DAT format tape, using a set of characteristics that will be supported by all 3B21D computer DAT units.
- When the *logdev* name is */dev/mt0vh12* or */dev/mt00*, the SCSI DAT unit will not write/read because these logical names specify *high* density, which is not currently supported by the 3B21D computer DAT units.

The *dcn* field of the *logdev* record is used to specify the device driver for the device file. For IOP tape units, the *dcn* equals 5, indicating */prc/bdf*, the IOP driver. When setting up *logdev* records for SCSI tape units, the *dcn* field will equal 4, indicating */prc/bde*, the DFC driver.

The *logdev* records also identify the *driver type*. For both IOP and SCSI tape units, the driver type will remain "r" to signify *record*.

Contents

Introduction	21-1
Software Update Types	21-2
■ Regular Software Update	21-2
SU Format	21-4
SCCS and MCC Interface	21-6
SU Reception from the SCANS	21-7
SU Reception from the CSCANS	21-8
Verification	21-9
Activation	21-9
Backout	21-10
■ Emergency Fix	21-10
SCCS and MCC Interface	21-11
Activation	21-11
Backout	21-11
Space Reclamation	21-12
■ Reclaim Space in SU Storage Directory	21-12
■ Reclaim Patch Space	21-12
Error Conditions	21-12
Enhanced SU Capabilities	21-14
SU Installation	21-16
SU Maintenance	21-20
Routine Operations and Maintenance Procedures	21-24
■ Schedule Making SUs Official	21-24
■ Install SUs	21-25

Contents

■ Check/Make Official Temporary SUs	21-26
■ Make SUs Official	21-29
■ Back Out SUs	21-32
■ Back Out Last Official SUs	21-33
■ Manual Stop and Start of SU Expansion	21-34
■ Obtain SU Status Report	21-35
■ Request Summary/Long Summary of SUs	21-36
■ Load SUs from SCANS or Tape	21-40
Page 1940 – Easy SU Installation	21-44
■ Page 1940 Description	21-44
■ Page 1940 Fields	21-45
■ Page 1940 Commands	21-46
■ Page 1940 Example Procedure	21-48
Page 1941 – SU Automation Scheduling	21-49
■ Page 1941 Description	21-49
■ Page 1941 Fields	21-50
■ Page 1941 Commands	21-51
Page 1942 – SU Automation Office Profile	21-53
■ Page 1942 Description	21-53
■ Page 1942 Commands	21-54
Page 1943 – SU Automation Healthcheck	21-56
■ Page 1943 Description	21-56
■ Page 1943 Commands	21-57
Page 1950 – SU Maintenance	21-58
■ Page 1950 Description	21-58
■ Page 1950 Commands	21-59
Page 1960 – SU Installation	21-61
■ Page 1960 Description	21-61
■ Page 1960 Commands	21-63

Introduction

Software Update (SU) is the process of activating orderly program changes in the switching equipment software. This chapter describes program changes that are possible within the database. The changes are made to a particular software release and/or software release issue to solve a system problem.

For details concerning the messages and the associated variable fields to implement SUs, refer to the following information products (IPs):

- 254-303-110, *PDS Input Message Manual*
- 254-303-111, *PDS Output Message Manual*
- 254-303-112, *MML Input Message Manual*
- 254-303-113, *MML Output Message Manual*.

Software Update Types

The types of software updates are as follows:

- Regular Software Update
- Emergency Fix.

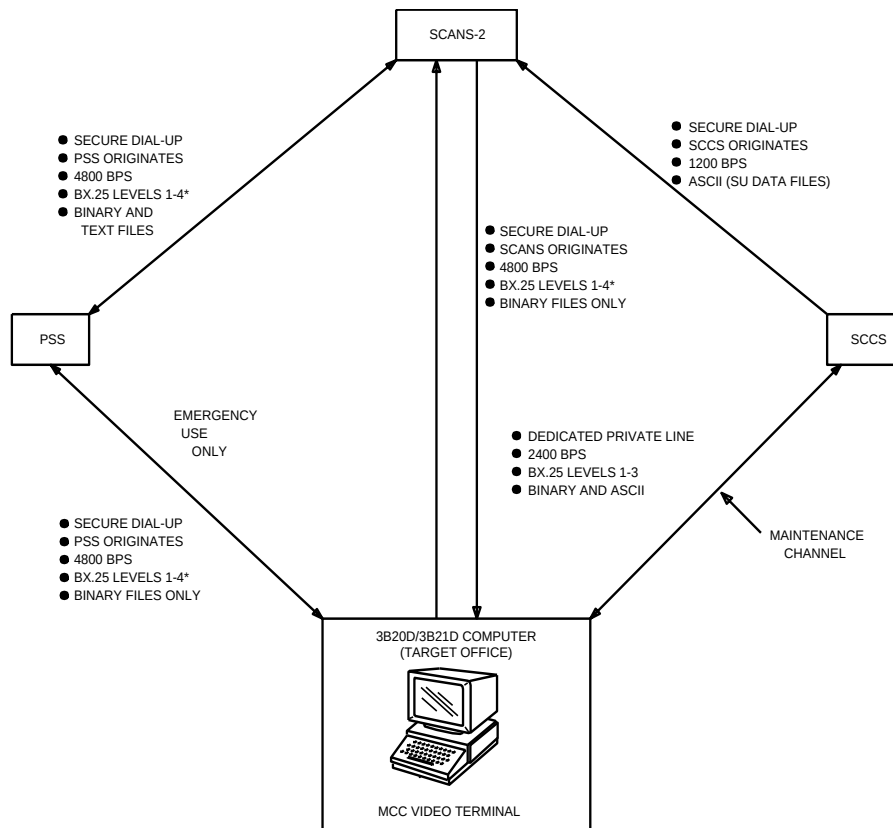
Regular Software Update

In-service offices receive most official software changes in the form of software updates. The SU originates as a fix for a problem within the software release.

Four external interfaces are employed to provide for the generation, distribution, and activation of software updates. Three of the interfaces (Figure 21-1) are as follows:

- **PSS:** The Programmer Support System (PSS) originates software updates. After an SU has been assembled, tested, and approved at the PSS, a software update identification number is assigned. The SU is then transmitted to the Software Change Administration and Notification System (SCANS) for distribution.
- **SCANS:** The SCANS is a time-shared computer system for orderly software update distribution. Maintenance personnel who subscribe to the SCANS may access the system daily to receive and record the software updates. The system also lists any software updates that have been canceled or changed.
- **SCCS:** Using a 1200-baud dial-up terminal, the Switching Control Center System (SCCS) has the capability of remote software update activation. The SCCS accesses the SCANS and triggers the delivery of an SU using the program update subsystem. Then the received SU is remotely activated from the SCCS via the maintenance channel.

The fourth external interface provides for the distribution of software updates only in offices that are equipped with the Customer Service Computer Access Network System (CSCANS) interface. The CSCANS is a Regional Bell Operating Company owned and operated computer system for automated software update distribution. Subscribing offices may access the CSCANS to receive software updates.



* BX.25 LEVEL 4 INCLUDES SESSION LAYER ONLY

Legend:

BPS - Bytes Per Second
MCC - Master Control Center
PSS - Programmer Support System
SCANS - Software Change Administration and Notification System
SCCS - Switching Control Center System
SU - Software Update

Figure 21-1. Software Update Distribution

Under normal operating conditions, the SU distribution point is the SCANS. However, in an emergency (such as a SCANS outage), an SU can be transmitted from the PSS over a data link directly to an office. Maintenance personnel at the SCCS or local office must make a verbal request to the Regional Technical Assistance Center to use the PSS distribution method. The field update coordinator then sets up an emergency data link from the PSS to the switch and manually transmits the SU (after maintenance personnel have primed the computer for reception of the SU files). Procedures for activating the SU are not altered.

The SU activation responsibility between Lucent Technologies and the operating telephone company is as follows:

- During preturnover (new office), retrofit, and restart intervals, the installer is responsible for obtaining and activating all current SUs which apply to that office.
- At all other times in a working office (when not in a retrofit or restart mode), the operating telephone company is responsible for obtaining and activating all applicable SUs.

SU Format

The SU format, illustrated in Figure 21-2, consists of at least four files:

- **SCANS Information File:** The SCANS information file contains information on the SU.
- **Header File:** The header file contains the necessary information for maintaining the integrity of the SU. The information consists of the SU number, software release, sequence number, name, size, and checksum of each file in the SU. This information is used by the verification process to verify each SU component before activation.
- **Message File:** The message file contains the commands necessary to install the SU, plus any special instructions required. Figure 21-3 shows an example of a Program Documentation Standard (PDS) message file.
- **Binary Update File:** The binary update file contains the binary data for a file targeted for the update.
 - For nonkillable processes, these take the form of minimal object (.m) files.
 - For killable processes, the update files are object files for target processes.

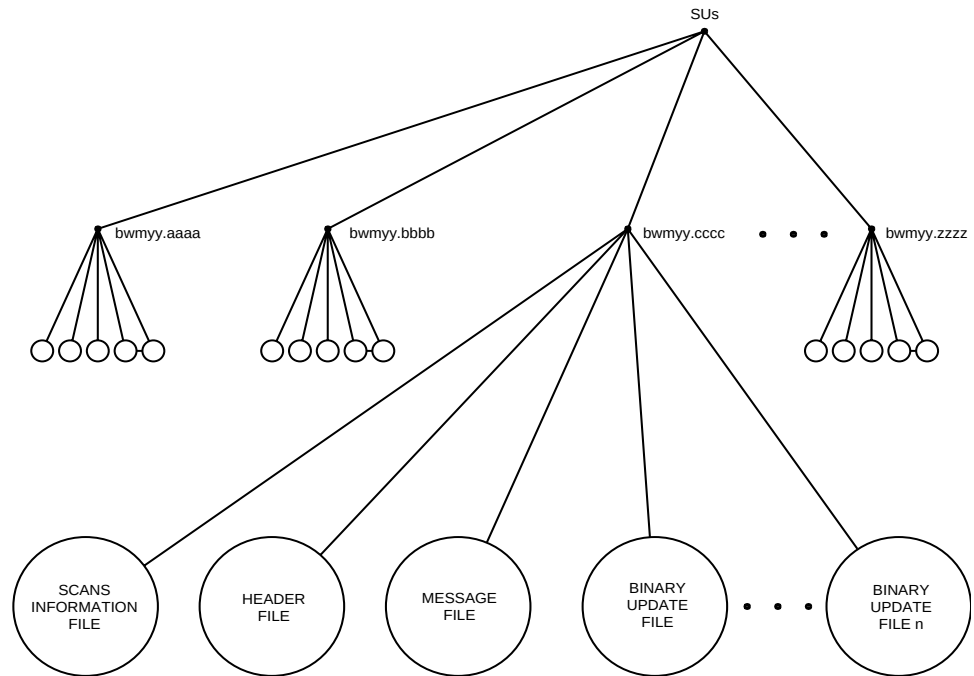


Figure 21-2. Software Update Structure

```
"BEGIN BWM94-001 PDS
"
"APPLY.-----"
"
UPD:APPLY, FILER, UPNM "BWM94-0001", FN "/bin/date", UF "date"!
UPD:APPLY, FUNCRC, UPNM "BWM94-0001", FN "/libc", UFFN "libc.ls"!
UPD:APPLY, FUNCRC, UPNM "BWM94-0001", FN "/prc/bdf", UFFN "iodrv.o.ls"!
"
"SOAK.-----"
"
"      This SU should soak for 12 hours before being made official.
"
"
"BKOUT.-----"
"
UPD:BKOUT:UPNM "BWM94-0001"!
"
"OFFICIAL.-----"
"
"
UPD:OFC:UPNM "BWM94-0001"!
UNIXCMD:/bin/cmpr -f /libc
UNIXCMD:/bin/cmpr -f /prc/bdf
"
"
"END BWM94-0001
```

Figure 21-3. Example of a PDS Message File

SCCS and MCC Interface

SUs are normally activated remotely by the SCCS. Communication between the SCCS and the program update subsystem is over the remote maintenance channel. The local office can be unattended.

SUs may also be activated locally at the Master Control Center (MCC) video terminal. If the SUs are to be requested from the SCANS by the local office, a 1200-baud terminal (other than the MCC) must be present. The terminal must be full duplex, capable of printing at least 80 characters per line and must have a 212A compatible modem attached. If the SUs are to be loaded into the switch from a tape, the 1200-baud terminal is not needed.

SU Reception from the SCANS

A dial-up link must be established with the SCANS to receive SUs. When the dial-up link is completed, the proper login, password, and subpassword is used to gain access to the SCANS database. When access is obtained, a listing is requested of all items from the SCANS. This listing is reviewed, and all applicable SUs are stored.

The office storage space required for each SU binary data package is provided in the software change size section of the SU. Prior to data transmission, it must be determined that sufficient file space is available in the directory on disk where the SUs are to be stored. If file space is insufficient, memory audit and space reclamation techniques are used to create space or the SUs are requested in stages (one or two at a time) as space allows.

When enough file space is available, the office is primed for binary data package delivery by using the **IN:REMOTE:START;** message. During priming, file space is reserved and a transaction identifier is established. This identifier provides additional security at the application level in the file transfer process.

The office will receive data from the SCANS for up to 24 hours after it is primed unless the data session is manually terminated using the **IN:REMOTE:STOP;** message. If the SCANS does not begin to send data within the 24-hour period, the data session times out and an **IN REMOTE ERROR** message is displayed at the SCCS (or the MCC). After a time-out, the office will need to be primed again.

After priming, a delivery request is made to the SCANS for the binary data packages. This request, which is usually made from a dial-up terminal between the SCANS and the SCCS, issues a Binary Overwrite (BOW) command to the SCANS. The BOW command includes the office name, the identification of the SUs, and a transaction identifier.

As soon as a port is available, the SCANS sets up a four-wire dial-up link to the target office. This link is used for delivery of the binary data packages. Assuming no equipment failures or unusually high demand, the SCANS will establish this link within 24 hours of the delivery request.

Using the access login and the transaction identifier, the SCANS accesses the 3B20D computer. It then establishes a 4800-bps BX.25 data link to the target office. The binary update files, along with the header and message files, are then transmitted to the target office to be placed on disk in the update directory (*/etc/bwm*) of a storage partition.

Upon successful reception of the binary overwrite files, the **IN REMOTE STOPPED** message is dumped at the SCCS workstation and/or the MCC video terminal. This message indicates that the binary files have been received and loaded into the */etc/bwm* directory.

Assuming that the binary update files have been received and verified, the BX.25 data link is automatically terminated.

The SUs received from the SCANS are the same SUs generated at the PSS. The SCANS does not alter the internal structure or format of any SU. The SCANS information file is not required by the software update subsystem. If this file is sent along with the BOW command, it is ignored.

SU Reception from the CSCANS

To receive SUs from the CSCANS, a dial-up link must first be established with the CSCANS. When the dial-up link is completed, the proper login and passwords must be used to gain access to the CSCANS database (for those offices that are so equipped). For subscribing offices, follow local CSCANS procedures for accessing the database and requesting applicable SUs.

The office storage space required for each SU binary data package is provided in the software change size section of the software. Prior to data transmission, it must be determined that sufficient file space is available in the directory on disk where the SUs are to be stored. If the file space is insufficient, memory audit and space reclamation techniques are used to create space, or the SUs are requested in stages (one or two at a time) as space allows.

When enough file space is available, the office is primed for binary data package delivery with the following message:

UPD:INITPW:PASSWD="xxxxxx",KEY="yy";

In priming, a password and key must be selected through use of the **UPD:INITPW** message. The identical password and key must be set in the CSCANS database. This password provides additional security at the application level in the file transfer process.

The office will receive data from the CSCANS at any time after it is primed and after the CSCANS database is updated. The office will normally not need to be primed again. If for security or other reasons, the office may be primed again with the following message:

UPD:INITPW:PASSWD="xxxxxx",KEY="yy";

The CSCANS database must also be updated accordingly with the newly chosen password and key.

After priming, a delivery request is made to the CSCANS for the binary data packages. Follow local CSCANS procedures for requesting a binary data package. Upon successful reception of the binary overwrite files, a **UPD:CSCANS** message is dumped at the SCCS workstation and/or the MCC video terminal to indicate that the binary files have been received and loaded into the */etc/bwm* directory.

Assuming that the binary update files have been received and verified, the BX.25 data link is automatically terminated.

Verification

A process is provided by the software update subsystem to verify SUs before overwriting the resident software release. When the system receives the verification command, a check is made to confirm the existence and correctness of all files and associated checksums.

If a software update error is detected through verification, the SU in question should be requested again from the SCANS. If the SU fails verification a second time, the Electronic Switching Assistance Center (ESAC) and the Regional Technical Assistance Center should be notified.

Activation

This section is a general description of an SU activation. Enhanced SU capabilities through the use of a menu-driven craft interface are also available.

After the SUs have been successfully verified, an executable message file for a specific SU is created. The message file contains an instruction set for activating a particular SU.

⇒ NOTE:

The SUs must be activated in numerical order.

The executable message file contains the following user-accessible sections. The APPLY, SOAK, and OFFICIAL sections are used for normal installation. The BKOUT section is used for emergency backout procedures. Once an executable message file has been created for a specific SU, that software update can be installed by entering forms of the **UPD:EXALL;** message as follows:

- The APPLY section is used to place an update into a temporary mode. It is executed by entering the **UPD:EXALL:APPLY;** message.
- The SOAK section contains the recommended period of time when the SU is tested and observed for proper operation. Although it is possible to execute the SOAK section by using the **UPD:EXALL:SOAK;** message, each step of the SOAK section, where applicable, should be performed on a manual basis to allow close observation of the soak interval.
- The BKOUT section is used only when it becomes necessary to back out of a temporary SU. It is executed by entering the **UPD:EXALL:BKOUT;** message. This section has been enhanced to allow for up to three official SUs to be backed out.

- The OFFICIAL section is used to make the temporary update permanent following successful completion of the soak interval. It is executed by entering the **UPD:EXALL:OFC;** message.

The APPLY, OFFICIAL, and BKOUT sections each contain one or more executable messages.

The normal progression for SU activation is to execute the APPLY section first, followed by the SOAK section, and finally the OFFICIAL section.

If the system is rebooted when temporary updates are in place (other than switching module updates), the temporary disk file is thrown away and the system boots from the official file. This effectively backs out the update(s).

When a temporary update is made permanent, an updated version of the file is built in the same directory as the original version. A windowless move then takes place to effectively make the new version official. The incore memory is reclaimed to recover the old functions as new patch space.

The SUs can also be activated by maintenance personnel at the SCCS or the MCC by printing the message file of the applicable SUs and manually installing the required messages.

Backout

During the soak interval, the temporary SU is observed to ensure proper performance. If the SU does not perform properly during the soak interval, it should be backed out using the **UPD:EXALL:BKOUT;** message. The BKOUT section backs out the designated change, plus any subsequently installed temporary changes. The backout process deletes changes in the reverse order of application. A sequential list of changes that are in the temporary state may be obtained using the **UPD:DISPLAY;STATUS=TMP;** message.

Emergency Fix

Regular SUs are performed in a timely and orderly fashion. Unexpected problems with the software release can occur that require immediate correction. This does not allow time for the regular SU development and issue processes. These immediate corrections are known as emergency fixes. Emergency fixes are accomplished on a word-by-word basis under the direction of the Lucent Technologies Product Engineering Control Center (PECC).

Emergency fixes are assigned a sequential craft number similar to the software update number. The program update subsystem provides emergency fixes with the same status and processes as SUs (that is, make temporary, back out, and make permanent). Emergency fixes specify the address to be changed, the new data to be inserted, and the old data to be matched. Emergency fixes are also known as address-data couplets.

SCCS and MCC Interface

As with SUs, most emergency fixes are activated remotely by the SCCS. Communication between the SCCS and the program update subsystem is via the maintenance channel. The local office can be unattended during the activation of the fix. Emergency fixes may also be activated locally through the MCC.

Activation

The **UPD:APPLY ...**; message causes a temporary change to be made at the specified location. After a suitable test and soak period, the **UPD:OFC:UPNM ...**; message makes the change permanent, and normal backout procedures can no longer be used.

Backout

Normal backout procedures can be accomplished only while the fix is in a temporary state. A backout procedure can be implemented using the **UPD:BKOUT:UPNM**; message. This message backs out the designated change, plus any subsequently installed changes, because the backout process can delete changes only in the reverse order of application. The **UPD:DISPLAY:STATUS=TMP**; message provides a sequential list of changes that are in a temporary state.

⇒ NOTE:

The **UPD:BKOUT:UPNM**; message restores only the words specified in the change. If memory other than that specified in this change is mutilated, a system BOOT may be required to restore the switching system to normal.

Space Reclamation

Reclaim Space in SU Storage Directory

As software updates are brought into the */etc/bwm* directory and activated, the space available in */etc/bwm* for future update storage is gradually reduced. To avoid running out of space, the file space occupied by software updates that have been made permanent should be cleared using the following message:

UPD:CLR:BWM;

A listing of permanent updates should be obtained using the following message:

UPD:DISPLAY:STATUS=OFC;

and compared to the contents of */etc/bwm* using the following message:

OP:STATUS:LISTDIR,DN="/etc/bwm";

Reclaim Patch Space

Update functions are installed in patch space. After an update has been soaked and made permanent, the old functions and decision functions are normally no longer needed. These are removed from execution, and the space is automatically reclaimed for use on future updates when the update is made permanent.

Error Conditions

Table 21-1 lists possible error conditions that may be encountered during file transfer. Should any error conditions arise during update activation that are not listed in the table, refer to the error condition listing in one of the following IPs for the particular message in question:

- 254-303-110, *PDS Input Message Manual*
- 254-303-111, *PDS Output Message Manual*
- 254-303-112, *MML Input Message Manual*
- 254-303-113, *MML Output Message Manual*.

Table 21-1. File Transfer Error Conditions (SCANS/CSCANS Interface)

Cause	Effect	Action to Be Taken
Insufficient space for incoming file (initial size for the <i>/etc/bwm</i> directory is 5 MB).	File transfer session terminated.	(1) Verify and install applicable software updates received from the SCANS-2/CSCANS. (2) Clear the <i>/etc/bwm</i> directory after updates have been made permanent. (3) Enter second request for remaining software updates.
Incoming file length exceeds maximum file length permitted (2 MB maximum).	File transfer continues with next file.	(1) Invalid software update. Contact the ESAC and the PECC.
File with identical pathname already exists.	File transfer continues with next file.	(1) Verify all software updates at completion of session. (2) Clear those software updates in which verification was unsuccessful. (3) Enter second request for any remaining software updates.
Incoming file interrupted by sending process.	File transfer session will be terminated if three consecutive files are interrupted.	(1) Enter second request for affected software updates. (2) If problem persists, contact field update administrator.
Sending process is unable to continue and aborts session.	File transfer session terminated.	(1) Verify and install applicable software updates received from the SCANS/CSCANS. (2) Enter second request for transmission of remaining software updates. (3) If problem persists, contact the SCANS/CSCANS administrator.
Unrecoverable conditions detected by the retrieve process.	File transfer session terminated.	(1) Verify and install applicable software updates received from the SCANS-2/CSCANS. (2) Enter second request for transmission of remaining software updates. (3) If problem persists, contact the PECC.

Enhanced SU Capabilities

The enhanced software update uses a menu-driven craft interface. This interface provides a user-friendly program update procedure which simplifies SU installation. These enhancements eliminate the need for lengthy input messages that must be precisely entered.

When a software update package is received, four types of files are included.

- A SCANS information file contains a description of the problem being corrected and administrative information used by the SCANS to determine to which offices the SU should be sent.
- A header file is used to verify the SU on-site. This file contains information such as target software release issue, file size, and checksum for all files in the SU.
- A message file (Figure 21-3) contains a set of craft commands required to install a given SU. The message file provides commands and instructions used by an automated display mechanism to guide the craft through the process of installing the associated SU. This process reduces the amount of time required for, and the potential errors associated with, manual message inputs. The craft may examine the contents of the message file and monitor the status of an update transaction via video display pages.
- One or more binary update files contain modified or new functions in the form of an object file to fix a process.

The menu-driven craft interface provides software update installation menu pages (Figure 21-4) and program update pages. Numbered commands, called pokes, are entered from these display pages to perform the desired procedures. Refer to the following sections for detailed procedures using the menu-driven craft interface:

- Page 1940 – Easy SU Installation
- Page 1941 – SU Automation Scheduling
- Page 1942 – SU Automation Office Profile
- Page 1943 – SU Automation Healthcheck
- Page 1950 – SU Maintenance
- Page 1960 – SU Installation.

CMD<		SOFTWARE UPDATE PAGE (109)	
<u>CMD</u>		<u>PAGE</u>	<u>TITLE</u>
1940	-	EASY	SU INSTALLATION
1941	-	SU	AUTOMATION SCHEDULING
1942	-	SU	AUTOMATION OFFICE PROFILE
1943	-	SU	AUTOMATION HEALTHCHECK
1950	-	SU	MAINTENANCE
1960	-	SU	INSTALLATION

Figure 21-4. Software Update Page 109

SU Installation

MCC Page 1941 (SU Automation Scheduling), Page 1942 (SU Automation Office Profile), and Page 1943 (SU Automation Healthcheck) provide a method to schedule SUs for automatic installation. With this method, the user is required to schedule a specific SU to be installed on a given day and time to a specific install state (verify, soak, or official). Once the SU has been scheduled, the user is not required to constantly monitor the SU progress. When the previously specified day/time arrive, the specified SU will begin automatic installation. Refer to the following sections for details on how to use MCC Display Pages 1941, 1942, and 1943:

- Page 1941 – SU Automation Scheduling
- Page 1942 – SU Automation Office Profile
- Page 1943 – SU Automation Healthcheck.

Many of these capabilities (for example, scheduling SUs for automatic installation) are also available from the CSCANS over the CSCANS interface. Subscribing offices should follow local CSCANS procedures for accessing the CSCANS database and administering these new capabilities. MCC Display Page 1940 and Page 1960 may still be used as alternative manual methods of loading or backing out SUs. The following sections provide details on how to use them:

- Page 1940 – Easy SU Installation
- Page 1960 – SU Installation.

The craft can install an SU with the SU installation menu. This menu is on the MCC Page 1960 (Figure 21-5). The craft interface simplifies the task of installing an SU by directing you through the installation process.

- The upper area of the page provides a list of menu items, each of which is identified by a poke number.
- The lower area of the page provides a display window for sections of the message file. The current section of the message file, called the working section, is displayed so it may be examined before execution. If the working section does not fit in the designated window, the craft can scroll forward and backward using the NEXT WINDOW and PREV WINDOW pokes. The message file may be printed on the MCC printer by entering the PRINT poke provided on the menu page.
- A RESPONSE line is provided to provide response messages to the craft during SU installation.

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR SU	MINOR SU PERPH	BLDG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD: _				1960 - INSTALL SU=BWM94-0001			
9000,Y	START SU Y	9570	NEXT WINDOW	91XX	DISPLAY (XX =	10	APPLY
9010	VERIFY CMPL	9575	PREV WINDOW	92XX	PRINT 20	SOAK	30 OFC
9560	STOP EXC	(Y = 10 CHAR SU NAME)		93XX	EXEC ALL 40	BKOUT	50 FILE)
9565,Z	RESET LINE Z	(Z = 3 DIGIT LINE NO)		94XX	EXEC NEXT		
				(F = FILENAME IN SU)	9260,F	PRINT FILE F	
SECTION EXECUTION STATUS							
APPLY CMPL		SOAK CMPL		OFC INPG		BKOUT	
RESPONSE: _____							
11	"						
12	"	OFFICIAL	_ _ _ _ _				
13	"						
14	" To make the update applied in this SU official, wait the						
15	" prescribed SOAK interval and then enter sequentially the						
16	" following command(s):						
17	"						
18	UPD:UPNM="BWM94-0001":OFC;						
19	"						

Figure 21-5. Page 1960 – SU Installation Menu

Sequencing checks are made between SUs to ensure proper sequencing for all SUs.

To select a particular SU, the START SU Y poke is entered along with the desired SU number. The menu page then displays the current status of the requested SU and sets the "SU=xxxnn-nnnn" indicator to the requested SU name.

Messages that are not in the language format required for the specific office are automatically deleted from the message file by the VERIFY poke. Additionally, this poke checks the message file integrity. The VERIFY poke must be executed before the SU can be applied to the system. When this poke is completed successfully, an indicator is provided on the display to notify the craft that the SU was successfully verified. The VERIFY poke may be executed as many times as desired. However, if there was no change to the message file since the last VERIFY poke was executed and the VERIFY flag is set to CMPL, subsequent VERIFY pokes are rejected. When the SU has been successfully verified, the contents of the APPLY section of the message file are automatically displayed.

The action indicators (DISPLAY, PRINT, EXEC ALL, and EXEC NEXT) are used in conjunction with the section indicators (APPLY, SOAK, OFC, BKOUT, and FILE) to perform the desired SU installation functions. A single poke number represents the action desired and the section of the message file affected. The first two digits represent the action. The second two digits represent the section.

Poke **9310** (EXEC ALL APPLY) applies the selected SU. This poke causes all commands in the APPLY section of the message file to be executed automatically and in sequential order. These commands may be executed one at a time using Poke **9410** (EXEC NEXT APPLY). Poke **9410** must be used repetitively to execute all the commands in the APPLY section. Poke **9310** may be used at any time after a Poke **9410** to execute all remaining command lines in the APPLY section of the message file.

Subsequent to an EXEC poke, the RESET LINE poke may be used to reset the current command to be executed. This poke is entered followed by a comma and a line number to which the execution pointer should be reset. The specified line is then highlighted and may be executed. If the requested line number is in the next or previous display window, the RESET LINE poke adjusts the display window.

If an error occurs during the execution of a command line, a summary of the error is displayed on the RESPONSE line of the menu page. The command line causing the error is displayed, and a detailed error message is printed on the MCC printer. If Poke **9310** is being used when the error occurs, execution of the next command line (if any are still remaining) is automatically stopped.

Poke **9560** (STOP EXC) may be used subsequent to the EXEC ALL poke to stop execution of the next command line in the section.

When all command lines in the section are successfully executed, a CMPL indicator associated with the APPLY status is displayed and the content of the SOAK section is automatically displayed. This guides the user to the next step of the SU installation process. This section of the message file contains the procedures required to activate portions of the SU and the required soak interval.

Poke **91xx** (DISPLAY) displays selected sections of the message file. The first ten lines of the section are displayed in the window. If more than ten lines exist for the selected section, the NEXT WINDOW and PREV WINDOW pokes may be used to scroll forward and backward through the section. These pokes do not affect the execution of commands. They may be used only after a DISPLAY or EXEC poke has been entered for a given SU.

Poke **9260,F** (PRINT FILE F) prints any American Standard Code for Information Interchange (ASCII) files associated with the currently installed SU.

If a failure occurs during the soak procedure, the SU may be backed out. Poke **9140** (DISPLAY BKOUT) displays the BKOUT section of the message file. This section provides information to the craft and commands required to back out the given SU.

Poke **9340** (EXEC ALL BKOUT) automatically executes all commands in the BKOUT section of the message file, one by one, and in sequence. When the execution of these commands is completed successfully, the content of the APPLY section is again automatically displayed on the menu page.

When the SU has successfully completed the SOAK section, it may be made official. This means that the software release file in the official disk partition is updated. Thus, the update can be saved across system bootstraps. However, reloading the system from the backup disk partition will destroy the update unless the updated software release file is copied to the backup partition. Poke **9330** (EXEC ALL OFC) makes the SU official. This poke cannot complete until all command lines in the APPLY section have completed successfully, the SOAK section has completed, and the timer has expired.

SU Maintenance

Menu Page 1950 displays pokes (see the “Page 1950 – SU Maintenance” section) that provide an on-line history of all SUs installed into the current software release issue in a 3B20D/3B21D computer. The history of an SU (Figure 21-6) consists of, but is not limited to, the following information:

- Name of software update
- Status of software update (temporary or official)
- Time of last transaction
- Pathname of target object file
- Pathname of the update files for the target object.

```
M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=6 GEN=R21.7.1
      PRODUCT PATH=/prc/kkern SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:10:10 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'1d
            PERMED  TEMPED  OGEN  OILD

      ATTRIBUTES=x'14224
            KERNAL  KILLABLE  FUNCR  AM_PROCESS
            UFFN
      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:53 #000428
```

```
M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=5 GEN=R21.7.1
      PRODUCT PATH=/prc/bdf  SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:09:52 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'b
            PERMED  INMEMORY  OGEN
      ATTRIBUTES=x'14244
            KERNAL  CRC_INTERNAL  FUNCR  AM_PROCESS
            UFFN
      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:53 #000429
```

Figure 21-6. Typical 3B20D/3B21D Computer On-Line SU History (Sheet 1 of 2)

```
M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=4 GEN=R21.7.1
      PRODUCT PATH=/libc      SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:08:51 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'b
            PERMED INMEMORY OGEN
      ATTRIBUTES=x'14250
            SHRLIB CRC_INTERNAL FUNCR AM_PROCESS
            UFFN
      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:53 #000430

M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=3 GEN=R21.7.1
      PRODUCT PATH=/prc/file1 SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:08:23 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'5
            PERMED TEMPED
      ATTRIBUTES=x'4c00
            FILER NEW_PRODUCT AM_PROCESS
      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:53 #000431

M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=2 GEN=R21.7.1
      PRODUCT PATH=/bin/data  SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:08:15 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'5
            PERMED TEMPED
      ATTRIBUTES=x'4428
            SUPERVISOR KILLABLE FILER AM_PROCESS
      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:53 #000432

M 17 UPD DISPLAY UPDATE DATABASE RECORD NUMBER=1 GEN=R21.7.1
      PRODUCT PATH=/prc/SUgenpcc      SUNAME=BWM94-0001
      APPLY DATE=Tue Apr 12 13:08:06 1994
      HDR SEQUENCE NUMBER=1
      STATUS=x'5
            PERMED TEMPED
      ATTRIBUTES=x'4428
            SUPERVISOR KILLABLE FILER AM_PROCESS

      DISPLAY OF UPDATE RECORD COMPLETED
04/12/94 13:17:54 #000433

M 17 UPD COMMAND DISPLAY COMPLETED
04/12/94 13:17:54 #000434
```

Figure 21-6. Typical 3B20D/3B21D Computer On-Line SU History (Sheet 2 of 2)

When menu Page 1950 is selected and displayed, the history for a particular SU can be requested.

Poke **9101** (SU Y) followed by a comma and the desired SU name requests a history of all updates associated with the specified SU to be printed on the Receive-Only Printer (ROP).

Poke **9102** (OFC) requests a history printout of all updates in the system which have been made official.

Poke **9103** (TEMP) requests a history printout of all SUs which are in the temporary state.

Poke **9104** (ALL) requests a history printout of all SUs in the active log.

- Poke **9104,BACKLOG** prints the history of the SUs in an archive log. If all SUs in the active log have been made official and memory space used by the old version of the replaced functions has been reclaimed, the history of these SUs may be stored in the archive log to conserve disk space. These histories may be printed on the ROP.
- Poke **9104,SUM** requests a list of the last official updates, the active craft updates, and the temporary SU.

The updates shown in Figure 21-6 are in a temporary state. The most important status flag is the installed flag (TEMPED or INMEMORY). A processor initialization can cause temporary updates in the 3B20D/3B21D computer to be removed. Consequently, the installed flag is reset and cleared from the software update history after completion of the initialization.

A PERMED flag is added to the history entry after the update is made official. For a function replacement, a reclaimed flag is added to the history after memory space used by the version of the replaced function is reclaimed.

A file replacement may be verified by checking the time stamp and file size on the target file at the directory where the original file is located. When a function replacement is installed, the new address of the function is printed on the printer. The new address may be used to establish a breakpoint to detect when the replacement function is called.

Menu Page 1950 also provides the capability to detect and recover update inconsistencies in the system.

Poke **9200** (VERIFY INCONSISTENCY) detects inconsistencies. For example, if a temporary update is booted out of the processor as a result of an initialization, an update inconsistency occurs.

Poke **9300** (RECOVER FORWARD) reapplies all temporary updates in an inconsistent state.

Poke **9400** (RECOVER BACKWARD) removes the temporary updates, starting from the last one in the system. If there is no inconsistency, a message will print to the ROP.

Pokes **9300** and **9400** automatically determine the starting and ending points of the software update sequence so that a recovery process can be performed.

As the number of SUs installed in the system increases, disk space used by the history files grows. When all SUs have been made official, most software update histories can be saved in an archive file in a condensed format. This is done by entering the **UPD:REDUCE** command manually. There is no corresponding poke on Page 1950.

Poke **9600** (CLEAR SU Y) clears disk space occupied by the SUs after the updates are made official. An automatic check is made to verify that the specified SUs were made official before the space is cleared.

- **9600,"SUname"** clears the space occupied by all SUs which have been made official.
- **9600** followed by a comma and the specific software update name clears the space occupied by a single SU.

Poke **9650** (EXPAND SU) expands a compressed SU after it has been completely transmitted to the switch. This same poke may also be used to stop expansion on the SU that is currently being downloaded.

Pokes **9700**, **9710**, **9720**, and **9730** manipulate the soak timer information.

- Poke **9700** (RESET TIMER) resets the soaking time to **HH** hours and **MM** minutes.
- Poke **9710** (PRINT TIMER INFO) prints the soak timer information on the ROP.
- Poke **9720** (ABORT SOAK TIMER) aborts the soak timer.
- Poke **9730** (SET DEFAULT SOAK TIMER) sets the default soak timer to **HH** hours and **MM** minutes for the office. The default soak timer is used to set the initial soak timer when manually installing an SU through MCC Page 1960 or Page 1940. It also overwrites the existing soak timer field in the SU Automation Office Profile (MCC Page 1942).



NOTE:

This poke will not modify any existing soak timer that is already in progress.

Poke **9900** (BACKOUT OFC) backs out, in the reverse order in which they were applied, the last three official SUs.

Routine Operations and Maintenance Procedures

This section presents the following procedures to perform routine operations and maintenance:

- Schedule Making SUs Official
- Install SUs
- Check/Make Official Temporary SUs
- Make SUs Official
- Back Out SUs
- Back Out Last Official SUs
- Manual Stop and Start of SU Expansion
- Obtain SU Status Report
- Request Summary/Long Summary of SUs
- Load SUs from SCANS or Tape.

Schedule Making SUs Official

Use the following procedure to schedule software updates:

Step	Procedure
1	Obtain the software update status report. Reference: "Obtain SU Status Report" section.
2	Dependent on results of Step 1: If SUS DUE, continue to Step 3. If NO SUS DUE, wait approximately 1 day and repeat from Step 1.
3	Request a listing of available software updates. Reference: "Request Summary/Long Summary of SUs" section.
4	Schedule a date for making software updates official per local instructions.
5	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Install SUs

Use the following procedure to install software updates:

Step	Procedure
1	Check or make official software updates in the soak state in the switch. Reference: "Check/Make Official Temporary SUs" section.
2	Load the software updates. Reference: "Load SUs from SCANS or Tape" section.
3	Make a listing of the software updates loaded in the 3B20D/3B21D computer.
4	Make the software updates official. Reference: "Make SUs Official" section.
	 NOTE: The MCC Page 1940 – Easy SU Installation provides a simplified alternative method of installing software updates. With this method, the user is required to enter a minimum number of pokes manually and is not required to constantly monitor the software update progress. Page 1940 uses the tools presently found on MCC Pages 1950 and 1960. Refer to the "Page 1940 – Easy SU Installation" section for details on how to use this alternative method of installing the software update. MCC Page 1960 is still the standard method to install software updates.
5	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Check/Make Official Temporary SUs

Use the following procedure to check or make temporary software updates.

⇒ NOTE:

In this procedure, the term “software updates in the soak state” does not refer to an actual temporary software update but rather to a software update that has not yet been made official.

Step	Procedure
1	Display the temporary software updates. Type and enter the following command on the MCC Page 1950: 9103 Response: In the field adjacent to Poke 9103 , IN PROGRESS is highlighted followed by Step (a), (b), or (c). (a) COMPLETED and a list of temporary software updates is printed at the ROP. (b) COMPLETED and no list is printed at the ROP. (c) ABORTED indicates no temporary updates in the switch.
2	Which response did you get: Step (a), (b), or (c)? If response is Step (a), continue to Step 3. If response is Step (b) or (c), STOP. YOU HAVE COMPLETED THIS PROCEDURE.
3	Make a list of temporary software updates that need to be made permanent.
4	Select the software update with the lowest sequence number. Use the list of temporary software updates.
5	Access the MCC Page 1960 – SU Installation.
6	Is a software update number displayed to the right of SU = ? If YES , continue to Step 7. If NO , go to Step 8.

Step	Procedure
7	Is the software update number shown the same software update that was previously selected? If YES, go to Step 9. If NO, continue to Step 8.
8	Start a software update. Enter the following command on the MCC Page 1960: 9000,"yyyyyyyyyy" Where: yyyyyyyyyy = a ten-digit software update number. Response: On the MCC Page 1960, status of software update is displayed under SECTION EXECUTION STATUS. The message file for next section to be executed is displayed.
9	Execute the SOAK section of the message file. Enter the following command on the MCC Page 1960: 9320 Response: SOAK SECTION COMPLETED: SOAK SECTION
10	Has the soak interval for this software update completed? Use information contained in the SOAK section of the message file. If YES, go to Step 12. If NO, continue to Step 11.
11	Wait until the soak interval has completed.
12	Make the software update permanent. Enter the following command on the MCC Page 1960: 9330 Response: EXECUTE ALL COMPLETED:OFFICIAL SECTION
13	Clear a software update. Enter the following command on the MCC Page 1950: 9600,"yyyyyyyyyy" Where: yyyyyyyyyy = a ten-digit software update number. Response: COMPLETED appears to the right of Poke 9600.

Step	Procedure
14	Was software update cleared successfully? If YES , go to Step 18. If NO , continue to Step 15.
15	Verify the correct software update name used.
16	Clear a software update. Enter the following command on the MCC Page 1950: 9600,"yyyyyyyyyy" Where: yyyyyyyyyy = a ten-digit software update number. Response: COMPLETED appears to the right of 9600 poke.
17	Was the software update successfully cleared? If YES , continue to Step 18. If NO , seek technical assistance.
18	Are there any other temporary software updates active in the switch? If YES , continue to Step 19. If NO , STOP. YOU HAVE COMPLETED THIS PROCEDURE.
19	Select the software update with the next highest sequence number and repeat the procedure from to Step 8.

Make SUs Official

Use the following procedure to make software updates official:

Step	Procedure
1	Use the following command to list software updates. Choose the software update with the lowest sequence number. OP:STATUS,LISTDIR,FN="/etc/bwm";
2	Start the software update process. Type and enter the following command on the MCC Page 1960: 9000,"yyyyyyyyyy" Where: yyyyyyyyyy = a ten-digit software update number, craft software update number, or temporary software update number. Response: ENTER 9010 TO VERIFY SU Example: BWM93-0011
3	Verify the software update and build an executable message file. Enter the following command on the MCC Page 1960: 9010 Response: UPD VFY COMPLETED. The APPLY section is displayed under SECTION EXECUTION STATUS. ⇒ NOTE: If the command fails, determine whether the SU exists in the directory or whether the SU name was typed in error. Retry SU a second time. Reload from tape or SCANS. If an error message occurs again, seek assistance.
4	Print the entire message file. Enter the following command on the MCC Page 1960: 9250 Response: A message file is printed at the ROP.
5	Review the message file for events that will take place during activation or any special actions which must be accomplished manually.

Step	Procedure
6	Apply the software update in a temporary mode. Enter the following command on the MCC Page 1960: 9310 Response: EXECUTE ALL COMPLETED: APPLY SECTION
7	Execute the SOAK section of the message file. Enter the following command on the MCC Page 1960: 9320 Response: EXECUTE ALL COMPLETED: SOAK SECTION  NOTE: All SUs soak for a default of 24 hours. If soak times require adjustments, use Page 1950.
8	Has the soak interval for this software update completed? Use information from the soak indicator on the MCC Page 1960. If CMPL (complete), go to Step 10. If TMPG (timer in progress), continue to Step 9.
9	Wait for the software update soak interval to complete.
10	Is the software update that is being activated a temporary software update? If YES, STOP. YOU HAVE COMPLETED THIS PROCEDURE. If NO, continue to Step 11.
11	Make the software update official. Enter the following command on the MCC Page 1960: 9330 Response: EXECUTE ALL COMPLETED: OFFICIAL SECTION

Step	Procedure
12	Are there any other software updates that need to be activated? If YES , continue to Step 13. If NO , go to Step 15.
13	Choose the software update with the next higher sequence number.
14	Repeat the procedure from Step 2.
15	Clear the software updates. Enter the following command on the MCC Page 1950: 9600,BWMnn-nnnn Response: COMPLETED to the right of the poke.
16	Check the <i>/etc/bwm</i> directory. Results: SUs made official.
17	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Back Out SUs

Use the following procedure to back out software updates. To back out the last official software updates, refer to the "Back Out Last Official SUs" section.

Step	Procedure
1	Access the MCC Page 1960 – SU Installation.
2	Is the software update number displayed to the right of SU the same as the software update that needs to be backed out? If YES , go to Step 4. If NO , continue to Step. 3
3	Start the software update process. Enter the following command on the MCC Page 1960: 9000,"yyyyyyyyyy" Where: yyyyyyyyyy = a ten-digit software update number. Response: On the MCC Page 1960, status of software update is displayed under SECTION EXECUTION STATUS. A message file is displayed for the next section to be executed.
4	Back out the software update. Enter the following command on the MCC Page 1960: 9340 Response: EXECUTE ALL COMPLETED: BACKOUT SECTION
5	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Back Out Last Official SUs

Use the following procedure to back out last official software updates:

Step	Procedure
1	Access the MCC Page 1950 – SU Maintenance. The last three official software updates (at most) are displayed in the reverse order. These software updates can be found in the lower left-hand corner of the page (see the “Page 1950 – SU Maintenance” section).
2	Enter the following command: 9900 After the command is entered, confirm your decision by typing Y or N. IN PROGRESS appears if the response is Y. Response: IN PROGRESS appears next to 9900 command line.
3	Did command complete successfully, and is COMPLETED displayed next to the 9900 command? If YES, continue to Step 4. If NO, repeat the procedure from Step 2. If the command continues to fail, seek technical assistance.
4	If the “new” last official software update needs to be backed out, return to Step 2.  NOTE: No more than three official software updates can be removed this way.
5	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Manual Stop and Start of SU Expansion

Software updates will be sent out (through the SCANS or the CSCANS) in a compressed format. Normally, with no manual intervention, software update expansion will occur automatically. If automatic software update expansion is not wanted, automatic expansion may be stopped and then manually restarted by following this procedure.

⇒ NOTE:

The automatic software update expansion is stopped only for a download session that is currently in progress.

Step	Procedure
1	Access the MCC Page 1950 – SU Maintenance.
2	Enter the following command: 9650,STOP Response: IN PROGRESS appears next to the 9650 command line.
3	Did the command complete successfully, and is COMPLETED displayed next to the 9650 command? If YES , continue to Step 4. If NO , repeat Step 2. If command continues to fail, seek technical assistance.
4	Is a manual software update expansion wanted now for a specific software update, and has that software update been completely received (from SCANS or CSCANS)? If YES , go to Step 6. If NO , continue to Step 5.
5	Wait until the software update has been completely received and until manual expansion is wanted.
6	Access the MCC Page 1950 – SU Maintenance.
7	Enter the following command: 9650,yyyyyyyyyy. Where: yyyyyyyyyy = a ten-character software update name. Response: IN PROGRESS appears next to the 9650 command line.

Step	Procedure
8	Did the command complete successfully and is COMPLETED displayed next to the 9650 command? If YES , STOP. YOU HAVE COMPLETED THIS PROCEDURE. If NO , repeat Step 7. If the command continues to fail, seek technical assistance.

Obtain SU Status Report

Use the following procedure to obtain a software update status report:

Step	Procedure
1	Is the SCANS linked up? If YES , go to Step 3. If NO , continue to Step 2.
2	Access the SCANS per local instructions.
3	At the SCANS terminal, type and enter: STA, L, U a Where: a = office name. Response: Receive the software update implementation status report (if any).
4	Are any software updates shown due? If YES , continue to Step 5. If NO , go to Step 6.
5	Record SOFTWARE UPDATES DUE.
6	Record NO SOFTWARE UPDATES DUE. Results: A copy of the software update status report with a list of software updates due or not due.

Request Summary/Long Summary of SUs

All communications between the 3B20D/3B21D computer and SCANS are via a full-duplex, 1200-baud dial-up terminal.

Use the following procedure to request a summary or a long summary of software updates:

Step	Procedure
1	Is the SCANS linked up? If YES , go to Step 3. If NO , continue to Step 2.
2	Access the SCANS per local instructions.
3	How are the software updates to be selected for viewing? (a) If by system name, continue to Step 4. (b) If by range of software update numbers, go to Step 6. (c) If by range of dates, go to Step 8. (d) If by software release name, go to Step 10. (e) If by software release issue, go to Step 12. (f) If by specific keyword, go to Step 14. (g) If by type of software update, go to Step 16.
4	At the SCANS terminal (SUMMARY), type and enter: SUM,S At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S Response: Summary of the software updates (if any) by the system name option.
5	Proceed to Step 17.

Step	Procedure
6	At the SCANS terminal (SUMMARY), type and enter: SUM,S,R a:b At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,R a:b Where: <i>a</i> = update number identifying first software update in range. <i>b</i> = update number identifying last software update in range. Response: Summary of the software updates (if any) by the system name option. Example: SU=nn - nnna . . . SU=nn - nnnz
7	Proceed to Step 17.
8	At the SCANS terminal (SUMMARY), type and enter: SUM,S,D aa lbb lcc :dd lee lff At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,D aa lbb lcc :dd lee lff Where: <i>aa</i> = the month the first software update of range was issued. <i>bb</i> = the day the first software update of range was issued. <i>cc</i> = the year the first software update of range was issued. <i>dd</i> = the month the last software update of range was issued. <i>ee</i> = the day the last software update of range was issued. <i>ff</i> = the year the last software update of range was issued. Response: Summary of the software updates (if any) by the range of software update issue dates.
9	Proceed to Step 17.

Step	Procedure
10	At the SCANS terminal (SUMMARY), type and enter: SUM,S,G a At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,G a Where: a = software release currently in the system. Response: Summary of the software updates (if any) by the 3B20D/3B21D computer software release name option.
11	Proceed to Step 17.
12	At the SCANS terminal (SUMMARY), type and enter: SUM,S,G a,I b At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,G a,I b Where: a = software release currently in the system. b = software release issue currently in the system. Response: Summary of the software updates (if any) by the 3B20D/3B21D computer software release issue option.
13	Proceed to Step 17.
14	At the SCANS terminal (SUMMARY), type and enter: SUM,S,K a At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,K a Where: a = keyword name, which could be a change notice number or other such keyword variable. Response: Summary of the software updates (if any) by the keyword option.
15	Proceed to Step 17.

Step	Procedure														
16	At the SCANS terminal (SUMMARY), type and enter: SUM,S,TY a At the SCANS terminal (LONG SUMMARY), type and enter: SUM+,S,TY a Where: a = one of the following software update types: <table><tr><td>GEN</td><td>Software release change</td></tr><tr><td>INF</td><td>Information only</td></tr><tr><td>APT</td><td>Auxiliary software change</td></tr><tr><td>APW</td><td>Auxiliary software change</td></tr><tr><td>PAR</td><td>Parameter</td></tr><tr><td>LIB</td><td>Library/Utility</td></tr><tr><td>BIN</td><td>Binary change.</td></tr></table> Response: Summary of the software updates (if any) by type option.	GEN	Software release change	INF	Information only	APT	Auxiliary software change	APW	Auxiliary software change	PAR	Parameter	LIB	Library/Utility	BIN	Binary change.
GEN	Software release change														
INF	Information only														
APT	Auxiliary software change														
APW	Auxiliary software change														
PAR	Parameter														
LIB	Library/Utility														
BIN	Binary change.														
17	At the SCANS terminal, type and enter: BWM,O a Where: a = office name. Response: Hard-copy printout of the software update text and the overwrite data for a 3B20D/3B21D computer office. Results: A SUMMARY or LONG SUMMARY of the software updates depending on which one was requested.														
18	STOP. YOU HAVE COMPLETED THIS PROCEDURE.														

Load SUs from SCANS or Tape

There is a 24-hour time limit imposed from the time the switch is primed until the SCANS must begin the data session. If the SCANS places a BOW request in the queue, it may take up to that 24-hour limit to start sending data to the switch.

- The `IN REMOTE STARTED` message is output to indicate the start of the data session.
- The `IN REMOTE REPT` message may be output periodically during the session to indicate various file rejects. It indicates termination of the session if followed immediately by the `IN REMOTE STOPPED` output message.

Use the following procedure to load software updates from SCANS or tape:

Step	Procedure
1	If loading software updates from SCANS, continue to Step 2. If loading software updates from tape, go to Step 13.
2	At the MCC, type and enter: IN:REMOTE,START; Response: <code>IN REMOTE INITIALIZED</code> <code>TRANSACTION ID: a</code> <code>TIME: b</code> Where: <i>a</i> = a transaction ID to be supplied to the SCANS when issuing a binary overwrite command. <i>b</i> = the time at which the 24-hour timer began, in the format hours:minutes.
3	At the SCANS terminal, access the SCANS per local instructions.
4	At the SCANS terminal, type and enter: BOW, S, R a[:b], O c, PW d Where: <i>a</i> = first or only software update in range. <i>b</i> = last software update in range (if ordering more than one software update). <i>c</i> = office name. <i>d</i> = transaction ID obtained in Step 2.

Step	Procedure																									
5	At the MCC, type and enter: IN:REMOTE,REPT; NOTE: In response to the IN:REMOTE,REPT input, the IN REMOTE REPT output indicates that the data session is still in progress. The IN REMOTE ERROR 12 indicates no IN REMOTE processes are active at this time (the data session is over, successfully or unsuccessfully).																									
6	Is the response IN REMOTE WAITING or IN REMOTE ERROR 12? If IN REMOTE WAITING, continue to Step 7. If IN REMOTE ERROR 12, go to Step 8.																									
7	Wait approximately 1 hour.																									
8	For a listing of SUs not in the switch, type and enter the following at the MCC: OP:STATUS:LISTDIR,FN="/etc/bwm"; Response: OP STATUS LISTDIR COMPLETED total 17 <table><tr><td>1</td><td>drwxrwxrwx</td><td>8 root</td><td>416 Mar</td><td>1 10:30 .</td></tr><tr><td>10</td><td>drwxr-xr-x</td><td>8 root</td><td>5024 Mar</td><td>2 11:00 ..</td></tr><tr><td>1</td><td>drwxrwxrwx</td><td>2 root</td><td>192 Mar</td><td>2 10:58 930011</td></tr><tr><td>...</td><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>drwxrwxrwx</td><td>2 root</td><td>112 Mar</td><td>1 10:30 930016</td></tr></table> If SUs or updates appear in this file, download and repeat from Step 5. NOTE: If Step 5 is repeated for the second time, verify the <i>/etc/bwm</i> directory, and repeat the procedure (start over) from Step 2.	1	drwxrwxrwx	8 root	416 Mar	1 10:30 .	10	drwxr-xr-x	8 root	5024 Mar	2 11:00 ..	1	drwxrwxrwx	2 root	192 Mar	2 10:58 930011	...					1	drwxrwxrwx	2 root	112 Mar	1 10:30 930016
1	drwxrwxrwx	8 root	416 Mar	1 10:30 .																						
10	drwxr-xr-x	8 root	5024 Mar	2 11:00 ..																						
1	drwxrwxrwx	2 root	192 Mar	2 10:58 930011																						
...																										
1	drwxrwxrwx	2 root	112 Mar	1 10:30 930016																						

Step	Procedure
9	At the MCC, type and enter: VFY:BWM,ALL; Response: UPD REPT INFORMATION VERIFICATION SUCCESSFUL FOR: SU NAME: or UPD REPT WARNING VERIFICATION UNSUCCESSFUL FOR: SU NAME:
10	Is this the first or second time verification has failed for software updates? If FIRST TIME, continue to Step 11. If SECOND TIME, seek technical assistance.
11	Determine which software updates failed verification and substitute for the original list. Use the list of software updates printed on the ROP.
12	Repeat the procedure from Step 2.
13	Mount the software update tape on the tape drive.
14	At the MCC, type and enter: COPY:TAPE:ATA,IN,TD="/dev/mt08",BSDIR="/etc/bwm"; Response: COPY TAPE IN COMPLETED
15	At the MCC, type and enter: VFY:"BWMaa-aaaa"; Where: aa-aaaa = a six-digit software update number. Response: UPD REPT INFORMATION VERIFICATION SUCCESSFUL FOR: SU NAME: or UPD REPT WARNING VERIFICATION UNSUCCESSFUL FOR: SU NAME:
16	Unmount the software update tape.

Step	Procedure
17	Ensure that the correct software update name was used in the previous steps during verification.
18	Type and enter: VFY:BWM"aa-aaaa"; Where: aa-aaaa = a six-digit software update number. Response: UPD REPT INFORMATION VERIFICATION SUCCESSFUL FOR: SU NAME: or UPD REPT WARNING VERIFICATION UNSUCCESSFUL FOR: SU NAME:
19	At the MCC, type and enter: COPY:TAPE,IN,TD="/dev/mt08",BSDIR="/etc/bwm"; Response: COPY TAPE IS COMPLETED
20	At the MCC, type and enter: VFY:BWMaa-aaaa; Where: aa-aaaa = a six-digit software update number. Response: VFY SU aaaa bb-bbbb c [d] [e]
21	Is c = OK or NG ? If OK , go to Step 16. If NG , seek technical assistance.
22	STOP. YOU HAVE COMPLETED THIS PROCEDURE.

Page 1940 – Easy SU Installation

Page 1940 provides a simplified procedure for installing SUs into the 3B20D/3B21D computer. The Easy SU Installation feature simplifies the SU application process by freeing the user from entering commands and constantly having to monitor the progress of the SU.

The Easy SU Installation page can be used to back out, install, or apply SUs with a single command. The page also provides control of the soak interval timer. This user specified time will override the 24-hour default value.

Several terms used in the commands are described as follows:

- **Install:** To take an official SU from the start command to the official command. The temporary SUs (beginning with TMP) cannot be made official.
- **Back Out:** To take a temporary or craft SU that is soaking and back it out of the system.
- **Apply:** To take a temporary or craft SU from the start command to the soak command.
- **Soak:** To leave an SU in the system for a certain period of time so as to allow the update to have a chance to interact with other pieces of software in the system.
- **Official:** To make an SU permanent in the system.

Page 1940 Description

Page 1940 (Figure 21-7) is divided into two basic areas: upper and lower.

- The upper area of the page contains the commands (pokes) used to control the SU processes. The commands are discussed in the “Page 1940 Commands” section.
- The lower area of the page provides a response indicating the progress of an SU procedure or a summary of an error message if an error occurs during the execution of a command. Also found in the lower area are the names of the Install, Back Out, and Apply SUs in addition to the value of the SU soak interval timer.

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR SU	MINOR SU PERPH	BLDG/PWR OS LINKS	BLDG INH	CKT LIM	SYS NORM LINKS
CMD: _ 1940 - Easy SU Installation							
9800		Start Execution			9850,F	Dump Inst SU File	
9810,Y		Change Install SU Name			9860	Stop Execution	
9820,{Y NONE}		Change Back Out SU Name			9870	Stop After Soak	ON
9830,{Y NONE}		Change Apply SU Name					
9840,HH,MM		Change SU Soak Interval Timer					
(Y = 10 Character SU Name, NONE = no SU, HH = Hours, MM = Minutes, F = Filename)							
RESPONSE: Easy SU Is Idle							
		Install SU Name			BWM94-0100		
		Back out SU Name			TMP94-0020		
		Apply SU Name			NONE		
		SU Soak Interval Timer			24-00		

Figure 21-7. Page 1940 – Easy SU Installation Menu

Page 1940 Abbreviations:

Inst Install
SU Software Update

Page 1940 Fields

When Page 1940 is displayed, the values for Install SU Name, Back out SU Name, Apply SU Name, and SU Soak Interval Timer fields will be populated with default information. The following describes each of these fields.

- Install SU Name: Name of the SU that should be installed and made official in the 3B20D/3B21D computer. Poke **9810** changes the name of the SU.
- Back out SU Name: Name of the temporary or craft SU to be backed out before the Install SU is to be installed. If no SU is in the soak state, the default value NONE will be displayed. Poke **9820** changes the name of the SU to be backed out.
- Apply SU Name: Name of the temporary or craft SU to be applied after the Install SU is in the official state. The Apply SU Name field always defaults to the value NONE. Poke **9830** changes the name of the SU to be applied.

- **SU Soak Interval Timer:** Defaults to 24 hours and 00 minutes. The SU Soak Interval Timer field will always be displayed. Poke **9840** changes the value.

Page 1940 Commands

The upper area of Page 1940 provides commands used to process SUs. These commands are as follows:

- **9800 – Start Execution.** Poke **9800** starts the execution of the easy SU process. If the Back out SU Name field is populated, it backs out that SU, installs and makes official the Install SU, and applies (if appropriate) the Apply SU. This command must not be entered before the Install SU Name field is populated (Poke **9810**) with a valid SU name.
After execution is started on Page 1940 and its response line states that Easy SU Is In Progress, the page is not updated until it completes successfully or runs into a failure. Look on Page 1960 to find the current status of SUs. Page 1960 is kept up to date while the Easy SU Installation process is executing.
- **9810,Y – Change Install SU Name.** Poke **9810** populates the Install SU Name field. This field can contain any valid 6- or 10-character SU name. It must be populated for the Easy SU Installation process to begin execution. This SU is the one which will be used as the Install SU.
Poke **9810** should always be the first command entered on the page. This is needed since this command initializes the page with default information, except for the data just entered. If the page is set up and this is the last command executed, the page will be initialized and the previous data input will have to be reentered.
- **9820,{Y|NONE} – Change Back Out SU Name.** Poke **9820** populates the Back out SU Name field. This field is populated for the user and should never have to be changed by the user. It can contain all valid SU names except official SUs (SUs that start with BWM). This SU is the one which will be backed out when execution is started.
- **9830,{Y|NONE} – Change Apply SU Name.** Poke **9830** populates the Apply SU Name field. This field can contain all valid SU names except official SUs (SUs that start with BWM). This SU will be applied after the Install SU is finished.
- **9840,HH,MM – Change SU Soak Interval Timer.** Poke **9840** changes the soak interval timer for the Install SU. The default value for the timer is 24 hours. The timer does not change until the SOAK section has been executed and the timer set. At that time, the Easy SU Installation process resets the soak interval timer to the value the user entered.

- **9850,F** – Dump Inst SU File. Poke **9850** prints any ASCII file associated with the Install SU on the ROP. This includes the *MSG*S and *SCANS* files. To print the *MSG*S file for the Install SU with this command, enter **9850,MSG**S.
- **9860** – Stop Execution. Poke **9860** stops the execution of all commands on Page 1940. Execution will be stopped only after the current step in the current state is completed. This command is the same as Poke **9560** on Page 1960.
- **9870** – Stop After Soak. Poke **9870** stops execution of the Easy SU Installation process after the Install SU has started the soak period. If this command is selected, Poke **9870** causes the Stop After Soak status field to show ON, and the execution of Easy SU Installation to stop awaiting user input. To restart the Easy SU Installation process, reenter Poke **9800**.

If Poke **9870** was not entered, the Stop After Soak status field will indicate OFF and the Easy SU Installation process will not stop until the Install SU has been made official.

Poke **9870** toggles the Stop After Soak status field. If it was on, entering Poke **9870** causes it to turn off. If it was off, entering Poke **9870** causes it to turn on.

Page 1940 Example Procedure

The following procedure is an example of using Page 1940. For this example procedure, assume the following:

- A temporary SU TMP94-0020 is soaking.
- An official SU BWM94-0100 is to be installed.
- A temporary SU TMP94-0029 is to be applied after the Install SU has been made official.
- A soak time of 2 hours is wanted for the Install SU.

The following example procedure shows how to use Page 1940:

Step	Procedure
1	Enter Poke 1940 . Result: The Page 1940 is displayed. The Back Out SU Name field is populated with TMP94-0020. The rest of the page contains default information.
2	Enter Poke 9810,940100 . Result: The Install SU Name field is populated with BWM94-0100.
3	Enter Poke 9830,940029 . Result: The Apply SU Name field is populated with TMP94-0029.
4	Enter Poke 9840,2,0 . Result: The SU Soak Interval Timer field will be populated with 2-00.
5	Enter Poke 9800 . Result: Execution of the Easy SU Installation process will be started.

Page 1941 – SU Automation Scheduling

Page 1941 provides a simplified procedure to schedule SU for automatic installation into the 3B20D/3B21D computer. The SU Automation Scheduling feature simplifies the SU installation process by allowing the user to schedule an SU to be automatically installed at a later date and time. This frees the user from entering commands and constantly monitoring progress of the SU.

The SU Automation Scheduling page can be used to perform the following:

- Schedule or reschedule (start time, start date, and/or install state) an SU for automatic installation.
- Resume a stopped automatic installation.
- Toggle SU automation ON (allowed) or OFF (inhibited).
- Execute an office healthcheck.
- Query SU automation status.
- Show the results of the last office healthcheck.
- Cancel the current SU scheduled.

Page 1941 Description

Page 1941 (Figure 21-8) is divided into two areas.

- The upper area of the page contains the commands used to control the SU automation process.
- The lower area of the page shows:
 - Current scheduled SU name
 - Date and time the SU is scheduled to be installed
 - Installation state (VERIFY, SOAK, or OFFICIAL)
 - Current SU automation status (ALLOWED or INHIBITED)
 - Date, time, and status of the last office healthcheck
 - Responses indicating the progress of the SU automation procedures.

SYS EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLDG INH	CKT LIM	SYS NORM
TRAFFIC	SYS INH	SU	SU PERPH	OS LINKS			
					1941 - SU AUTOMATION SCHEDULING		
9920,X	Schedule SU For Automation			9925	Resume Stopped SU Automation		
9921,X	Reschedule SU Name			9926	Inhibit SU Automation		
9922,MMDDYY	Reschedule SU Start Date			9927	Allow SU Automation		
9923,HHMM	Reschedule SU Start Time			9928	Execute Office Healthcheck		
9924,Z	Reschedule SU Install State			9929	Query SU Automation Status		
				9930	Show Last Healthcheck Results		
				9931	Cancel Scheduled SU		
1942	SU Automation Office Profile						
1943	SU Automation Healthcheck						
(X=10 Char SU Name, MMDDYY=Month,Day,Year, HHMM=Hours,Minutes, Z=VFY SOAK OFC)							
SU SCHEDULED BWM93-0001 STATE SCHEDULED10/15/1993 19:30 SU INSTALL STATE OFC							
SU AUTOMATION ALLOWED LAST HEALTHCHECK STATUS10/12/1993 15:46 PASSED							
RESPONSE: SU AUTOMATION SCHEDULED							

Figure 21-8. Page 1941 – SU Automation Scheduling Menu

Page 1941 Abbreviations:

Char	Character
OFC	Official
SU	Software Update
VFY	Verify

Page 1941 Fields

The following describes each of the Page 1941 fields:

- **SU SCHEDULED:** Name of SU currently scheduled for automatic installation or NONE.
- **STATE:** Current state of automation, for example, IDLE, SCHEDULED DATE/TIME, STOPPED, IN PROGRESS, and so forth.
- **SU INSTALL STATE:** Application stage that the currently scheduled SU will be installed to. Application stages are verify (VFY), soak (SOAK), or official (OFC).
- **SU AUTOMATION:** Current status [on (ALLOWED) or off (INHIBITED)]. If the status is INHIBITED, SUs cannot be scheduled for automatic installation.

- **LAST HEALTHCHECK STATUS:** Date and time that the last office healthcheck was run and its result, or it will indicate that a healthcheck is currently in progress.
- **RESPONSE:** Information relative to SU Automation Scheduling.

Figure 21-8 shows that BWM93-0001 is scheduled to be installed officially on 10/15/1993 starting at 19:30. The SU automation feature is allowed, and the status of the last healthcheck (run on 10/12/1993 15:46) is PASSED.

Page 1941 Commands

The upper area of Page 1941 provides commands to schedule SUs. These commands are as follows:

- **9920,X** – Schedule SU For Automation. Poke **9920** schedules an SU for automatic installation. The start day, time, and install state will be derived from the default values listed in the SU Automation Office Profile (see the “Page 1942 – SU Automation Office Profile” section). Note that for the start date/time, the SU will be scheduled for the very next time slot that is NOT off, as listed in the office profile. After this command is successfully accepted, the indicators which show the current SU name, start day/time, and install state will be updated accordingly.
- **9921,X** – Reschedule SU Name. Poke **9921** reschedules an SU for automatic installation. This command is used when a scheduled SU needs to be changed. It will modify only the SU name, not the start day, time, or install state.
- **9922,MMDDYY** – Reschedule SU Start Date. Poke **9922** reschedules the start date for automatic installation. This command is used when the start date of a scheduled SU needs to be changed. The valid argument is MMDDYY, where MM is a two-digit month (01-12), DD is a two-digit day (01-31), and YY is a two-digit year (00-99). An SU can be scheduled up to 49 days in advance.
- **9923,HHMM** – Reschedule SU Start Time. Poke **9923** reschedules the start time for automatic installation. This command is used when the start time of a scheduled SU needs to be changed.
- **9924,Z** – Reschedule SU Install State. Poke **9924** reschedules the install state for automatic installation. This command is used when the install state for a scheduled SU needs to be changed. The valid install states are VFY (verify), SOAK, and OFC (official).
- **9925** – Resume Stopped SU Automation. Poke **9925** resumes an automatic SU installation that was previously stopped or aborted.
- **9926** – Inhibit SU Automation. Poke **9926** inhibits (turns off) SU automation. An SU cannot be scheduled for automatic installation when SU automation is INHIBITED.

- **9927** – Allow SU Automation. Poke **9927** permits (turns on) SU automation. An SU can be scheduled for automatic installation only when SU automation is ALLOWED.
- **9928** – Execute Office Healthcheck. Poke **9928** requests that an office healthcheck be executed. The final result (PASSED or FAILED) will show on this page. The detailed results of this command will be sent to the ROP.
- **9929** – Query SU Automation Status. Poke **9929** reports the status of SU automation (ALLOWED or INHIBITED) and the reason for it being inhibited, if a reason exists.
- **9930** – Show Last Healthcheck Results. Poke **9930** reports to the ROP the detailed results of the last healthcheck that was run.
- **9931** – Cancel Scheduled SU. Poke **9931** cancels an SU that is currently scheduled for automatic installing but has not yet started installation.

Page 1942 – SU Automation Office Profile

Page 1942 provides the current status and a simplified procedure for modifying the office profile. The SU Automation Office Profile feature is used when scheduling an SU for automatic installation, when executing an office healthcheck, and when determining whether the CSCANS is allowed to update the SU Automation Office Profile.

Page 1942 Description

Page 1942 (Figure 21-9) lists all valid fields of the office profile for SU automation (except for the SU automation healthcheck conditions which are listed on Page 1943). It shows the value that is currently set up for each field and provides the commands to reset the value.

At the lower area of the page, the result for each command request is displayed.

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR SU	MINOR SU PERPH	BLDG/PWR OS LINKS	BLDG INH	CKT LIM	SYS NORM
1942 - SU AUTOMATION OFFICE PROFILE							
99YY,Z Modify Office Profile							
YY							
01 Install SU Up To (VFY SOAK OFC)				OFC			
02 Soak Time (HHMM)				24:00			
03 Alarm Level (MAJOR MINOR)				MAJOR			
04 Allow CSCANS Control (Y N)				Y			
05 Advance Warning Message (HHMM)				00:30			
07 Clear SU After SU Is Official (Y N)				Y			
Time Interval Automation Is Not Allowed Each Day (HHMM,HHMM)							
08 Sun 12:00 - 18:30		09 Mon 07:00 - 18:00		10 Tue 07:00 - 18:00		11 Wed 07:00 - 18:00	
12 Thu 07:00 - 18:00		13 Fri 07:00 - 18:00		14 Sat 12:00 - 17:00			
(HH=Hours, MM=Minutes)							
RESPONSE: UPDATE OFFICE PROFILE COMPLETED							

Figure 21-9. Page 1942 – SU Automation Office Profile Menu

Page 1942 Abbreviations:

CSCANS	Customer Service Computer Access Network System
OFC	Official
SU	Software Update
VFY	Verify
Y	Yes or No

Page 1942 Commands

Page 1942 commands are as follows:

- **9901,{VFY|SOAK|OFC}** – Install SU Up To (VFY|SOAK|OFC). Poke **9901** modifies the default SU automation install state.
- **9902,HHMM** – Soak Timer (HHMM). Poke **9902** modifies the default soak time (HH = hours; MM = minutes) that will be used during the soak stage of an SU that is automatically installed.
- **9903,{MAJOR|MINOR}** – Alarm Level (MAJOR|MINOR). Poke **9903** modifies the alarm level that will be used when an office alarm is fired during an SU automation process. The valid values are MAJOR and MINOR.
- **9904,{Y|N}** – Allow CSCANS Control (Y|N). Poke **9904** modifies the CSCANS control field. **Y** indicates that CSCANS is allowed to send commands to modify this office profile. **N** means CSCANS is not allowed to send commands to modify this profile.
- **9905,HHMM** – Advance Warning Message (HHMM). Poke **9905** sets the amount of time prior to starting SU automation to send this warning message. Before SU automation starts, a warning message is printed to the ROP to inform office personnel.
- **9907,{Y|N}** – Clear SU After SU Is Official (Y|N). Poke **9907** indicates whether an SU should be automatically cleared (CLR:SU) after it is automatically installed and made official. **Y** means the SU will be cleared. **N** means the SU will not be cleared.
- **9908,HHMM,HHMM** – Poke **9908** modifies the time interval that SU automation will not be scheduled for (by default) on Sunday.
- **9909,HHMM,HHMM** – Poke **9909** modifies the time interval that SU automation will not be scheduled for (by default) on Monday.
- **9910,HHMM,HHMM** – Poke **9910** modifies the time interval that SU automation will not be scheduled for (by default) on Tuesday.
- **9911,HHMM,HHMM** – Poke **9911** modifies the time interval that SU automation will not be scheduled for (by default) on Wednesday.
- **9912,HHMM,HHMM** – Poke **9912** modifies the time interval that SU automation will not be scheduled for (by default) on Thursday.

- **9913,HHMM,HHMM** – Poke **9913** modifies the time interval that SU automation will not be scheduled for (by default) on Friday.
- **9914,HHMM,HHMM** – Poke **9914** modifies the time interval that SU automation will not be scheduled for (by default) on Saturday.

Page 1943 – SU Automation Healthcheck

Page 1943 provides a simplified procedure for setting (turning ON) and clearing (turning OFF) predefined healthcheck conditions. This page also provides help commands to request further information on a given condition.

Page 1943 Description

Page 1943 (Figure 21-10) allows users to set or clear each SU automation healthcheck condition by using the **4XX** and **5XX** commands, respectively. Detailed information explaining each condition can be retrieved via the **9XX** help command. Conditions that are SET on this page are checked during an office healthcheck. Each SET condition is backlighted on the display. The result of each SET or CLR request is shown on the lower area of the page (RESPONSE).

Each of the five predefined automatic healthcheck conditions is described in Table 21-2.

Table 21-2. Predefined SU Automation Healthcheck Conditions

Number (XX)	Condition	Description
00	3B BREAKPOINTS	Breakpoints in the 3B20D/3B21D computer
01	3B DUPLEX	3B20D/3B21D computer disks in duplex mode
02	SU IS TEMP	Temporary SUs on switch
03	SU INCONSISTNT	SU inconsistencies
04	SCC LINK	Is SCC link active?

SYS EMER TRAFFIC	CRITICAL SYS INH	MAJOR SU	MINOR SU PERPH	BLDG/PWR OS LINKS	BLDG INH	CKT LIM	SYS NORM MISC
1943 - SU AUTOMATION HEALTHCHECK							
4XX SET		5XX CLEAR		9XX HELP			
00	3B BREAKPOINTS	01	3B DUPLEX	02	SU IS TEMP	03	SU INCONSISTNT
04	SCC LINK						
RESPONSE: UPDATE OFFICE PROFILE COMPLETED							

Figure 21-10. Page 1943 – SU Automation Healthcheck Menu

Page 1943 Abbreviation:

SU Software Update

Page 1943 Commands

The commands provided on this page are as follows (where **XX** = 00 – 04):

- **4XX** – Sets healthcheck condition **XX** (turns it ON).
- **5XX** – Clears healthcheck condition **XX** (turns it OFF).
- **9XX** – Requests further information on healthcheck condition **XX**.

Page 1950 – SU Maintenance

Page 1950 provides commands to display SU history, to recover backward and forward, to clear the SU, and to back out the last official SU. The soak period for an SU can be set/changed by the craft (Poke **9700**). Poke **9710** can be used to check the status of a specific SU soak period. Poke **9720** can be used to abort the soak timer.

Page 1950 Description

SU keeps an on-line log of the history of all SUs installed against the current software release issue in a 3B20D/3B21D computer.

The entire Page 1950 is commands. Figure 21-11 is an example that shows the history of BWM94-0067 has been completed.

SYS TRAFFIC	EMER	CRITICAL	MAJOR	MINOR	BLDG/PWR	BLDG INH	CKT LIM	SYS NORM
		SYS INH	SU	SU PERPH	OS LINKS			
CMD: _ 1950 - SU MAINTENANCE								
DISPLAY SU HISTORY								
9101,"Y"	SU	Y	COMPLETED		9200	VERIFY INCONSISTENCY		
9102	OFC				9300	RECOVER FORWARD		
9103	TEMP				9400	RECOVER BACKWARD		
9104	ALL				9600,"Y"	CLEAR SU Y		
					9650,"X"	EXPAND SU		
LAST OFC SU					SOAK TIMER FOR SU =			
2nd FROM TOP			BWM94-0066		9700, HH, MM	RESET TIMER		
3rd FROM TOP			BWM94-0065		9710	PRINT TIMER INFO		
9900 BACKOUT OFC			NONE		9720	ABORT SOAK TIMER		
					9730, HH, MM	SET DEFAULT SOAK TIMER		
						12:00		
(X = Y OR STOP) (Y = 10 CHAR SU NAME) (HH=HOURS) (MM = MINUTES)								

Figure 21-11. Page 1950 – SU Maintenance Menu

Page 1950 Abbreviations:

CHAR Character
OFC Official
SU Software Update
TEMP Temporary

Page 1950 Commands

Page 1950 commands are as follows:

- **9101,Y** – SU Y. Poke **9101** prints the history of the specified SU (Y = SU name) at the ROP (**UPD:UPDDSPY:BWM=Y**).
- **9102** – OFC. Poke **9102** prints a history of all official SUs at the ROP (**UPD:UPDDSPY,OFC**).
- **9103** – TEMP. Poke **9103** prints a history of all temporary state SUs at the ROP (**UPD:UPDDSPY,TEMP**).
- **9104** – ALL. Poke **9104** prints the history of all active log SUs (**UPD:UPDDSPY,ALL**) or archive log SUs (**,BACKLOG**) at the ROP. This command accepts the **COMMAND,DATA!** format.
 - **9104,BACKLOG** prints the history of all the SUs in an archive log.
 - **9104,SUM** prints a summary consisting of the last official software update (SUyy-nnnn), a list of active craft software updates (CFTyy-nnnn), the temporary software update (TMPyy-nnnn) (if one exists), and their respective status.
- **9200** – VERIFY INCONSISTENCY. Poke **9200** detects and prints at the ROP any update inconsistencies (**UPD:VFYCON**).
- **9300** – RECOVER FORWARD. Poke **9300** reapplies all the temporary updates which are in the table of inconsistencies (**UPD:RECOVERY,FRWD**).
- **9400** – RECOVER BACKWARD. Poke **9400** removes all the temporary updates causing inconsistencies starting from the very last one in the system to the first (**UPD:RECOVERY,BKWD**).
- **9600,Y** – CLEAR SU Y. Poke **9600** removes the files in the specified SU package (Y = SU name). This command is used to free disk space after the SU is made official. It removes the files in the specified SU (or ALL for all SUs) (**UPD:CLRBWM:BWM=Y | ALL**).
- **9650,X** – EXPAND SU. Poke **9650** expands the compressed files in the specified SU package (X = SU name). This command is used to expand a compressed SU when the automatic expansion during an SU download through SCANS or CSCANS was manually stopped or aborted (**UPD:EXPAND:UPNM=Y | STOP**).

- **9700,HH,MM** – RESET TIMER. Poke **9700** resets timer for soak period to HH (HH = hour) and MM (MM = minutes) (**UPD:RESET:TM=HH-MM**). There are two applications:
 - **Make SU Official Immediately:** The craft can make the SU official immediately by entering Poke **9700,0,0** after all of the commands have been executed via the soak command (Poke **9320** or **9420** on Page 1960). This command allows the craft to bypass the soak period.
 - **Change time period of the soak:** After the craft has initialized the SU soak period (via Poke **9320** or **9420** on Page 1960), the time period of the soak can be changed by entering Poke **9700,HH,MM**. The time period of the soak can be changed only after the soak period has been initialized. If this occurs, the craft receives an error message indicating that the soak has not been initialized. Only one SU can have its soak period set/changed at a time.
 - **9710** – PRINT TIMER INFORMATION. Poke **9710** prints timer information [completion time, start time, and (if applicable) time the soak period was changed] on the ROP (**UPD:PRINT:SKTM**).
 - **9720** – ABORT SOAK TIMER. Poke **9720** aborts the time that was reset previously (**UPD:RESET:ABORT**).
 - **9730,HH,MM** – SET DEFAULT SOAK TIMER. Poke **9730** sets the default soak timer to **HH** hours and **MM** minutes for the office. The default soak timer is used to set the initial soak timer when manually installing an SU through MCC Page 1960 or Page 1940. It also overwrites the existing soak timer field in the SU Automation Office Profile (MCC Page 1942).
- ⇒ **NOTE:**
This poke will not modify any existing soak timer that is already in progress.
- **9900** – BACKOUT OFC. Poke **9900** backs out the last SU that was made official. The name of the last official SU is displayed on this display page in the LAST OFC SU field (**UPD:BOLO**).

Page 1960 – SU Installation

Page 1960 provides commands for installing SUs and the status of the installation.

Page 1960 Description

Page 1960 (Figure 21-12) consists of two areas.

- The upper area consists entirely of commands that provide the capability to verify an SU, display or print the contents of the message file in the SU, or execute commands in the message file.
- The lower area is the message file display area. The message file is a file in an SU package which contains a set of craft input messages grouped in sections describing how to install the SU. There are up to ten numbered lines for displaying the command lines of the message file.

The craft can select and print the ASCII files with Poke **9260,F** (F = filename in SU) that are associated with a particular SU on the ROP. If the requested file is not an ASCII file, an error message is printed on the ROP.

If Poke **9010** (VERIFY) has been entered, an indicator to its right displays the status of the verification. The three possible phases for the status indicator are:

- INPG – In progress
- CMPL – Complete
- ABRT – Abort.

If verifying the SU fails, further information will be printed on the ROP.

The SECTION EXECUTION STATUS area provides status indicators for each of the execution sections (APPLY, SOAK, OFC, and BKOUT). If there is an error, the status indicator of the section which was executing will show the status ABRT and be backlighted. The error causing command line in the message file display area will also be backlighted (black on red for color terminals).

An additional indicator, timer in progress (TMPG), is displayed in the SOAK indicator block after the soak timer is set.

If all the command lines in the section being executed are successful, the status CMPL will be displayed in the appropriate status indicator block (APPLY **CMPL**, SOAK **CMPL**, and OFC **INPG**) in the SECTION EXECUTION STATUS area.

The RESPONSE indicator is used to display a summary of an error message if an error occurs during the execution of a command line.

Further information can always be found printed at the ROP.

Figure 21-12 is an example of Page 1960 showing that an SU is being made official (name of the SU is in the upper right-hand corner).

SYS TRAFFIC	EMER	CRITICAL SYS INH	MAJOR SU	MINOR SU PERPH	BLDG/PWR OS LINKS	BLD INH	CKT LIM	SYS NORM
CMD: _					1960 - INSTALL SU=BWM94-0001			
9000,Y	START	SU Y	9570	NEXT WINDOW	91XX	DISPLAY (XX =	10	APPLY
9010	VERIFY	CMPL	9575	PREV WINDOW	92XX	PRINT 20	SOAK	30 OFC
9560	STOP	EXC	(Y = 10	CHAR SU NAME)	93XX	EXEC ALL 40	BKOUT	50 FILE)
9565,Z	RESET	LINE Z	(Z = 3	DIGIT LINE NO)	94XX	EXEC NEXT		
			(F =	FILENAME IN SU)	9260,F	PRINT FILE F		
SECTION EXECUTION STATUS								
	APPLY	CMPL	SOAK	CMPL	OFC	INPG	BKOUT	
RESPONSE: _								
11	"							
12	"	OFFICIAL	_	_	_	_	_	_
13	"							
14	"	To make the update applied in this SU official, wait the						
15	"	prescribed SOAK interval and then enter sequentially the						
16	"	following command(s):						
17	"							
18	UPD:UPNM="BWM94-0001":OFC;							
19	"							

Figure 21-12. Page 1960 – SU Installation Menu

Page 1960 Abbreviations:

BKOUT Backout
CHAR Character
CMPL Complete
EXC Execute
EXEC Execute
INPG In-Progress Message
NO Number
OFC Official
PREV Previous
SU Software Update

Page 1960 Commands

In addition to the SU installation commands, any available paging command may be entered from Page 1960.

- **9000,Y** – **START SU Y**. Poke **9000** starts the updating software session using SU Y (**UPD:UPNAME:BWM Y**).
- **9010** – **VERIFY**. Poke **9010** verifies the SU (**UPD:VFYBWM**).
- **9560** – **STOP EXC**. Poke **9560** stops the execution of the SU (**STP:EXC,UPD**).
- **9565,Z** – **RESET LINE Z**. Poke **9565** resets the execution pointer in the message file to command line Z (**UPD:RESET:LINE Z**).
- **9570** – **NEXT WINDOW**. Poke **9570** scrolls the message display area forward to the next window (**UPD:NXTWNDW**).
- **9575** – **PREV WINDOW**. Poke **9575** scrolls the message display area backward to the previous window (**UPD:PRVWNDW**).
- **9110** – **DISPLAY APPLY**. Poke **9110** displays the APPLY section of the SU in the message display area (**UPD:DISPLYUPD,APPLY**).
- **9120** – **DISPLAY SOAK**. Poke **9120** displays the SOAK section of the SU in the message display area (**UPD:DISPLYUPD,SOAK**).
- **9130** – **DISPLAY OFC**. Poke **9130** displays the OFC section of the SU in the message display area (**UPD:DISPLYUPD,OFC**).
- **9140** – **DISPLAY BKOUT**. Poke **9140** displays the BKOUT section of the SU in the message display area (**UPD:DISPLYUPD,BKOUT**).
- **9150** – **DISPLAY FILE**. Poke **9150** displays the beginning of the *MSG*S file of the SU in the message display area (**UPD:DISPLYUPD,FILE**).
- **9210** – **PRINT APPLY**. Poke **9210** prints the APPLY section of the SU at the ROP (**UPD:PRINT,APPLY**).
- **9220** – **PRINT SOAK**. Poke **9220** prints the SOAK section of the SU at the ROP (**UPD:PRINT,SOAK**).
- **9230** – **PRINT OFC**. Poke **9230** prints the OFC section of the SU at the ROP (**UPD:PRINT,OFC**).
- **9240** – **PRINT BKOUT**. Poke **9240** prints the BKOUT section of the SU at the ROP (**UPD:PRINT,BKOUT**).
- **9250** – **PRINT FILE**. Poke **9250** prints the complete *MSG*S file of the SU at the ROP (**UPD:PRINT,FILE**).
- **9260,F** – **PRINT FILE F**. Poke **9260** prints the ASCII files that are associated with a particular SU on the ROP (**UPD:PRINT:BWMFILE,FN F**) (F = filename in SU).

- **9310** – EXEC ALL APPLY. Poke **9310** executes all APPLY command lines one by one (**UPD:EXALL,APPLY**)[**UCL**].
- **9320** – EXEC ALL SOAK. Poke **9320** executes all SOAK command lines one by one (**UPD:EXALL,SOAK**).
- **9330** – EXEC ALL OFC. Poke **9330** executes all OFC command lines one by one to make an SU official (**UPD:EXALL,OFC**)[**UCL**].
- **9340** – EXEC ALL BKOUT. Poke **9340** executes all BKOUT command lines one by one to back out the SU. It is used if the SU fails (**UPD:EXALL,BKOUT**).
- **9410** – EXEC NEXT APPLY. Poke **9410** executes the current APPLY command line (**UPD:EXNXT,APPLY**).
- **9420** – EXEC NEXT SOAK. Poke **9420** executes the current SOAK command line (**UPD:EXNXT,SOAK**).
- **9430** – EXEC NEXT OFC. Poke **9430** executes the current OFC command line (**UPD:EXNXT,OFC**).
- **9440** – EXEC NEXT BKOUT. Poke **9440** executes the current BKOUT command line (**UPD:EXNXT,BKOUT**).

Contents

Introduction	22-1
Installation Procedure	22-5
■ Channel Board Installation Procedure	22-5
■ Software Installation Procedure	22-5
3B20D/3B21D Computer Product Installation	22-5
Sun™ System Product Installation	22-5
■ Removing the Installed Package from the Workstation	22-9
Maintenance Commands for DCI	22-10
■ Terminating the Running DCI Driver	22-10
■ Removing DCI from Service	22-10
■ Restoring DCI to Service	22-10
■ Diagnosing the DCI Board	22-11
SG/ECD Setup	22-12
■ SG Forms	22-12
■ ECD Forms	22-12
<i>ucb</i> Form	22-13
<i>mdct</i> Form	22-14
<i>logdev</i> Form	22-14
<i>chopt</i> Form	22-15
<i>dciopt</i> Form	22-15
<i>device</i> Form	22-17
<i>classdef</i> Form	22-17
<i>errthld</i> Form	22-17
■ Examples of ECD Forms	22-18

Introduction

This chapter provides information on the Dual Serial Channel (DSCH) Computer Interconnect (DCI) feature. The feature is implemented in the Real-Time Reliable (RTR) Release 21.3 and later software. This section presents an overview of the DCI feature, configuring the Equipment Configuration Database (ECD), installation procedures for the *Sun** system, and maintenance commands for the DCI board. It also provides references to other DCI feature related documents.

Unlike other RTR features, the DCI feature impacts both the 3B20D/3B21D computer and the *Sun Workstation**. Additional information regarding standard installation on a workstation running *Solaris** 2.3 can be found in the following:

- ***Solaris 2.3 Developer's Guide to Application Packaging*** (801-5280-10)
- *System Services and Application Programming Guide* (800-5240-10)
- *Hardware Owner's Guide* (801-2175-10)
- ***Solaris 2.3 Basic Installation Guide*** (801-5278-10).

* Sun, Sun Microsystems, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

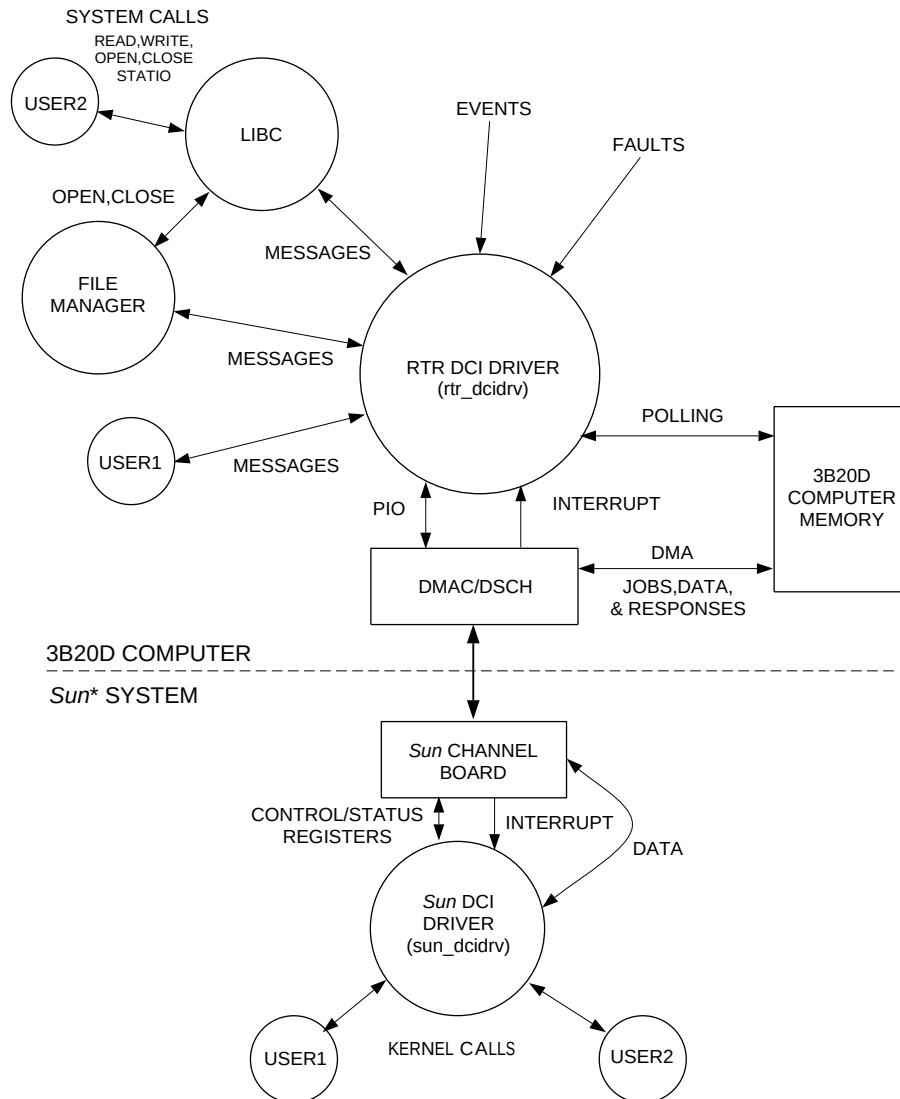
The DCI feature allows an application to off-load the 3B20D/3B21D computer to the *SPARCstation*^{*} by transferring and processing data on the *Sun* system.

To support the DCI feature, the following changes have been implemented both on the 3B20D/3B21D computer and the *SPARCstation*.

- A separate driver has been developed to support the feature on the 3B20D/3B21D computer side.
- A loadable device driver has been added on the *Sun* system.
- The RST, RMV, and DGN maintenance commands have been added.
- A new growth/degrowth procedure has been developed.
- A *Sun* channel board has been developed.
- A set of new DSCH cables has been developed.

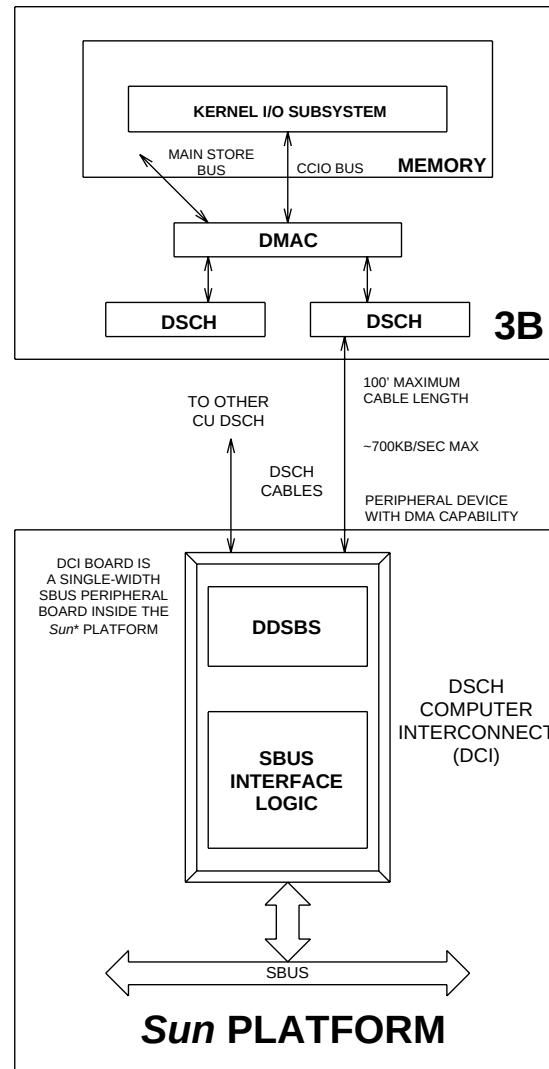
Figures 22-1 and 22-2 display the high-level software and hardware architecture of the DCI feature.

* All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems Inc.



* Sun, Sun Microsystems, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

Figure 22-1. Software Architecture of the DCI Feature



* Sun, Sun Microsystem, the Sun Logo, Sun Workstation, and Solaris are trademarks or registered trademarks of Sun Microsystems Inc. in the United States and other countries.

Figure 22-2. Hardware Architecture of the DCI Feature

Installation Procedure

To support the feature, two new software drivers have been added, one on the *Sun Workstation* and the other on the 3B20D/3B21D computer. A new channel board has been developed for the workstation. This section describes the software and hardware installation procedure.

Channel Board Installation Procedure

As part of the DCI feature, a new *Sun* channel board has been developed. The "SBUS Installation Procedure" described in the *Sun* hardware manual must be followed. Small Computer System Interface Bus (SBUS) slot 0 and slot 2 should be used to install the channel boards.

Each channel board has two connectors. Once installed on the *Sun* system, the channel cables from both Control Units (CUs) must be connected to the channel board connectors. As viewed from the backplane of the workstation, the connector on the left-hand side should be connected to CU 0. The connector on the right-hand side should be connected to CU 1.

Software Installation Procedure

To install the DCI feature, new software must be installed on both the 3B20D/3B21D computer and the *Sun* system.

3B20D/3B21D Computer Product Installation

As part of the DCI feature, a new software driver and a set of new diagnostic products were added. To install and use the DCI feature, both the System Generation (SG) database and the ECD must be modified.

The "SG/ECD Setup" section provides information for the database updates.

***Sun*TM System Product Installation**

To install the package, it must be transferred to an installable media 4MM Digital Audio Tape (DAT) unit tape and/or 3.5-inch floppy disk.

The following command should be used to transfer files from the support processor system to the 4MM DAT tape or 3.5-inch floppy disk:

```
Is ATT3bdci.pkg | cpio -ov > /dev/"device name"
```

To install the Lucent Technologies *UNIX*® RTR 3B DCI (ATT3bdci) package into the *SPARCstation* target machine, perform the following procedure:

Step	Procedure
1	Log in as root on the system console of the target machine.
2	Insert the ATT3bdci installable tape or floppy disk.
3	Execute the following: <pre># cd /var/spool/pkg # cpio -iuv < /dev/"device name" # pkgadd -d ATT3bdci.pkg</pre> The following will be displayed:

```
The following packages are available:  
1  ATT3bdci      Lucent Technologies UNIX RTR 3B Dual Serial Channel Interface
```

```
Select package(s) you wish to process (or 'all' to process  
all packages). (default: all) [?,??,q]:
```


Step	Procedure

4 Enter **1**.

The following will be displayed:

```
Processing package instance <ATT3bdci> from </dev/diskette>

Lucent Technologies UNIX RTR 3B Dual Serial Channel Interface

Copyright (c) 1996 by T
All Rights Reserved

Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
2 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.

The following files are already installed on the system and are being
used by another package:
/etc/devlink.tab
/etc/system
/kernel <attribute change only>
/kernel/drv <attribute change only>
Do you want to install these conflicting files [y,n,?,q]
```

5 Enter **y**.

The following will be displayed:

```
## Checking for setuid/setgid programs.
This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of this package [y,n,?]
```

Step	Procedure

6 Enter **y**.

The following will be displayed:

```
Installing UNIX RTR 3B Dual Serial Channel Interface as <ATT3bdci>

## Installing part 1 of 1.
/kernel/drv/dci
/opt/ATT3bdci/include/dci_const.h
/opt/ATT3bdci/include/dci_ioctl.h
/opt/ATT3bdci/include/dci_perf.h
/opt/ATT3bdci/man/man7/dci.7
[ verifying class <none> ]
Modifying /etc/devlink.tab
Modifying /etc/system
[ verifying class <sed> ]
## Executing postinstall script.

Installation of <ATT3bdci> succeeded.

The following packages are available:
  1  ATT3bdci      Lucent Technologies UNIX RTR 3B Dual Serial Channel Interface

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]:
```

7 Enter **q**.

Removing the Installed Package from the Workstation

To remove the *UNIX* RTR 3B DCI (ATT3bdci) package from the *SPARCstation* target machine, perform the following instructions:

Step	Procedure
1	Log in as root on the system console of the target machine.
2	Execute the following: <pre># cd / # pkgrm ATT3bdci</pre> The following will be displayed:

```
The following package is currently installed:  
ATT3bdci      Lucent Technologies UNIX RTR 3B Dual Serial Channel Interface  
  
Do you want to remove this package [y,n,?,q] y  
  
## Removing installed package instance <ATT3bdci>  
  
This package contains scripts which will be executed with super-user  
permission during the process of removing this package.  
  
Do you want to continue with the removal of this package [y,n,?,q] y
```

- 3 Enter **y**.
The following will be displayed:

```
## Verifying package dependencies.  
## Processing package information.  
## Executing preremove script.  
  
Removal of <ATT3bdci> succeeded.
```

Maintenance Commands for DCI

Terminating the Running DCI Driver

Execute the following command (for PDS) to terminate the DCI driver.

STOP:DCI

An information code of 32c, indicating that the driver has been terminated, will be printed on the Receive-Only Printer (ROP).

Removing DCI from Service

Execute the following command (for PDS) to remove a DCI from service:

RMV:DCI x

Where *x* is the DCI unit number; for example, **DCI 2**.

The status of the DCI should change from Active (ACT) to Out of Service (OOS). A message will be printed on the ROP indicating that the DCI has been removed from service.

Restoring DCI to Service

Execute the following command (for PDS) to diagnose and restore the DCI to service only if diagnostics report All Tests Pass (ATP):

RST:DCI x

Where *x* is the DCI unit number.

Execute the following command (for PDS) to restore the DCI to service unconditionally (no diagnostic tests are performed):

RST:DCI x;UCL

Where *x* is the DCI unit number.

The status of the DCI should change from OOS to ACT and a message will be printed on the ROP indicating that the DCI unit has been restored to service.

Diagnosing the DCI Board

Diagnosing the *Sun* channel board involves running diagnostics on the 3B20D/3B21D computer.

Execute the following commands on the 3B20D/3B21D computer:

```
DGN:DCI x  
SW:CU  
DGN:DCI x
```

Where x is the DCI unit number.

SG/ECD Setup

This section describes the SG and ECD database forms which must be updated to support the DCI feature.

SG Forms

Field 10 (`boot_process.pcreate`) of the *sge2* form in the SG database must be updated with `"/prc/bdj."` To bring up the driver on boot, the *boot image* must be built.

ECD Forms

The following ECD forms are used:

<i>ucb</i>	The <i>ucb</i> form (Figures 22-3, 22-4, and 22-5) describes a physical device.
<i>mdct</i>	The <i>mdct</i> form (Figures 22-6, 22-7, and 22-8) contains information necessary to link a logical device to one or more physical devices.
<i>logdev</i>	The <i>logdev</i> form (Figures 22-9, 22-10, and 22-11) specifies the information about a particular logical device.
<i>chopt</i>	The <i>chopt</i> form (Figure 22-12) is used as the DCI option record.
<i>dciopt</i>	The <i>dciopt</i> form (Figure 22-13) replaces the <i>chopt</i> form as the DCI option record for <i>UNIX</i> RTR Operating System Release 21.17 and later.
<i>device</i>	The <i>device</i> form (Figure 22-14) defines a spooler output device or log files.
<i>classdef</i>	The <i>classdef</i> form (Figure 22-15) specifies a list of devices or log files that are to be associated with a particular spooler output class.
<i>errthld</i>	The <i>errthld</i> form (Figure 22-16) associates hardware/software errors with the error counter. It also specifies thresholds to be used with error counters.

ucb Form

One *ucb* form (Figure 22-3) must be inserted for each DCI.

- For *UNIX* RTR Operating System Release 21.3, fields 11 (*device_port*) and 12 (*device_type*) must be set to **SNIC**.
For *UNIX* RTR Operating System Release 21.7 and later, fields 11 (*device_port*) and 12 (*device_type*) must be set to **DCI**.
- Field 13 (*internal_device*) should have a value from 0 to 3. They should be assigned in numerical order. This version will support only four DCI devices.
- Fields 23 (*device_num*) and 28 (*interrupt_num*) should be set to the port number of the DSCH that the DCI is connected to.
- Field 27 (*mv*) should be set to **1** for the 3B20D computer and to **2** for the 3B21D computer.
- Field 30 (*pathname*) should be set to **pu/dci** for diagnostics.
- Field 56 (*error_table*) should be set to a new error threshold form, **dc ierr**.
- Field 57 (*option_name*) must have the name of an *chopt* form. If *UNIX* RTR Operating System Release is 21.17 or later, use the *dciopt* form instead of the *chopt* form. In Figure 22-3, field 57 is set to **dciopt0**.

For DCIs using the High Availability-Operations and Maintenance Platforms (HA-OMP) feature, the *ucb* form (Figure 22-4) needs the following information:

- Field 39 (*down_unit_name*) should be set to the same string as field 3 (*k_unit_name*) of the *ucb* record for the corresponding adjunct.
- Field 40 (*down_unit_number*) should be set to the same number as field 4 (*k_unit_number*) of the *ucb* record for the corresponding adjunct.

If using the HA-OMP feature, one *ucb* form (Figure 22-5) must be inserted for each Adjunct.

- Field 3 (*k_unit_name*) must be equal to field 39 (*down_unit_name*) of the corresponding DCI *ucb* record.
- Field 4 (*k_unit_number*) must be equal to field 40 (*down_unit_number*) of the corresponding DCI *ucb* record.
- Field 11 (*device_port*) must be set to **DCI**.
- Field 12 (*device_type*) must be set to **SOFT**.
- Field 49 (*unit_name*) should be set to the same string as field 3 (*k_unit_name*) of the *ucb* record for the corresponding DCI.
- Field 50 (*unit_number*) should be set to the same number as field 4 (*k_unit_number*) of the *ucb* record for the corresponding DCI.

***mdct* Form**

One *mdct* form (Figure 22-6) is needed for each DCI to handle normal operations. One *mdct* form (Figure 22-7) is also needed for diagnostic requests.

- Field 1 (*mdct_name*) for the diagnostic *mdct* form must be **dcidX**, where *X* is a number between 0 and 3.
- Field 2 (*logical_device*) must be set to **dc** to indicate the DCI device.
- Field 3 (*mdct_type*) is set to **T** for a normal operation *mdct* form and to **d** for a diagnostic *mdct* form.

Another *mdct* form (Figure 22-8) is needed to support I/O to the active and standby adjuncts in an HA-OMP configuration.

- Field 1 (*mdct_name*) must be some unique name; for example, **dcio**.
- Field 2 (*logical_device*) must be set to **dc** to indicate a DCI device.
- Field 3 (*mdct_type*) must be set to **j** for an HA-OMP configuration.
- Fields 8 (*active_devices unit_name*) and 9 (*active_devices unit_number*) are set to the same values as fields 3 (*k_unit_name*) and 4 (*k_unit_number*) of the two Adjunct *ucb* records.

Each HA-OMP *mdct* record must have a set of *logdev* records which refer to that *mdct* record.

***logdev* Form**

One *logdev* form (Figure 22-9) is needed for each DCI to run diagnostics.

- Field 1 (*logical_name*) should be **/dgn/dcdx**, where *x* is the field 4 (*k_unit_number*) value defined in the *ucb* form.
- Field 5 (*dcn*) should be set to **9** to match the name of the driver as “bdj.”
- Field 6 (*driver_type*) should be set to **r** for this diagnostic *logdev* form.

For normal *logdev* forms (Figure 22-10),

- Field 1 (*logical_name*) should begin with **/dev**.
- Field 2 (*partition*) should be set to the logical channel number to be used for this logical device. Two *logdev* forms with the same field 1 should not be opened at the same time.

For HA-OMP configurations (Figure 22-11), *logdev* records will contain the following information:

- Field 1 (*logical_name*) should begin with **/dev**.
- Field 2 (*partition*) should be set to a value with the range of 0x0 to 0xFF and 0x400 to 0x4FF. Partition values within the range of 0x0 and 0xFF will be for channels accessing the active adjunct. Partition values

within the range of 0x400 to 0x4FF will be for channels accessing the standby adjunct

- Field 3 (*mdct_name*) must be the same as field 1 (*mdct_name*) of the *mdct* record for the HA-OMP Adjuncts.

***chopt* Form**

At least one *chopt* form (Figure 22-12) is used as the DCI option record. Only three fields are used by the DCI driver.

- Field 4 (*poll*) is the minimum polling period in milliseconds. The minimum value is **10** msec.
- Field 5 (*time_out*) is the maximum polling period in milliseconds.
- Field 6 (*picconf*) is the period, in seconds, to attempt to restore a DCI that is removed by the *rtr_dcdrv* program because of nondestructive errors (such as job time-out). The default value is **120** seconds. The minimum value is **30** seconds. If the field is set to **0**, auto restore will be disabled.

***dciopt* Form**

The *dciopt* form (Figure 22-13) will replace the *chopt* form starting with UNIX RTR Operating System Release 21.17. DCIs connected to different adjuncts cannot share the same *dciopt* form because each adjunct would overwrite the other's adjunct state (field 7).

- Field 1 (*option_name*) is the option block name and is the same as field 57 (*option_name*) on the *ucb* form of the corresponding DCI.
- Field 2 (*adj_prCSR_id*) is the ID of the adjunct processor to which the 3B20D/3B21D computer is attached via the DCI link. Acceptable ID values are 0 through 3.
- Field 3 (*min_poll_intv*) is the minimum amount of time (in milliseconds) the adjunct's DCI driver will wait before checking the job queue. If a job is not found, the driver increases this wait time by 10 milliseconds until a job is found or until it reaches the maximum polling interval (*max_poll_intv*).

For UNIX Real-Time Reliable (RTR) Operating System Release 21.31 and 21.33, on the 3B21E platform (VCDX and DRM), this field is the time in seconds that a heartbeat job has to complete before a heartbeat timeout will occur. The value in this field is offset by 10, so a value of 25 will have a timeout of 15 seconds. If, however, the value in this field is less than 15, the effective timeout period will be 15 seconds.

Field value	Effective value
250	240
16 - 249	Field value - 10
15	5
10 - 14	15

The value for this field should be only be altered in consultation with Customer Technical Support.

For UNIX Real-Time Reliable (RTR) Operating System Release 21.35 and later, field 7 (heartbeat_timeout) will be used to indicate the time in seconds that a heartbeat job has to complete before a heartbeat timeout will occur.

- Field 4 (max_poll_intv) is the maximum amount of time (in milliseconds) the adjunct's DCI driver will wait before checking the job queue.

For UNIX Real-Time Reliable (RTR) Operating System Release 21.31 and 21.33, on the 3B21E platform (VCDX and DRM), this field is the number of consecutive times the DCI may timeout before it will be removed from service. The value in this field is offset by 10, so a value of 13 means that the DCI will be removed from service if three consecutive heartbeat timeouts occur without any successful data transfers. If the value in this field is greater than 60, however, the effective value will be 3.

Field value	Effective value
61 - 250	3
60	50
11 - 59	Field value - 10
10	0

The value for this field should be only be altered in consultation with Customer Technical Support.

For UNIX Real-Time Reliable (RTR) Operating System Release 21.35 and later, field 8 (heartbeat_threshold) will be used to indicate the number of consecutive times the DCI may timeout before it will be removed from service.

- Field 5 (auto_restore) is the amount of time (in seconds) the 3B20D/3B21D computer DCI driver waits before attempting to restore a DCI device previously removed by the driver because of a non-3B impacting error.
- Field 6 (dcitype) specifies whether or not the DCI configuration is a HA-OMP. A value of **1** means the DCI configuration is an HA-OMP, if not, the value should be set to **0**.

- Field 7 (`heartbeat_timeout`) specifies the time in seconds that a heartbeat job has to complete before a heartbeat timeout will occur. This field is used in UNIX Real-Time Reliable (RTR) Operating System Release 21.35 and later. The value for this field should be only be altered in consultation with Customer Technical Support.
- Field 8 (`heartbeat_threshold`) specifies the number of consecutive times the DCI may timeout before it will be removed from service. This field is used in UNIX Real-Time Reliable (RTR) Operating System Release 21.35 and later. The value for this field should be only be altered in consultation with Customer Technical Support.

***device* Form**

A *device* form (Figure 22-14) is needed to have the DCI driver errors go to its own error log file *DCDRVLOG*. It is the same as the file except for field 1 (`logical_devname`) and field 2 (`path`). Field 1 is ***DCDRVLOG***, and Field 2 is ***/etc/log/DCDRVLOG***.

***classdef* Form**

The DCI system uses the field 1 (`class_name`) value of **10** as the message class. The *classdef* form (Figure 22-15) was modified to include the error log device *DCDRVLOG* in the list of devices.

***errthld* Form**

An error threshold form, *errthld* (Figure 22-16), is inserted for the DCI as part of the feature. It is supplied by the *UNIX* RTR operating system.

Examples of ECD Forms

Figures 22-3 through 22-16 are examples of ECD forms. The figures illustrate how to configure the database to use the DCI feature. In the examples, the *ucb* form for the DCI device has been inserted after the *ucb* form for DFC 2 device.

ucb (1/4)

(Recent Change and Verify)

Information About Each Physical Device

1.k_complex_name:_____	2.k_complex_number:____
3.k_unit_name:DCI	4.k_unit_number:0
ucb_type	5.essential:NEVER
	6.top:y
	7.unique_unit:y
	8.pseudo_node:n
	9.restorable:y
	10.removable:n
11.device_port:SNIC	12.device_type:SNIC
13.internal_device:0	14.plant:y
15.critical:n	16.portswitch:n
	17.possswitch:_

ucb (2/4)

minor_device_status	18.boot:n	19.rexinh:y	20.errlog:n
21.major_status:00S	22.equipage:0x4		
23.device_num:1	24.channel_num:16		
25.hv:_____	26.mt:_____	27.mv:0x1	
28.interrupt_num:1	29.service_intrpt:___		
30.pathname:pu/dci			
31.location			
32.floor:1	33.aisle:a	34.plate:1	35.pack:1

Figure 22-3. Example of DCI *ucb* Form (Sheet 1 of 2)

ucb (3/4)				
	complex_name	complex_number	unit_name	unit_number
36.down	37)_____	38)_____	39)_____	40)_____
41.side	42)_____	43)_____	44)_____	45)_____
46.cntrl_0	47)CU	48)0	49)CH	50)16
51.cntrl_1	52)CU	53)1	54)CH	55)16
56.error_table:dciterr		57.option_name:dciopt0		
58.pcid:0		59.pcsdid:0		60.h_type:_____
61.handler_id:_____				

ucb (4/4)		
62.packcode:_____	63.issue:_____	64.version:_____
65.cvaddr:_____	66.u_model:_____	

Figure 22-3. Example of DCI ucb Form (Sheet 2 of 2)

ucb			
(1/4)			
Information About Each Physical Device			
(Recent Change and Verify)			
1.k_complex_name:_____	2.k_complex_number:____		
3.k_unit_name:DCI	4.k_unit_number:0		
ucb_type	5.essential:NEVER	6.top:y	7.unique_unit:y
	8.pseudo_node:n	9.restorable:y	10.removable:n
11.device_port:DCI	12.device_type:DCI		
13.internal_device:0	14.plant:y		
15.critical:n	16.portswitch:n	17.posswitch:_____	

ucb			
(2/4)			
minor_device_status	18.boot:n	19.rexinh:y	20.errlog:n
21.major_status:00S	22.equipage:0x4		
23.device_num:1	24.channel_num:16		
25.hv:_____	26.mt:_____	27.mv:0x1	
28.interrupt_num:1	29.service_intrpt:___		
30.pathname:pu/dci			
31.location	32.floor:1	33.aisle:a	34.plate:1
			35.pack:1

Figure 22-4. Example of HA-OMP DCI *ucb* Form (Sheet 1 of 2)

ucb (3/4)				
	complex_name	complex_number	unit_name	unit_number
36.down	37)_____	38)_____	39)ADJ	40)0
41.side	42)_____	43)_____	44)_____	45)_____
46.cntrl_0	47)CU	48)0	49)CH	50)16
51.cntrl_1	52)CU	53)1	54)CH	55)16
56.error_table:dciterr		57.option_name:dciopt0		
58.pcid:0		59.pcsdid:0		60.h_type:_____
61.handler_id:_____				

ucb (4/4)		
62.packcode:_____	63.issue:_____	64.version:_____
65.cvaddr:_____	66.u_model:_____	

Figure 22-4. Example of HA-OMP DCI *ucb* Form (Sheet 2 of 2)

ucb (1/4)			
Information About Each Physical Device (Recent Change and Verify)			
1.k_complex_name:_____	2.k_complex_number:____		
3.k_unit_name:ADJ	4.k_unit_number:0		
ucb_type	5.essential:NEVER	6.top:n	7.unique_unit:y
	8.pseudo_node:y	9.restorable:n	10.removable:n
11.device_port:DCI	12.device_type:SOFT		
13.internal_device:0	14.plant:n		
15.critical:n	16.portswitch:n	17.posswitch:_____	

ucb (2/4)			
minor_device_status	18.boot:n	19.rexinh:y	20.errlog:n
21.major_status:00S	22.equipage:		
23.device_num:	24.channel_num:		
25.hv:_____	26.mt:_____	27.mv:_____	
28.interrupt_num:___	29.service_intrpt:___		
30.pathname:_____			
31.location	32.floor:	33.aisle:	34.plate:
			35.pack:

Figure 22-5. Example of HA-OMP Adjunct *ucb* Form (Sheet 1 of 2)

ucb (3/4)				
	complex_name	complex_number	unit_name	unit_number
36.down	37)_____	38)_____	39)_____	40)_____
41.side	42)_____	43)_____	44)_____	45)_____
46.cntrl_0	47)_____	48)_____	49)DCI	50)0
51.cntrl_1	52)_____	53)_____	54)_____	55)_____
56.error_table:_____		57.option_name:_____		
58.pcid:0		59.pcsdid:0		60.h_type:_____
61.handler_id:_____				

ucb (4/4)		
62.packcode:_____	63.issue:_____	64.version:_____
65.cvaddr:_____	66.u_model:_____	

Figure 22-5. Example of HA-OMP Adjunct *ucb* Form (Sheet 2 of 2)

mdct (1/3)			
Logical To Physical Device Transaction (Recent Change and Verify)			
1.mdct_name: dci0		2.logical_device: dc	
3.mdct_type: T		4.mdct_id_char: 0	
5.active_devices			
6.complex_name	7.complex_number	8.unit_name	9.unit_number
1) _____	_____	DCI	0
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____

mdct (2/3)			
10.backup_devices			
11.complex_name	12.complex_number	13.unit_name	14.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____
15.spare_devices			
16.complex_name	17.complex_number	18.unit_name	19.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____

mdct (3/3)			
20.oos_devices			
21.complex_name	22.complex_number	23.unit_name	24.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____
25.gettyid: _____			

Figure 22-6. Example of Normal Operations *mdct* Form

mdct (1/3)				
(Recent Change and Verify)				
Logical To Physical Device Transaction				
1.mdct_name:dcid0	2.logical_device:dc			
3.mdct_type:d	4.mdct_id_char:0			
5.active_devices				
6.complex_name	7.complex_number	8.unit_name	9.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	

mdct (2/3)				
10.backup_devices				
11.complex_name	12.complex_number	13.unit_name	14.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	
15.spare_devices				
16.complex_name	17.complex_number	18.unit_name	19.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	

mdct (3/3)				
20.oos_devices				
21.complex_name	22.complex_number	23.unit_name	24.unit_number	
1) _____	_____	_____	_____	
2) _____	_____	_____	_____	
3) _____	_____	_____	_____	
4) _____	_____	_____	_____	
25.gettyid:_____				

Figure 22-7. Example of Diagnostic *mdct* Form

mdct (1/3)			
Logical To Physical Device Transaction (Recent Change and Verify)			
1.mdct_name:dupladj0		2.logical_device:dc	
3.mdct_type:j		4.mdct_id_char:h	
5.active_devices			
6.complex_name	7.complex_number	8.unit_name	9.unit_number
1) _____	_____	ADJ	0
2) _____	_____	ADJ	1
3) _____	_____	_____	_____
4) _____	_____	_____	_____

mdct (2/3)			
10.backup_devices			
11.complex_name	12.complex_number	13.unit_name	14.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____
15.spare_devices			
16.complex_name	17.complex_number	18.unit_name	19.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____

mdct (3/3)			
20.oos_devices			
21.complex_name	22.complex_number	23.unit_name	24.unit_number
1) _____	_____	_____	_____
2) _____	_____	_____	_____
3) _____	_____	_____	_____
4) _____	_____	_____	_____
25.gettyid:_____			

Figure 22-8. Example of HA-OMP Adjunct *mdct* Form

Logical Device Information	logdev (1/1) (Recent Change and Verify)
1.logical_name:/dgn/dcd0	2.partition:0x0
3.mdct_name:dcid0	4.channel_id:___
5.dcn:9	6.driver_type:r
7.gettyid:_____	8.usr:rwx
9.grp:rwx	10.other:rwx

Figure 22-9. Example of Diagnostic *logdev* Form

Logical Device Information	logdev (1/1) (Recent Change and Verify)
1.logical_name:/dev/dci0_0	2.partition:0x0
3.mdct_name:dci0	4.channel_id:___
5.dcn:9	6.driver_type:i
7.gettyid:_____	8.usr:rwx
9.grp:rwx	10.other:rwx

Figure 22-10. Example of Non-HA-OMP *logdev* Form

Logical Device Information	logdev (1/1) (Recent Change and Verify)
1.logical_name:/dev/adjh0_1024	2.partition:0x400
3.mdct_name:dupladj0	4.channel_id:___
5.dcn:9	6.driver_type:i
7.gettyid:_____	8.usr:rwx
9.grp:rwx	10.other:rwx

Figure 22-11. Example of HA-OMP *logdev* Form

Option Block For IOP Controller	chopt	(1/1) (Recent Change and Verify)
1.option_name:dciopt0	2.iop_id:0	
3.handler_id:___	4.poll:10	
5.time_out:250	6.picconf:30	
7.pcoos:0xffff	8.comoos:0xf000	
9.intvps:40	10.dgnflg:___	

Figure 22-12. Example of *chopt* Form

DCI Option Block Record	dciopt	(1/1) (Recent Change and Verify)
1.option_name:dciopt0	2.adj_prcsr_id:0	
3.min_poll_intv:10	4.max_poll_intv:250	
5.auto_restore:30	6.dcitype:1	
7.heartbeat_timeout:15	8.heartbeat_threshold:3	

Figure 22-13. Example of *dciopt* Form

Device Definition Record		device (1/3) (Recent Change and Verify)	
1.logical_devname:DCDRVLOG			
2.path:/etc/log/DCDRVLOG			
3.seqn:n	4.logfile:y	5.temp:n	6.priority:n
7.maxq:50	8.blksize:512		
9.linelen:0	10.logflimit:15000		
11.time_stamp:msgon	12.multiple_msg:y		

		device (2/3)	
13.msg_head			
head1	head2	head3	head4
x0	x0	x0	x0
head5	head6	head7	head8
x0	x0	x0	x0
head9	head10	head11	head12
x0	x0	x0	x0
head13	head14	head15	head16
x0	x0	x0	x0
30.msg_trailer			
tr1	tr2	tr3	tr4
x0	x0	x0	x0
tr5	tr6	tr7	tr8
x0	x0	x0	x0
tr9	tr10	tr11	tr12
x0	x0	x0	x0
tr13	tr14	tr15	tr16
x0	x0	x0	x0

		device (3/3)	
47.printer_page:0	48.paging_limit:6	49.protocol:no_prot	
50.dialogue:n	51.alt_device:_____	52.page_per_rpt:n	
53.language:ENGL			

Figure 22-14. Example of device Form

Class Definition Record		classdef	(1/1) (Recent Change and Verify)
1.class_name:10	2.alarm:y		
3.device_list	logical_devname	8.rep_dest:_____	
	1)rop0	11)_____	
	2)ttyz	12)_____	
	3)DCDRVLOG	13)_____	
	4)_____	14)_____	
	5)_____	15)_____	
	6)_____	16)_____	
	7)_____	17)_____	
	8)_____	18)_____	
	9)_____	19)_____	
	10)_____	20)_____	

Figure 22-15. Example of *classdef* Form

		errthld		(1/3)					
Error Threshold Values		(Recent Change and Verify)							
1.error_table:dciterr									
2.short_timer:300000		3.long_timer:3600000							
4.err_values									
	errval1	errval2	errval3	errval4	errval5	errval6	errval7	errval8	errval9
1)0	1	2	3	4	5	6	7	8	
2)9	10	11	12	13	14	15	16	17	
3)18	19	20	21	22	23	24	25	26	
4)27	28	29							
5)									
6)									

		errthld		(2/3)	
14.thr_table					
	sthr	lthr		sthr	lthr
1)3		12	14)3		12
2)3		12	15)3		12
3)3		12	16)3		12
4)3		12	17)3		12
5)3		12	18)3		12
6)3		12	19)3		12
7)3		12	20)3		12
8)3		12	21)3		12
9)3		12	22)3		12
10)3		12	23)3		12
11)3		12	24)3		12
12)3		12	25)3		12
13)3		12	26)3		12
				27)3	12
				28)3	12
				29)3	12
				30)3	12
				31)	
				32)	
				33)	
				34)	
				35)	
				36)	
				37)	
				38)	
				39)	

		errthld		(3/3)	
14.thr_table(cont)					
	sthr	lthr		sthr	lthr
40)			45)		
41)			46)		
42)			47)		
43)			48)		
44)			49)		
				50)	
				51)	
				52)	
				53)	
				54)	

Figure 22-16. Example of *errthld* Form

Contents

Introduction	23-1
Scope of the UUCP Feature Chapter	23-2
■ Definition of Terms	23-2
Supported/Unsupported UUCP Commands	23-3
■ Supported UUCP Commands	23-3
■ Unsupported UUCP Commands	23-4
Feature Activation/Deactivation	23-4
■ Introduction to Activating the UUCP utility	23-4
■ Activating the UUCP Utility Procedure	23-5
■ Deactivating the UUCP Utility Procedure	23-6
Temporary Deactivation	23-6
Permanent Deactivation	23-6
■ 3B21D Computer UUCP Security	
Considerations	23-6
Local System Accessibility	23-7
Conversation Sequence Numbers	23-8
Local Directory and File Accessibility	23-8
File System Integrity	23-9
Operation of UUCP	23-10
■ Hardware Connections	23-10
Direct Connection	23-11
Datakit ® VCS Connection	23-11
■ UUCP Software	23-11
Directories and Their Purpose	23-12
Software Programs and Their Purpose	23-13

Contents

Supported User Programs	23-14
UUCP Daemons and Their Purpose	23-14
How UUCP Operates in the Passive Machine Configuration	23-15
Operation of UUCP on a Local Machine	23-15
Operation of UUCP on a Remote Machine	23-15
Operation of uux on a Local Machine	23-15
Operation of uux on a Remote Machine	23-15
Administrative Programs	23-16
Supporting Database Files and Their Purpose	23-17
UUCP Administration	23-18
■ Introduction to the UUCP Configuration Files	23-18
Config File	23-19
Systems File	23-20
Permissions File	23-20
REQUEST Keyword Rule	23-21
SENDFILES Keyword Rule	23-21
READ and WRITE Keyword Rules	23-22
NOREAD and NOWRITE Keyword Rules	23-22
CALLBACK Keyword Rule	23-23
COMMANDS Keyword Rule	23-23
VALIDATE Keyword Rule	23-24
MYNAME Keyword Rule	23-24
PUBDIR Keyword Rule	23-24
Sample Permissions Files	23-24
Maxuuxqts File	23-25
remote.unknown	23-25
■ Administrative Tasks	23-26
Cleanup of Undeliverable Jobs	23-26
Cleanup of the Public Area	23-26
Machine Log Files	23-26
Compaction of Log Files	23-27
uuclean.sh and cron	23-27
■ Setting Up UUCP Configurations	23-28
UUCP Logins and Passwords Administration	23-28
■ Example of the UUCP Configuration Setup	23-29
■ UUCP Setup for Direct Connection - An Example	23-33
Setup on Remote Machine Running Traditional UNIX ® System	23-34

Contents

Systems File	23-34
Devices File	23-34
Dialers File	23-35
Dialcodes File	23-35
Setup on the 3B21D Computer	23-35
Systems File	23-35
Permissions File	23-35
Update /etc/passwd File on the 3B21D Computer	23-36
■ UUCP Setup for Datakit VCS Connection - An Example	23-36
Setup on the Remote Machine	23-36
Systems File	23-36
Devices File	23-36
Dialers File for Datakit VCS	23-37
Dialcodes File	23-37
Setup on the 3B21D Computer	23-37
Systems File	23-37
Permissions File	23-38
Update /etc/passwd File on the 3B21D Computer	23-38
UUCP Security	23-39
■ Planning and Recommended Precautions	23-39
■ 3B21D Computer UUCP Security	23-39
Troubleshooting	23-41
Error Code Interpretation	23-45
■ Error Messages	23-45
ASSERT Messages	23-45
STATUS Messages	23-47
Error Codes and Meaning	23-50
uuccheck Messages and Corrections	23-53
UUCP Administrative Files	23-63
■ Temporary (TM.) File	23-63
■ Lock (LCK) File	23-63
■ Work (C.) File	23-63
■ Data (D.) File	23-64
■ Execute (X.) File	23-64
UUCP Manual Pages	23-65

Introduction

The UUCP (*UNIX** System to *UNIX* System Copy) support on the 3B21D computer provides a method for moving files to/from the 3B21D computer or executing commands from remote sites on the 3B21D computer. The *UNIX* system that is sending files or receiving files must be a traditional *UNIX* system or the *UNIX* Real-Time Reliable (RTR) operating system running on a 3B21D computer.

Some possible uses for UUCP are as follows:

- Distribute software maintenance files; for example, Software Update (SU) package or a Broadcast Warning Message (BWM) package from a remote central site.
- Transmit requests to the 3B21D computer to run routine administrative or bookkeeping commands.

As implemented on the 3B21D computer, the UUCP package is a subset of the complete UUCP commands and supports only the “g” or “G” protocol.

The UUCP commands are supported only on the 3B21D computer running on 5ESS® switch software releases 5E14 FR4 load, 5E15, or later. The corresponding RTR loads are 21.0.115, 21.35, and later. The UUCP commands are supported only on the UN582. The UUCP commands are not supported on the 3B20D computer or Very Compact Digital Exchange (VCDX) platforms.

* UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

⇒ **NOTE:**

The location of the `uucp spool` directory is different based on the software release running on the 5ESS switch as follows:

- For the 5E15 software release 5E15, the *spool* directory is located under */var*.
- For the 5E14 software release, the *spool* directory is located under */etc/bwm*.

This information product (IP) is written for the 5E15 software release and references only the */var* directory. If the 5ESS switch is running on the software release 5E14, the user must convert all references of */var* to */etc/bwm*.

Scope of the UUCP Feature Chapter

This chapter is intended for the UUCP system administrator and UUCP user. The following assumptions are made in this chapter:

- The UUCP system administrator is an experienced network administrator with some practical knowledge of setting up networks and adding hardware to the computer.
- The UUCP system administrator is knowledgeable in setting up UUCP configuration files on the traditional *UNIX* systems.
- The UUCP user is expected to use UUCP over an RS-232 connection or a *Datakit*® Virtual Circuit Switch (VCS) network.
- The remote system may support additional UUCP commands.

Definition of Terms

The following is a list of terms used in this chapter.

UN582	A Multipurpose Controller (MPC) residing on the 3B21D computer platform configured as asynchronous communication.
UUCP	A group of programs and files that allows files to be copied between <i>UNIX</i> systems and the capability to run commands on remote systems.
active system	A remote system with the UUCP utilities and the hardware required to establish communication links.
local system	A system on the “near” end of a communication link. The 3B21D computer is always a local system.

network	A group of systems that are set up to exchange information and resources.
passive system	A system that has the UUCP utilities which can be called by remote systems but can never originate calls of its own. The 3B21D computer is always a passive system.
Predefined connection	In a <i>Datakit</i> VCS predefined connection, known as PreDefined Destination (PDD), the destination is predefined so that a call is set up automatically as soon as the end devices come on line.
remote system	A system on the “far” end of a communication link; a system which communicates with the 3B21D computer.
Virtual Circuit Switch	A Virtual Circuit Switch (VCS) connection between communicating endpoints that are established on demand.

Supported/Unsupported UUCP Commands

Supported UUCP Commands

The UUCP package on a 3B21D computer *UNIX* RTR system environment supports the following commands:

rmail	The rmail command sends mail to users and is used by UUCP commands.
uuccheck	The uuccheck command checks for the presence of UUCP directories, programs, and support files and verifies the consistency between the permissions and password files.
uuclean.sh	The uuclean.sh is a shell script that performs various cleanup activities on files and directories related to the uucp programs. uuclean.sh is usually run by the cron process.
uucleanup	The uucleanup command cleans up the UUCP <i>spool</i> directory by removing entries older than a selectable number of days to prevent the <i>spool</i> directory from growing too large.
uucp	The uucp command queues files for transfer from one <i>UNIX</i> system to another <i>UNIX</i> system.
uulog	The uulog shell script queries the UUCP log files.
uuname	The uuname command displays a list of all remote systems the local system can contact.

uusetup.sh	The uusetup.sh shell script runs interactively to set up an initial UUCP configuration.
uustat	The uustat network status command displays the job status of requested transfers.
uux	The uux command creates the work, data, and execute files needed to run commands on a remote system.

Unsupported UUCP Commands

Not all UUCP commands have been supported on the 3B21D computer environment. The UUCP commands not supported on the 3B21D computer are **uuto**, **uugetty**, **Uutry**, **uupick**, and **uukick**. However, these commands may be supported on the remote system.

Additional unsupported commands are **rnews**, **cu**, **uuencode**, and **uudecode**.

Initiating outbound connections and polling are ***not supported*** on the current 3B21D computer implementation.

Feature Activation/Deactivation

Introduction to Activating the UUCP utility

The **uusetup.sh** is an interactive script used to activate the UUCP utility. The **uusetup.sh** script asks for information about remote systems and adds entries to the *Systems* and *Permissions* files for those systems as well as login accounts for file transfers.

The **uusetup.sh** script will enter information that conforms to the recommended security practices for UUCP. The script may also be run at any time to add authorized new systems. The **uusetup.sh** script expects the file entries to be in the same format created by **uusetup.sh**. If the file has been edited by hand, it could confuse **uusetup** and defeat its security checks; however, **uucheck** is the definitive check of the configuration. The **uusetup.sh** script should be executed by the ROOT login. The ROOT login is denoted by the presence of the # sign prompt in this procedure.

The **uusetup.sh** will lead you through the setup process automatically by asking questions. Before performing Step 3 in the procedure to activate the UUCP utility, if you are not familiar with executing the **uusetup.sh** interactive script, see section "Example of the UUCP Configuration Setup" for a typical setup scenario.

Activating the UUCP Utility Procedure

Using the ROOT login, the UUCP system administrator can activate the UUCP utility by executing the following steps:

1. Execute the UUCP Link Growth procedure to grow the *UNIX* System Maintenance Terminal (TTY) to be used by the UUCP utility. (Refer to 254-303-100, *Growth/Retrofit Tasks*, for the UUCP Link Growth procedure.) This UUCP TTY will usually be referred to as “ttyU” in the remainder of this chapter. Please replace “ttyU” with the actual tty ID when executing the following steps; for example, TTY 24 is ttym.
2. Activate the UUCP utility on the remote system linked to the local 3B21D computer system by following activation procedures for the remote system’s UUCP package.
3. Use the following 3B21D computer UUCP setup script to set up the UUCP login and configure the files:

```
# /usr/lib/uucp/uusetup.sh
```

- a. Enter all requested information.
- b. When finished entering all the remote systems, enter an empty remote system name. (Press the **Return** key when prompted to “Enter a new REMOTE system name.”)

The script finishes by automatically running the **uucheck** command.



NOTE:

See section “Example of the UUCP Configuration Setup” for more information.

Deactivating the UUCP Utility Procedure

Temporary Deactivation

Using the ROOT login, the UUCP system administrator can deactivate the UUCP utility by removing the *Permissions* file, */etc/uucp/Permissions*. If this deactivation is only temporary, the system administrator should rename the file as follows:

```
mv /etc/uucp/Permissions /etc/uucp/permissions
```

Assuming this has been done, the UUCP utility can be quickly reactivated by entering:

```
mv /etc/uucp/permissions /etc/uucp/Permissions
```

Permanent Deactivation

If the system administrator wishes to permanently deactivate the UUCP package, the *Permissions* file should be removed, as previously indicated, AND, all login entries in the password file, */etc/passwd*, related to uucp should be removed INCLUDING the uucp login.

Additionally, the UUCP Link Degrowth procedure should be followed to remove the TTY link hardware associated with UUCP. For information on the UUCP link degrowth procedure, refer to 254-303-100, *Growth/Retrofit Tasks*.

3B21D Computer UUCP Security Considerations

Like all systems with UUCP installed, the security of the 3B21D computer system depends on restricting the accessibility to the following:

- Local system
- Directories and files within the local system.

This section discusses guidelines for enhancing the security of the local system, in general, and recommendations for the use of the 3B21D computer UUCP package. If these guidelines and recommendations are NOT followed, the security of the local system will be less than desired.

The 3B21D computer UUCP package defines several security levels to control accessibility checks. See "Config File" in the "UUCP Administration" section. The recommended security level implements the guidelines as indicated in the

following sections. If the user sets the security level below the recommended level, the security of the local system will also be less than desired.

Local System Accessibility

Local system access is controlled through the `/etc/passwd` password file. Remote system access of the local system is enabled by the `Systems` file resident on the remote system and is usually named `/etc/uucp/Systems`. The `Systems` file contains both the login names and the unencrypted passwords for the local system.



CAUTION:

It is IMPERATIVE that NEITHER of these files be accessible by unauthorized logins. Otherwise, all security based on control of accessibility is LOST!

A well known way for access control to be compromised is by having an unauthorized user experientially determine a *login/password* combination. To reduce the probability of this happening, the following guidelines are implemented within the 3B21D computer UUCP package and checked by **uucheck**:

- No login, except the UUCP administrator's login, should contain the string "uucp".
- The login for the remote system must not contain the remote system's name nor should the remote system's name contain the login name.
- All access to the local system by the remote system should be validated.



NOTE:

This requires every LOGNAME entry in the *Permissions* file to have a VALIDATE entry. For more details, see section "Permissions File".

- Each remote system should have a separate login.



NOTE:

This requires that every VALIDATE entry have only one remote system name and that each LOGNAME entry be unique. For more details, see section "Permissions File".

However, even if all these guidelines are followed, it is still easy for a determined hacker to spoof their way into the local system once they have determined a login/password combination. As an extra level of protection, the 3B21D computer UUCP package supports conversation sequence numbers. It is strongly recommended that this capability be used as indicated in the following section.

⇒ **NOTE:**

Using conversational sequence numbers with a remote system running UUCP Version 2 or Basic Networking Utilities (BNU) is NOT recommended since this will require making the configuration directory on the remote system writable by UUCP. If you want to use conversational sequence numbers, you will need to upgrade your UUCP package on the remote system to a package (for example, TAYLOR UUCP) that supports conversational sequence numbers while maintaining the security of the configuration files.

Conversation Sequence Numbers

When using conversational sequence numbers, both the remote and local systems maintain counters indicating how many conversations they have had between each other. At connection time, both sides check the other's counter. If any mismatch is found, access is denied. This, in turn, will alert the UUCP system administrators that a spoof attempt has succeeded at which time *Systems*, *Permissions*, and */etc/passwd* entries should be changed.

For the 3B21D computer UUCP package, the conversational sequence numbers are kept in */var/uucp/.Admin/SQFILE*. This file contains one entry for each remote system. Each entry consists of the remote system's name followed by a decimal number whose value is between 0 and 9998. A time stamp is also provided indicating the time of the last conversation.

⇒ **NOTE:**

This value must be the same as the local system's entry in the SQFILE on the remote system.

Local Directory and File Accessibility

The local system administrator and the local UUCP administrator, using the *UNIX* system file protection mechanism and the *Permissions* file facilities, control what directories UUCP can read from and write to and what files can be accessed. As delivered, the 3B21D computer UUCP package is structured such that access is restricted to a specific "pubdir" */var/spool/uucppublic*. **uuccheck** will verify that this security is not compromised by entries in the *Permissions* and/or */etc/passwd* files.

⇒ **NOTE:**

It is important to never change permissions or ownership of any UUCP product files, configuration files, or directories.

When specifying *Permissions* file entries, the following guidelines should be strictly followed:

- Never use any login directory in either a READ or WRITE rule.
- Never use any uucp directory other than the uucp *spool* directory, */var/spool*, in either a READ or WRITE rule.
- Make sure uucp has read/write access for any PUBDIR rules. Generally, PUBDIR should have global read/write access.
- If you specify a READ or WRITE rule, make sure the PUBDIR directory is included in the path.
- Make sure all NOREAD rules are proper subdirectories of some READ rule or PUBDIR in the same entry.
- Make sure all NOWRITE rules are proper subdirectories of some WRITE rule or PUBDIR rule in the same entry.

Another major consideration affecting local directory and file accessibility depends on the commands that can be executed on the local system from the remote system. Accessibility of all files specified on the command line are verified by the UUCP package. However, any file access internal to the command is beyond the scope of the UUCP package. Internal access is controlled by the *UNIX* system file protection mechanism and will limit access based on the login ID of the remote user.

In general, it is strongly recommended that the list of commands that can be run by a remote system user be severely limited.

By default, the remote system is not allowed to execute any commands.

File System Integrity

Another major concern for UUCP users is file system integrity. This is especially crucial for 3B21D computer users. A well known problem is running a file system out of space. To minimize the probability of running out of space, the following guidelines should be followed:

1. Be sure to run the UUCP cleanup script, */usr/lib/uucp/uuclean.sh*, to clean up the UUCP directories. Analyze all output and mail messages generated by **uuclean.sh** to detect file system space problems as early as possible.
2. Never put a directory within a critical file system, such as */* (root), in a WRITE or PUBDIR rule in the *Permissions* file.

The 3B21D computer UUCP package has several design constraints to minimize impact on crucial file systems.

1. All UUCP system log files, work files, and debug files are kept in */var*.
2. For any 3B21D computer UUCP command, redirection of any standard error output, including but not limited to debugging output, will always be placed in */var/uucp/.Admin/prog_name.dbg*.
3. All mail messages will be placed in the user's mail box in */var/mail*. If it does not exist, it will be created.
4. The system administrator can limit the size of a user's mailbox by creating an entry in */var/mail/MAXSIZE*. For details, see the **mail()** command description in 235-700-200, *UNIX RTR Operating System Reference Manual*.
5. The system administrator can specify that mail should leave a minimum of free blocks in */tmp* and */var* using special entries (*#MINMAILTMP* and *#MINMAILDIR*, respectively) in */var/mail/MAXSIZE*. For details, see the **mail()** command description in 235-700-200, *UNIX RTR Operating System Reference Manual*.

As a result of the previously mentioned design changes and by following these guidelines and those in the previous section, the use of the UUCP package should be limited to noncrucial file systems specified in the WRITE and PUBDIR rules and */var* itself.

Operation of UUCP

This section introduces hardware and software prior to describing how UUCP operates. Before the local system can communicate with a remote system, a two-way communication link must be established between the two computers. This involves the following:

- Physical hardware connection
- UUCP software.

Hardware Connections

The UUCP feature is supported only on the UN582 MPC. The UN582 MPC supports an RS-232 connection. The RS-232 connection setting is 8 data bit, 1 start bit, 1 stop bit, and no parity.

Before the 3B21D computer can communicate with a remote system, a communication link must be established to the remote system. There are two types of hardware supported to establish a communication link to another system as follows: direct connection and *Datakit* VCS connection.

Direct Connection

The direct connection can be a hardwired direct link using a null modem RS-232 cable connected from a port on the UN582 board to a port on the remote system.

The 3B21D computer number for the cable used in this connection is ED3T076-20 G57 or G58. The G57 cable has a female connector to attach to the remote system. The G58 cable has a male connector to attach to the remote system. The maximum G57 or G58 cable length is 50 feet. The connections for this cable are shown in Table 23-1.

Table 23-1. ED3T076-20 G57 and G58 Signal Connections

3B21D Computer	Remote Machine
TxD	RxD
RxD	TxD
RTS looped back to CTS	
	RTS looped back to CTS
SG	SG
DSR & DCD	DTR
DTR	DSR & DCD

Datakit® VCS Connection

The second type of communication link is established through a *Datakit* VCS. In this case, the 3B21D computer must be a node on a *Datakit* VCS. This will allow the 3B21D computer to be connected from any system to establish a link to any system connected to the private network via the *Datakit* VCS.

If ordering x-dimension RS-232 cable from the *Datakit* VCS side, the cable number is ED5D500-21, G39G. The RTR number for the cable is ED3T076-20, G52A. The maximum cable length is 50 feet.

⇒ NOTE:

During *Datakit* VCS setup, the parity must be set to NULL.

UUCP Software

The UUCP utilities are composed of software programs, daemon processes, and supporting configuration files. The supporting configuration files contain information such as security restrictions, etc.

Directories and Their Purpose

There are several directories that contain the programs and support files of UUCP.

The directories used by the UUCP utilities are listed as follows:

<i>/usr/bin</i>	This directory is used to store executable programs and is used by the UUCP utilities for the same purpose. The binary programs uulog , uucp , uname , uustat , and uux are located here.
<i>/usr/lib/uucp</i>	The programs contained in the <i>/usr/lib/uucp</i> directory are uucico , uuclean.sh , uucleanup , uuccheck , uuxqt , uusetup.sh , uucpevol.sh , and remote.unknown .
<i>/etc/uucp</i>	The configuration and control files contained in the <i>/etc/uucp</i> directory are <i>Dialers</i> , <i>Dialcodes</i> , <i>Systems</i> , <i>Devices</i> , <i>Maxuuxqts</i> , <i>Permissions</i> , and <i>Config</i> .
<i>/var/spool/locks</i>	This directory contains the lock (<i>LCK</i>) files for the UUCP hardware devices. Lock files are discussed later in this chapter.
<i>/var/spool/uucp</i>	This directory is the <i>spool</i> directory for "work" that is to be processed by the UUCP utilities. It contains a tree-like structure of subdirectories associated with remote systems that the 3B21D computer communicates with or has communicated with recently.
<i>/var/spool/uucppublic</i>	This directory is the default <i>public</i> directory for UUCP transfers. The <i>public</i> directory is used to store files that have been sent to the 3B21D computer. The UUCP administrator can overwrite the default directory by using the <i>PUBDIR</i> entry in the <i>Permissions</i> file.
<i>/var/mail</i>	This directory contains mail generated by the UUCP utilities and others.

The following subdirectories located in */var/uucp* are used for administrative purposes such as storing log and status information.

<i>/var/uucp/.Admin</i>	This directory contains administrative, debug, and sequence files.
<i>/var/uucp/.Corrupt</i>	This directory contains corrupt work files and reexecute files that could not be processed.
<i>/var/uucp/.Login</i>	This directory is the login directory for the uucp login.
<i>/var/uucp/.Log</i>	This directory contains subdirectories which store the log files for different programs.
<i>/var/uucp/.Log/uucico</i>	This subdirectory contains uucico execution logs.
<i>/var/uucp/.Log/uucp</i>	This subdirectory contains uucp execution logs.
<i>/var/uucp/.Log/uux</i>	This subdirectory contains uux execution logs.
<i>/var/uucp/.Log/uuxqt</i>	This directory contains uuxqt execution logs.
<i>/var/uucp/.Old</i>	This directory contains compressed log files and administrative files.
<i>/var/uucp/.Sequence</i>	This directory contains job sequence numbers for each system.
<i>/var/uucp/.Status</i>	This directory contains status files for each system.
<i>/var/uucp/.Workspace</i>	This directory contains UUCP temporary workspace area.
<i>/var/uucp/.Xqtdir</i>	This directory contains workspace for remote execution.

Software Programs and Their Purpose

There are several types of programs associated with the UUCP utilities. Some of these programs are used by users to transfer data and obtain status information, while others are used for administrative purposes. The following paragraphs contain a brief description of the programs and their purpose. Descriptions for each of these commands are provided in 235-700-200, *UNIX RTR Operating System Reference Manual*.

Supported User Programs

uucp Queues a transfer job for the **uucico** daemon.

uname Lists the name of all the systems known to UUCP.

uustat Displays status information for requested transfers (**uucp** and **uux**). It also provides a means of controlling queued transfers.

uux Queues a remote execution job for the **uuxqt** daemon.

UUCP Daemons and Their Purpose

There are two daemons that are part of the UUCP utilities. These daemons are routines that run as background processes to handle file transfers and command executions. These daemons will not be run by a user under ordinary circumstances.

uucico This daemon is a background routine that is referred to as the transport program for UUCP requests. On the remote systems, **uucico** selects the device used for the link, establishes the link to the 3B21D computer, performs the required login sequence, performs permission checks, transfers data and "execute" files, logs results, and notifies specified users of transfer completions via **mail**. The remote system can be denied permission to transfer these files by the **uucico** program that runs on the 3B21D computer. The **uucico** daemon on the local system is started when the remote system connects to a 3B21D computer. The **uucico** daemon runs under a nonroot user ID of "uucp", UID=5.

uuxqt This daemon is the execution program for remote execution requests and is started just before **uucico** terminates. **uuxqt** searches the *spool* directory for "execute" files (X.) that have been sent from a remote system. When an X. file is found, **uuxqt** opens it to get the list of data files that are required for the execution. It then checks to see if the required data files are available and accessible. If the files are present and can be accessed, **uuxqt** checks the *Permissions* file to verify that it has permission to run the requested command.

⇒ NOTE:

The remote system must be connected to the local system to transfer results from the previous execution of the **uux**. The **uuxqt** daemon runs under a nonroot user ID of "uucp", UID=5.

How UUCP Operates in the Passive Machine Configuration

The **uucp** command performs all of the preliminary work in queuing file transfers to remote systems. It creates work files which contain the instructions for transferring the queued file(s). Depending on the options specified, it may make a copy of the file to be transferred in the *spool* directory. These files are called data files.

Operation of UUCP on a Local Machine

If the **uucp** command is executed on the 3B21D computer, the work and data files are created on the 3B21D computer. The remote systems must connect to a 3B21D computer to initiate file transfer between the remote and the 3B21D computer. When the remote system connects to the 3B21D computer, the **uucico** daemon process is started on the 3B21D computer during the *login* process. If there are any queued files on the 3B21D computer, the **uucico** on the 3B21D computer transfers them to a remote system.

Operation of UUCP on a Remote Machine

If the **uucp** command is executed from the remote *UNIX* system platform, **uucp** calls the **uucico** daemon which in turn attempts to contact the 3B21D computer and if successful, **uucico** on the remote system starts file transfer to the 3B21D computer. The **uucico** daemon on the 3B21D computer is started when the remote system logs in to the 3B21D computer.

Operation of uux on a Local Machine

If the **uux** command is executed on the 3B21D computer, it creates work, data, and execute files for executing commands on a remote system. The execute files contain the command string to be run on the remote system and a list of the data files. The data files are those files required for the command execution.

When the remote system connects to the 3B21D computer, the **uucico** daemon is started on the 3B21D computer. The work, data, and execute files are transferred to the remote system when the remote system connects to the 3B21D computer. The command is executed by **uuxqt** running on the remote system.

Operation of uux on a Remote Machine

If the **uux** command is executed on the remote system, it creates work, data, and execute files for executing commands on the 3B21D computer. The work file contains the same information as those created by **uucp**. The execute files contain the command string to be run on the 3B21D computer and a list of the data files. The data files are those files required for the command execution.

After the files are created, the **uucico** daemon is started on the remote system. The **uucico** on the remote system establishes the connection to the 3B21D computer and starts the **uucico** daemon on the 3B21D computer. Once the **uucico** is started on the 3B21D computer, after initial handshaking, files are

transferred between the two computers. After file transfer is completed, the connection is dropped. The **uucico** daemon on the 3B21D computer starts **uuxqt** to execute command.

⇒ **NOTE:**

Since the connection is dropped, the remote system must contact the 3B21D computer a second time to transfer results from the previous **uux** command execution. This second connection must happen after the execution of **uuxqt** on the local system has completed. The completion of **uuxqt** is undeterminate and depends on the specific commands being executed.

Administrative Programs

uucheck This command checks for the presence of UUCP directories, programs, and support files. It also checks certain parts of the *Permissions* and *Config* files for syntactic errors and checks consistency between the *Permissions* file and */etc/passwd* file.

⇒ **NOTE:**

The **uucheck** program should be run any time the *Config* or *Permissions* file is changed.

uucleanup The **uucleanup** program has several functions that are all associated with the cleanup of the *spool* directory. It is normally run by a shell script called **uuclean.sh** which is usually started by **cron**.

The **uuclean.sh** shell script is used to delete selected files from the UUCP related directories as follows:

<i>/var/spool/uucp</i>	Default cleanup depends on file type as follows: — Default cleanup for C. files after 7 days — Default cleanup for D. files after 7 days — Default cleanup for X. files after 7 days — Default cleanup after 2 days for other files in <i>uucp</i> directory.
<i>/var/mail</i>	Default cleanup removes the uucp login messages after 14 days.
<i>/var/spool/uucppublic</i>	Default cleanup after 30 days.
<i>/var/uucp/.Admin</i>	Compressed into log files in <i>/var/uucp/.Old</i> each time uuclean.sh is run.

<i>/var/uucp/.Corrupt</i>	Default cleanup after 10 days.
<i>/var/uucp/.Log</i>	Compressed into log files in <i>/var/uucp/.Old</i> each time uuclean.sh is run.
<i>/var/uucp/.Old</i>	Default backup after 14 days.
<i>/var/uucp/.Sequence</i>	Default cleanup after 30 days.
<i>/var/uucp/.Status</i>	Compressed into log files in <i>/var/uucp/.Old</i> each time uuclean.sh is run.
<i>/var/uucp/.Workspace</i>	Default cleanup after 1 day.
<i>/var/uucp/.Xqtdir</i>	Default cleanup after 1 day.

uulog This program displays the contents of a specified remote system's log file. Individual log files are created for each remote system the 3B21D computer communicates with using the **uucp** and **uux** programs.

Supporting Database Files and Their Purpose

As mentioned earlier, several of the UUCP programs require information contained in configuration and control support files. These support files are located in the */etc/uucp* directory.

⇒ NOTE:

To keep the 3B21D computer UUCP implementation in sync with other *UNIX* system UUCP implementation, the following configuration files are delivered to the customer. Since the 3B21D computer is always a passive system for the current UUCP implementation, the UUCP administrator needs to configure only *Config*, *Permissions*, and *Systems* files.

The **uucico** and **uuxqt** programs require supporting information from the following files:

<i>Config</i>	This file contains the following fields that are described in the "Config File" section. ▪ MAILFLAG ▪ SPOOLMOUNT ▪ SECURITY LEVEL ▪ NODENAME.
<i>Devices</i>	This file contains all systems known to the local RTR system, direct links plus direct connect device information, and <i>Datakit</i> VCS network devices. This dummy file is not used in the current implementation.

<i>Dialers</i>	This file contains a dummy entry for direct connect or character strings required to negotiate with <i>Datakit</i> VCS network devices in the establishment of connections to remote computers. This dummy file is not used in the current implementation.
<i>Dialcodes</i>	This file contains the dial-code abbreviations used in the Phone field of the <i>Systems</i> file. This dummy file is not used in the current implementation.
<i>Maxuuxqts</i>	This file contains the maximum number of uuxqt processes that can be running concurrently. The recommended limit of concurrent uuxqt processes is 1 in order not to use up excessive Central Processing Unit (CPU) cycles. The user should not modify this file.
<i>Permissions</i>	This file defines the level of access that is granted to systems when they attempt to transfer files or remotely run commands on the 3B21D computer.
<i>Systems</i>	The <i>Systems</i> file contains the information needed by the uucico daemon to establish a connection link from the remote system.

UUCP Administration

This section discusses the files and tasks associated with the operation of the UUCP utilities. The amount of effort required to administer the UUCP utilities depends on the number of nodes connected and amount of traffic that enters or leaves the 3B21D computer.

Introduction to the UUCP Configuration Files

A major part of UUCP setup is the configuration of the files. These files are located in the */etc/uucp* directory and must be edited to set up UUCP functionality.

⇒ NOTE:

The UUCP implementation on the 3B21D computer does not support callout capability. The 3B21D computer is always a passive system. Because of this restriction, the UUCP administrator needs to modify only the following files:

- *Config* - Contains a list of variable parameters. The UUCP administrator can manually set up the parameters to configure the UUCP utility.
- *Systems* - Contains the information needed by the **uucico** daemon to establish a communication link from a remote system.

- *Permissions* - Defines the level of access granted to remote hosts that attempt to transfer files or execute commands on the local computer.

The following conventions are used and must be followed for creating/editing all configuration files:

- A line beginning with a hash mark (#) is a comment.
- The field separator is a space or tab.
- Unused fields in the *Systems* files must contain a hyphen.
- Long lines can be continued with a backslash at the end of the line, excluding *Config* file lines.

Config File

This is the RTR Configuration Control file (*/etc/uucp/Config*). This file can be used to set RTR specific configurations for the RTR version of the UUCP package. Each field must be defined on a new line. Blank spaces are not allowed between field name and field value.

The following configuration control parameters are available:

- **SPOOLMOUNT={TRUE|FALSE}** — indicates if the spooler directory is in a mountable disk partition (TRUE) or not (FALSE).
If **SPOOLMOUNT=TRUE**, checks will be made to verify the partition is properly mounted before executing any UUCP process. Default = TRUE.
- **SECURITYLEVEL=n** — sets the security level for the UUCP program to “n”. The purpose of **SECURITYLEVEL** and different levels are explained in the section “RTR uucp Security Considerations”. Default security level is 7.
- **MAILFLAG={TRUE|FALSE}** — enables (TRUE) or disables (FALSE) the use of **rmail** by the UUCP programs to inform the end user of errors and/or status. Default = FALSE.
- **DEBUGpgm=n** — sets the debug level for UUCP program “pgm” to n.
pgm can be **uucp** | **uucico** | **uux** | **uuxqt** | **uuname** | **uustat** | **uucleanup** | **uuccheck**
Debug information will either go to the user's terminal or to a debug file in */var/spool/.Admin/pgm.dbg*.
- **NODENAME=system-name** — overrides the system-name in the RCV DB EAIPT record “eaiop0”. **NODENAME** must be used if the EAIPT record contains characters that are invalid in forming UUCP temp file names. There is no default. UUCP will attempt to use the EAIPT record unless the **NODENAME** keyword is populated.

- **MAXDEBUGLEVEL=n** — prevents uucp programs from setting their debug levels greater than “n”. This is useful in preventing remote system users from using **uucico** to generate debugging info on the local system and possibly filling up */var*. If debugging is required, the UUCP administrator should modify **MAXDEBUGLEVEL** in this file for the duration of the debugging and then reset it as soon as the debugging is completed to the default of 0. Default = 0.

Systems File

The *Systems* file (*/etc/uucp/Systems*) contains the information needed by the **uucico** daemon to establish a communication link to a remote system.

Each entry in the *Systems* file has the following format:

System-name Time Type Class/Speed Phone Chat-Script

where each field is defined in the following paragraphs:

System-name	This field contains the node name of the remote system.
Time	Since the 3B21D computer is always a passive system, this field is not used. By default, it must be populated “Never”.
Type	Since the 3B21D computer is always a passive system, this field is not used. By default, it must be populated “-”.
Class/Speed	Since the 3B21D computer is always a passive system, this field is not used. By default, it must be populated “-”.
Phone	Since the 3B21D computer is always a passive system, this field is not used. By default, it must be populated “-”.
Chat-Script	Since the 3B21D computer is always a passive system, this field is not used. By default, it should be populated “-”.

Permissions File

The *Permissions* file (*/etc/uucp/Permissions*) is used to specify the permissions that remote systems have with respect to login, file access, and command execution. Rules are provided for restricting the ability to request files and the ability to receive files queued by the local site. In addition, a rule is available to specify the commands that a remote site can run on the local system.

Each entry is a logical line, with physical lines terminated by a backslash to indicate continuation. Entries are made up of rules delimited by a blank space. Each rule is a name-value pair in the following format:

name=value

Some values can be a colon-separated list. No blank space is allowed within an rule assignment. Comment line begins with (#) as the first character and occupy the entire line up to a newline character.

There are two types of *Permissions* file entries:

- LOGNAME** Specifies permissions and file transfer path that take effect when a remote system logs in on a 3B21D computer to send a file.
- MACHINE** Specifies permissions that take effect when a remote system logs into a 3B21D computer to execute commands.

The LOGNAME and MACHINE entries should be combined into a single entry.

REQUEST Keyword Rule

When a remote system calls the 3B21D computer and requests to receive a file, this request can be granted or denied. The REQUEST rule specifies whether the remote system can request to set up file transfers from the 3B21D computer. The string

REQUEST=yes

specifies that the remote system can request to transfer files from the 3B21D computer. The string

REQUEST=no

specifies that the remote system cannot request to receive files from the 3B21D computer. The "no" string is the default value. It will be used if the REQUEST rule is not specified.

SENDFILES Keyword Rule

When a remote system calls the 3B21D computer and completes its work, it may attempt to take work the 3B21D computer has queued for it. The SENDFILES rule specifies whether a 3B21D computer can send the work queued for the remote system. The string

SENDFILES=yes

specifies that the 3B21D computer may send the work that is queued for the remote system as long as it logged in as one of the names in the LOGNAME entry.

⇒ NOTE:

Setting the SENDFILES rule to anything other than "yes" will result in not sending files from the 3B21D computer to the remote system.

READ and WRITE Keyword Rules

The READ and WRITE entries specify the various parts of the file system that **uucico** can read from or write to. The READ and WRITE rules can be used with either MACHINE and/or LOGNAME actions.

The default for both the READ and WRITE rules is the *uucppublic* directory as shown in the following strings:

```
READ=/var/spool/uucppublic WRITE=/var/spool/uucppublic
```

The strings

```
READ=/ WRITE=
```

specify permission to access any file that can be accessed by a local user with "other" permissions.

The value of these rules is a colon-separated list of pathnames. The READ rule is for requesting files, and the WRITE rule is for depositing files. To grant permission to deposit files in */usr/news* as well as the *public* directory, the following values would be used with the WRITE rule:

```
WRITE=/var/spool/uucppublic:/usr/xyz
```

It should be pointed out that if the READ and WRITE rules are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the */usr/xyz* pathname was the only one specified in a WRITE rule, permission to deposit files in the *public* directory would be denied.

If PUBDIR entry is defined, then READ/WRITE will refer to PUBDIR and not to the *uucppublic* directory.

NOREAD and NOWRITE Keyword Rules

The NOREAD and NOWRITE rules specify exceptions to the READ and WRITE entries or defaults. The strings

```
READ=/ NOREAD=/etc WRITE=/var/spool/uucppublic
```

would permit reading any file except those in the */etc* directory and its subdirectories and writing only to the default */var/spool/uucppublic* directory. NOWRITE works in the same manner as the NOREAD rule. The NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

NOREAD/NOWRITE always take precedence over READ/WRITE, respectively.

CALLBACK Keyword Rule

The CALLBACK rule is ***not supported*** and will cause the **uucp** program to abort if specified on the 3B21D computer.

COMMANDS Keyword Rule



CAUTION:

The COMMANDS rule overrides the default commands list that the remote system can run locally.

The **uux** program will generate remote execution requests and queue them to be transferred to the remote system. The COMMANDS entry can be used to override the default command list and specify the commands that a remote system can run on the 3B21D computer. The string

`COMMANDS=rmail`

indicates that a remote system can execute **rmail** on the 3B21D computer. The default search path is as follows:

`/bin:/usr/bin`

In addition to the names, there can be full pathnames of commands, which is strongly recommended. For example,

`COMMANDS=rmail:/usr/bin/xyz`

specifies that command **rmail** uses the default path. When the remote system specifies **xyz** or **/usr/bin/xyz** for the command to be run, **/usr/bin/xyz** will be run regardless of the default path.

Including the "ALL" value in the list means that any command from the remote system specified in the entry will be run. If a user uses this value, it gives the remote system full access to the 3B21D computer.

The string

`COMMANDS=/usr/bin/xyz:ALL`

illustrates two points. The pathnames specified for **xyz** will be used, instead of the default if the requested command does not contain the full path names for **xyz**. Also, the ALL value can appear anywhere in the string and it will allow the user to execute any command.



CAUTION:

The use of ALL in the COMMANDS rule is not recommended because it degrades security.

The VALIDATE rule should be used with the COMMANDS rule whenever potentially dangerous commands like **cat** and **uucp** are specified with the COMMANDS rule.

VALIDATE Keyword Rule

The VALIDATE rule is used in conjunction with LOGNAME and MACHINE rules to restrict access. It is used to provide a certain degree of verification of the caller's identity. The use of the VALIDATE rule requires that privileged remote systems have a unique login/password for UUCP transactions. An important aspect of this validation is that the login/password associated with this rule be protected. The LOGNAME rule

```
LOGNAME=friend VALIDATE=eagle
```

specifies that if one of the remote systems that claims to be "eagle" logs in on the 3B21D computer, it must have used the login ID "friend".

MYNAME Keyword Rule

The MYNAME rule enables the user to designate a unique UUCP node name for the local computer, in addition to the name returned by **uname -l** command. This keyword will override both the system-name specified in the *eaiopt* record in the ECD and the NODENAME parameter specified in the *Config* file, */etc/uucp/Config*.

PUBDIR Keyword Rule

The PUBDIR rule specifies the public directory for local access. The PUBDIR rule can be used to overwrite the default *uucp* public directory (*spoolMNTPT/spool/uucpublic*); for example,

```
PUBDIR=/var/spool/uucppublic/dirA
```

Sample Permissions Files

Example 1. This first example represents the most restrictive access to the computer.

```
LOGNAME=user1 VALIDATE=node1 MACHINE=node1
```

It states that login "user1" has all the default permissions/restrictions.

- The remote system can only send files to **uucppublic**.
- The remote system cannot request to receive files (REQUEST rule).
- If any other system than "node1" attempts to log in as "user1", connection is refused.

- No files that are queued for the remote system will be transferred during the current session (SENDFILES rule).
- No command can be run by defaults.

This rule alone is sufficient to start communications with remote systems, permitting files to be transferred only to the */var/spool/uucppublic* directory.

Example 2. The next example is for remote systems that log in but have fewer restrictions.

```
LOGNAME=user2 REQUEST=yes SENDFILES=yes \  
READ=/ WRITE=/ VALIDATE=node2 MACHINE=node2
```

This rule places the following permissions/restrictions on a system that logs in as “user2.”

- Files can be requested from the 3B21D computer (REQUEST rule).
- If any other system than “node2” attempts to log in as “user2,” the connection is refused.
- Files can be transferred to any directory or any file that is writable by the user “other.”
- Any files readable by the user “other” can be requested (READ rule).
- Any requests queued for the remote system will be run during the current session. These are files destined for the remote system that has called in (SENDFILES rule).
- No command sent for execution on the 3B21D computer will succeed.

***Maxuuxqts* File**

The *Maxuuxqts* (*/etc/uucp/Maxuuxqts*) text file contains a number to limit the number of simultaneous **uuxqt** programs running. This file is delivered with a default rule of one. Values other than one are ***not supported*** in the 3B21D computer implementation.

remote.unknown

The **remote.unknown** program (*/etc/uucp/remote.unknown*) is a shell file that is run when a remote site that is not in the *Systems* file calls in to start a conversation. The shell script, by default, will append the name and time information to the */var/uucp/.Admin/Foreign* file.

Administrative Tasks

To support UUCP, a minimum amount of maintenance is required to ensure that the network runs properly. When more than one remote system is involved, the job becomes more difficult because there are more entries to update. The following sections describe the routine administrative tasks that must be performed by the UUCP administrator or are automatically performed by the UUCP **uuclean.sh** shell script.

One of the biggest problems in a UUCP network is dealing with the backlog of jobs that cannot be transmitted to other systems. The following cleanup activities should be routinely performed.

Cleanup of Undeliverable Jobs

The **uustat** program provides a user with information about the latest attempts to contact various systems and the age and number of jobs in the queue for remote systems. The **uustat** program should be invoked regularly to provide information about the status of connections to various systems and the size and age of the queued requests. Of particular interest is the age of the oldest request in each queue.

The **uuclean.sh** shell script should be invoked by administrator to remove any jobs that have been queued for several days and cannot be sent. By default, the leftover data (*D.*) and work (*C.*) files are removed after 7 days and run (*X.*) files are removed after 7 days. The **uuclean.sh** also provides feedback to the user indicating when jobs are not being accomplished and when these jobs are being deleted.

Cleanup of the Public Area

In order to keep the local file system from overflowing when files are sent to the public area (*/var/spool/uucppublic*), the **uuclean.sh** shell script is set up to remove any files that are older than 7 days and directories that are empty.

Machine Log Files

Log files are created for each remote system with which the 3B21D computer communicates. Each system may have four log files, one each for **uucico**, **uuxqt**, **uux**, and **uucp** requests, depending on the type of communication that has taken place. The log files are kept in the */var/uucp/Log* directory. By default, the files are kept 14 days before they are removed.

Compaction of Log Files

The UUCP commands create individual log files for each system and each program. The **uulog** program gives the user access to the information in these files by system name. These files are combined and stored in the */var/uucp/Old* directory whenever **uuclean.sh** is run. The *.Old* files are cleaned up after 14 days.

uuclean.sh and **cron**

The **cron** daemon is a tool that proves to be very useful in the administration of the *UNIX* systems.

The **uuclean.sh** shell script cleans up the UUCP log files and directories. By default, archived log files are updated so that no log information over 14 days old is kept. Log files for individual systems are taken from the */var/uucp/Log* directory, merged, and placed in the */var/uucp/Old* directory along with the older log information. Files and directories that are no longer needed in the *spool* directories are removed. Old mail messages for uucp logins are cleaned up after default 14 days. After cleanup is performed, the UUCP administrative login "uucp" is mailed a summary of the status information gathered during the current cleanup interval.

The **uuclean.sh** shell script should be run by an entry in the *crontab* file. It can be run daily, weekly, or whenever, depending on the amount of UUCP traffic that enters and leaves the 3B21D computer.

The **cron** program executes commands at specified dates and times according to the instructions in the */usr/lib/crontab* file.

The following is a example of a *crontab* file entry for the **uuclean.sh** shell script.

```
15 4 * * * /bin/su uucp -c "/usr/lib/uucp/uuclean.sh"
```

NOTE:

Since other RTR processes are running at midnight, it is recommended to start **uuclean.sh** any time except at midnight.

Please review the **cron** command description in 235-700-200, *UNIX RTR Operating System Reference Manual*, for more information.

Setting Up UUCP Configurations

As described earlier, the UUCP customer is expected to use either a direct connection using an RS-232 cable or an asynchronous *Datakit* VCS connection. The examples for both type of connections are included in this section. A brief description of the */etc/passwd* file is included to simplify the understanding of the UUCP configuration. The UUCP system administrator must know the following setup information before setting up the UUCP configuration files:

- The port number/id must be known prior to setting up the UUCP configuration files.
- Speed of the device.
- *Datakit* VCS destination node name, if “datakit” is used.
- Both local and remote system names.

UUCP Logins and Passwords Administration

The */etc/passwd* file contains the following information for each user:

- Login name
- Encrypted password
- Numerical user ID
- Numerical group ID
- Optional free form information
- Initial working directory
- Program to use as shell.

Each field within each user's entry is separated from the next field by a colon. The uucp administrative login is the owner of all the UUCP object and spooled data files. The user ID and group ID for the uucp administrative login must be the number 5. The following is a sample entry in the */etc/passwd* file for the administrative login:

```
uucp:zAvLCKp:5:5:UUCP.Admin:/var/uucp/.Login:
```

The **xyz** access login allows remote systems to log in on the 3B21D computer. The following is a sample entry in the */etc/passwd* file for the access login:

```
xyz:zaaAA:285:285:uucp login:/var/spool:/usr/lib/uucp/uucico
```

Note that the standard shell is not given to the **xyz** login. The shell that **xyz** receives is the **uucico** daemon which controls the conversation when a remote system logs into a system.

The assignment of passwords for the **uucp** and **xyz** logins is left up to the UUCP administrator. The passwords should be at least six to eight characters. Only the first eight characters of the password are significant. It is recommended not to use password aging functionality.

⇒ NOTE:

If the password for the access login is changed for security reasons, make certain that the remote systems that are a part of the network are properly notified of the change.

Example of the UUCP Configuration Setup

The following is an example scenario for a typical UUCP configuration setup. The notes provide additional explanations while the screens show an example of running the **uusetup.sh** script.

Execute the **uusetup** program by entering:

```
# /usr/lib/uucp/uusetup.sh
```

The interactive process occurs as follows:

UUCP Configuration Setup

```
The uucp account has not been set up.  
Choose a password for the uucp account:  
New password:
```

```
Re-enter new password:
```

```
Your current setting does not allow UUCP jobs to send mail.  
Do you want to allow UUCP jobs to send mail? (y,n, default=n)  
y
```

```
Enter a new REMOTE system name, hit 'return' to finish  
dallas  
Adding entry  
dallas Never - - - -
```

⇒ NOTE:

The **uusetup.sh** script just asked for the name of a remote system that is to be allowed to perform UUCP transfers with the local system. In the example, “dallas” is the REMOTE system name entered. During realtime activation, when prompted for the REMOTE system name, enter a real REMOTE system name and not “dallas”.

What account will be used by system dallas to log in for file transfers? This account should not currently exist, and not contain the system name or the characters 'uucp' for security.

centralhq

Searching for an unused uid...

Trying uid=399

Adding the following line to /etc/passwd:

centralhq::399:399:UUCP login for

dallas:/var/spool:/usr/lib/uucp/uucico

Set the password for account centralhq. You will need to use the same password to configure the login sequence on system dallas.

New password:

Re-enter new password:

Writing Permissions file entry:

LOGNAME=centralhq VALIDATE=dallas MACHINE=dallas SENDFILES=yes
REQUEST=yes

This entry allows the remote system to request files from the 3B21D computer subject to the file access granted by the READ= and WRITE= keywords. To modify this, change the REQUEST= and SENDFILES= keywords. The default file access permissions allow authorized remote systems read and write access to the public directory /var/spool/uucppublic Change this by editing the READ= and WRITE= keywords in the Permissions By default the remote system is not allowed to execute any commands. Change this with the COMMANDS keyword. For more information consult the UUCP administrators guide.

⇒ NOTE:

During realtime activation, remember that each remote system added should have a different remote system name (for example, not "dallas") and a different login account name (for example, not "centralhq"). The login account name should not contain the word "uucp" or the name of the system with which it is associated. A password must be set for that login name.

Enter a new REMOTE system name, hit 'return' to finish

ft._worth

Adding entry

ft._worth Never - - -

What account will be used by system ft._worth to log in for file transfers? This account should not currently exist, and not contain the system name or the characters 'uucp' for security.

backuphq

Searching for an unused uid...

Trying uid=399

Trying uid=398

Adding the following line to /etc/passwd:

backuphq::398:398:UUCP login for

ft._worth:/var/spool:/usr/lib/uucp/uucico

Set the password for account backuphq. You will need to use the same password to configure the login sequence on system ft._worth.

New password:

Re-enter new password:

Writing Permissions file entry:

LOGNAME=backuphq VALIDATE=ft._worth MACHINE=ft._worth SENDFILES=yes
REQUEST=yes

This entry allows the remote system to request files from the 3B21D computer subject to the file access granted by the READ= and WRITE= keywords. To modify this, change the REQUEST= and SENDFILES= keywords. The default file access permissions allow authorized remote systems read and write access to the public directory /var/spool/uucppublic. Change this by editing the READ= and WRITE= keywords in the Permissions. By default the remote system is not allowed to execute any commands. Change this with the COMMANDS keyword. For more information consult the UUCP administrators guide.

⇒ NOTE:

The **uusetup.sh** script will recursively ask you for REMOTE system names to be set up. During realtime activation when finished adding REMOTE system names, enter a return without any text when prompted to enter a new REMOTE system name.

Enter a new REMOTE system name, hit 'return' to finish

CTRL-LF

You should make sure that there is a **cron** entry to run the cleanup task `uuclean.sh` periodically to clean up old jobs and log files. See the Administrative tasks section of the UUCP User's Guide or the file `/usr/lib/uucp/uuclean.sh`

Now running `uucheck`. Please correct any warnings or inconsistencies that are reported. Errors indicate serious problems that will prevent UUCP from operating properly. For a more detailed listing of your setup you may run `uucheck -v`. Also run `uucheck` after changing the UUCP configuration.

Begin `uucheck`

End `uucheck`

#

UUCP Setup for Direct Connection - An Example

To use direct connection, the RS-232 cable is connected from an asynchronous UN582 port to a serial port on a remote computer running the *UNIX* operating system.

The following sections describe, as an example, how to set up configurations for direct connection. A new entry must be added in each file for each connection.

Setup on Remote Machine Running Traditional UNIX® System

Systems File

The *Systems* file must have the following entry:

```
#System-name  Time    Type    Speed/Class  Phone  Chat-Script
labxyz        Any     Direct  Any          -      ogin: demo1 ssword: bbb120
```

In this example, the local *UNIX* system is set up to connect to a remote system *labxyz*, which will initiate a call to the 3B21D computer at any time with any speed. The *Type* field is the link to the entry in the *Devices* file. Once connected to a remote site, as explained in the *Chat-Script*, the local system expects login prompt, followed by a password prompt.

Devices File

The *Devices* file must have the following entry:

```
#Type      Line      Line2      Speed/Class      Dial-Token-Pairs
Direct     /dev/term/b  -          Any              direct
```

The *Type* field must match with the *Type* field in the *Systems* file. The *Line* field is the serial port through which the RS-232 cable is connected to the 3B21D computer. Since there is a direct link to the 3B21D computer, the Dial-Token-Pairs entry contains *direct*. The Dial-Token-Pairs is the link to the *Dialers* file.

Dialers File

The *Dialers* file must have the following entry:

#Dialer	Substitution	expect-send
direct		

Dialcodes File

Since the *phone* field in the *Systems* file is not populated, the *Dialcodes* file is not required any modifications.

Setup on the 3B21D Computer

Since the UUCP implementation on the 3B21D computer does not support *callout facility*, only the *Systems* and *Permissions* files need to be modified.

Systems File

This is the standard UUCP *Systems* file. Only the *sys-name* field (#1) is used in the current implementation. However, the whole file is parsed. As a result, all the other fields should be defaulted as follows:

#System-name	Time	Type	Speed/Class	Phone	Chat-Script
node1	Never	-	-	-	-

Permissions File

The following is the example of the *Permissions* file:

```
LOGNAME=bigcat MACHINE=node1 VALIDATE=node1 SENDFILES=yes \  
REQUEST=YES READ=/tmp WRITE=/tmp COMMANDS=rmail:ls:who
```

which states to allow “node1” to log into the 3B21D computer as “bigcat”, validates login, allows read and write to all directories, requests, or sends files, and executes the commands listed in the *COMMANDS* rule. In this example, “node1” is a *UNIX* system connected to the 3B21D computer.

Update */etc/passwd* File on the 3B21D Computer

Add an entry in the */etc/passwd* file on the 3B21D computer to allow access to the user “bigcat”.

```
bigcat::285:285:uucp login:/var/spool:/usr/lib/uucp/uucico
```

The system administrator must run the **uuccheck** command to cross check the *Permissions* file and the password file.

UUCP Setup for *Datakit* VCS Connection - An Example

Setup on the Remote Machine

The following sections describe how to set up UUCP configuration for *Datakit* VCS. A new entry must be added in each file for each connection.

Systems File

The *Systems* file for the remote site must have the following entry:

#System-name	Time	Type	Speed/Class	Phone	Chat-Script
3b21d-system	Any	DK	Any	il/ihe/rtr1	ogin: demo1 ssword: demo1name

In this example, the remote *UNIX* system is set up to connect to a local system *3b21d-system*, using *Datakit* VCS. The *Type* field is the link to the entry in the *Device* file. Once connected to a remote site, as explained in the *Chat-Script* field, the local system expects *login* prompt, followed by a *password* prompt.

Devices File

The *Devices* file must have the following entry:

#Type	Line	Line2	Speed/Class	Dial-Token-Pairs
DK	term/b	-	Any	DK

The *Type* field must match with the *Type* field in the *Systems* file. The *Line* field is the serial port through which the *Datakit* VCS cable is connected to the 3B21D computer. The *Dial-Token-Pairs* is the link to the *Dialers* file.

Dialers File for Datakit VCS

The *Dialers* file must have the following entry:

```
#Dialer Substitution expect-send
DK      ""      ""      TION:--TION: \D
```

where

"" First set equals no special character translation table
"" Second set equals expect nothing
- Delay to give some setting time
TION:--TION: Wait for the destination prompt. If it does not come, send a carriage return and wait again.
\D Send the specified token which is specified in the *Systems* file.

Dialcodes File

Since the *phone* field in the *Systems* file is not populated, the *Dialcodes* file is not required any modifications.

Setup on the 3B21D Computer

Since the UUCP implementation on the 3B21D computer does not support *callout facility*, only the *Systems* and *Permissions* files need to be modified.

Systems File

This is the standard UUCP *Systems* file. Only the *System-name* field (#1) is used. However, the whole file is parsed. As a result, all the other fields should be defaulted, as shown.

The *Systems* file must have the following entry:

```
#System-name  Time    Type    Speed/Class  Phone    Chat-Script
node1         Never   -       -            -        -
```

Permissions File

The following is the example of the *Permissions* file:

```
LOGNAME=demo1 MACHINE=node1 VALIDATE=node1 SENDFILES=yes \
REQUEST=YES READ=/tmp WRITE=/tmp COMMANDS=rmail:ls:who
```

which states to allow “node1” to log into the 3B21D computer as “demo1”, validates login, allows to read and write to all directories, requests or sends files, and executes the commands listed in the COMMANDS rule. In this example, “node1” is a *UNIX* system connected to a 3B21D computer. By using the **uname -l** command on the remote system, the user can find out the system name.

Update */etc/passwd* File on the 3B21D Computer

Run **passwd(2G)** command to add an entry in the */etc/passwd* file on the 3B21D computer to allow access to the user “fatcat”. For more information, see the **passwd(2G)** command description in 235-700-200, *UNIX RTR Operating System Reference Manual*.

```
demo1::5:5:uucp login:/var/spool:/usr/lib/uucp/uucico
```

The system administrator must run the **uucheck** command to cross check the *Permissions* file and the password file.

UUCP Security

A great deal of consideration should be given to provide a remote system with a privileged login and password for UUCP transactions. Giving a remote system a special login and password with file access and remote execution capability is like giving anyone on that system a normal login and password on the 3B21D computer.

The default */etc/uucp/permissions* file provides the maximum amount of security for UUCP links. The UUCP administrator can set additional parameters for each system to define.

- Ways it can receive files from a remote system
- Directories for which it has read and write permissions
- Commands it can use for remote execution.

Planning and Recommended Precautions

In setting up a network of *UNIX* RTR systems, there are several considerations that should be taken into account before configuring each system on the network. Some basic decisions about access to 3B21D computers and a remote system in the network must be made before attempting to set up the configuration files. The systems on the network must decide what level of access should be granted to other members of the network. The *UNIX* RTR system password mechanism is used to grant access to other systems.

3B21D Computer UUCP Security

The 3B21D computer UUCP package uses the standard *UNIX* system login password protection and *UNIX* system file and directory access to control access to the local system and to control access to files and directories within the local system as described in section "3B21D Computer UUCP Security Considerations".

The 3B21D computer UUCP package also uses a BNU styled *Permissions* file, */etc/uucp/Permissions*, with READ, NOREAD, WRITE, NOWRITE, and PUBDIR entries to further define access by UUCP users to files within the local system. These entries are discussed in sections "UUCP Administration" and "Permissions File".

The 3B21D computer UUCP package also implements a set of security guidelines, as discussed in section "3B21D Computer UUCP Security

Considerations". The standard BNU UUCP package security guidelines have been enhanced for the 5ESS switch environment as follows:

Conversational Sequence Numbers:

The standard BNU UUCP package keeps the conversational sequence numbers file, SQFILE, in the same directory as the UUCP configurations files. To access the SQFILE, its directory must be writable by UUCP. This is generally considered a security problem. Therefore, the 3B21D computer UUCP package has placed the SQFILE in */var/uucp/.Admin* in order to avoid this security problem.

It is recommended that conversational sequence numbers be used as long as the remote system UUCP package allows the use of conversational sequence numbers WITHOUT having to make the UUCP configuration file directory writable by UUCP. Taylor UUCP provides this functionality.

Security Level Control:

The UUCP administrator may control the level of the security checks by adding the SECURITYLEVEL=*n* keyword to the *Config* file, */etc/uucp/Config*. The security levels are as defined as follows:

- The recommended default security level is SECURITYLEVEL=7.
- The lowest level that can be set is SECURITYLEVEL=3.

If the SECURITYLEVEL is set to three (3) or above, the following must be true for a remote system connection to be successful:

- Each LOGNAME in the *Permissions* file must be unique (no duplicates).
- Each LOGNAME must have a VALIDATE keyword.
- Each VALIDATE must have only one remote system listed.

If SECURITYLEVEL is set to five (5) or above, the following must be true for a remote system connection to be successful:

- The LOGNAME must not contain the string "uucp".

If SECURITYLEVEL is set to seven (7) or above, the following must be true for a remote system connection to be successful:

- The LOGNAME must not contain the name of the remote system.
- The remote system must not contain the name of the LOGNAME.

Troubleshooting

The most common problems are errors in the */etc/uucp/Config* or */etc/uucp/Permissions* file. The first corrective action should be to run the **uuccheck** command to verify these files. If the problem still exists, see Table 23-2.

Table 23-2. UUCP Troubleshooting Corrective Action

Symptom	Corrective Action
I executed a uux job to the 3B21D computer, but I have not gotten any output.	1. From the remote system, execute /usr/lib/uucp/uucico -r1 -sname_of_3b -f 2. If still no output, verify that the 3B21D computer has SENDFILES=yes set in <i>/etc/uucp/Permissions</i> and reexecute Step 1. 3. If still no output, check mail to verify that the job did not fail. 4. If no jobs are queued on either the local or remote systems for each other, perhaps there was no output. If possible, try executing the uux job with the -z option to force mail on success cases.
I am trying to fetch a file from the 3B21D computer, but I do not get the file or any error messages.	1. Make sure that SENDFILES=yes is set in the <i>/etc/uucp/Permissions</i> file. 2. Check for any ASSERT of failing status messages related to the queued job. See "Error Code Interpretation" section and the remote system's UUCP documentation.
I have queued up a job on the 3B21D computer, but it never gets sent.	1. The 3B21D computer does not support callout; the connection must be established from the remote system by executing /usr/lib/uucp/uucico -r1 -sname_of_3b -f 2. Make sure the SENDFILES=yes is set in the <i>/etc/uucp/Permissions</i> file. 3. Check for any ASSERT of failing status messages related to the queued job. See "Error Code Interpretation" section and the remote system's UUCP documentation.

Table 23-2. UUCP Troubleshooting Corrective Action (Contd)

Symptom	Corrective Action
I used uulog -f and now I can't get my prompt back.	You are in a tail -f . Use the Delete key or ^C to escape.
I am trying to look at uux and uucp log files using the uulog command and I keep getting a usage statement.	Only one log file (<i>uucp</i> , <i>uux</i> , <i>uuxqt</i> , or <i>uucico</i>) can be viewed at a time via the uulog command.
When I do a uustat , I see jobs that look like: jon1N70ed 04/14-10:25:05 (POLL)	1. <i>/var</i> ran out of space. Manually run uuclean.sh to try to reclaim jobs. 2. Remove unneeded jobs.
I get "not found" when I try to execute uuto , uugetty , Uutry , uusched , uupick , uukick , cu , uuencode , or uudecode .	These commands are not currently supported.
I can't do any remote execution jobs.	By default, no commands are allowed by the UUCP system. Add the desired commands as a COMMANDS rule to the line containing the system you wish to execute the uux command from in the <i>/etc/uucp/Permissions</i> file.
No mail is sent to the root login when uuclean.sh is run.	Verify that the <i>/etc/uucp/Config</i> file contains MAILFLAG=TRUE in it.
When I execute uustat -q , I see something similar to the following message: ASSERT ERROR CAN'T UNLINK /var/spool/uucp/jon0/Z/C.jon0Z126b (uucico)	Using the uustat -k command while the job was being processed may have caused this message. The file(s) may or may not have been transmitted successfully.

Table 23-2. UUCP Troubleshooting Corrective Action (Contd)

Symptom	Corrective Action
My performance is bad, is there anything I can do to increase the throughput?	The UUCP system deployed on the 3B21D computer will support two baud (9.6 and 19.2K) rates and two protocols (g and G). The optimal speeds can be achieved by using 19.2K with the "G" protocol. The baud is selected via the ttop form in the Equipment Configuration Database (ECD). The protocol is selected by the remote system. Refer to the vendor documentation for the UUCP implementation on the remote system for information on how to configure the system to select the "G" protocol. For example, on a <i>Solaris</i> * system, you may change the <i>Devices</i> file to contain a line similar to: Direct,G(7,128)g /dev/term/b - 19200 direct. For the "G" protocol on the 3B21D computer, packet sizes of 64K and 128K are supported.
I keep getting an error about my sequence check failing.	The Sequence file on the 3B21D computer (<i>/var/uucp/.Admin/SQFILE</i>) has entries that look like the following: Machine number Where "Machine" is the remote system and "number" is a sequence number from 01 to 1998 that both systems keep in sync. If security is breached, it is possible for these to get out of sync and may need to be reset. Refer to the vendor documentation for the remote system's UUCP implementation on information about where to find the Sequence file. These numbers can be set to any value between 1 and 1998 but must be identical. In addition, when these numbers are found out of sequence, measures should be taken to tighten security (all system passwords and uucp login names should be changed) and additional steps may be required to assure that the system has not been compromised. The SQFILE will contain a time stamp when the last update to the entry was made. This can be useful in determining if security has been breached.
I am running out of disk space to transfer files.	The file system used to spool incoming or outgoing jobs can run out of disk space and prevent jobs from being sent or received. The <i>crontab</i> file must be modified to run the uuclean.sh at least once a day.

*

Sun, Sun Microsystems, the Sun logo, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Table 23-2. UUCP Troubleshooting Corrective Action (Contd)

Symptom	Corrective Action
uuclean.sh terminates without cleaning up the <i>/var</i> partition, usually with an error message: “/usr/lib/uucp/uuclean.sh fatal error: can’t write /var/uucp/.Old/Old-logx” or “/usr/lib/uucp/uuclean.sh: file-name: cannot create”	The <i>/var</i> partition is probably out of inodes. To verify this, enter: fd /var and check the inode count. If this is the case, you will need to examine the <i>/var</i> and its subdirectories manually and remove enough files so uuclean.sh can be used to clean up the <i>/var</i> partition. If there are inodes available, seek technical assistance to determine why the file cannot be written by uucp.
Connection from remote system to local system is rejected with “LCK” return code indicating the lock file for the remote system could not be set on the local system, but uustat on the local system does not indicate the remote system is locked or talking. (Sun* Uutry debug output is: imsg > ^PRLCK^@msg-RLCK HANDSHAKE FAILED: REMOTE HAS A LCK FILE FOR ME).	The <i>/var</i> partition is probably out of inodes. To verify this, enter: fd /var and check the inode count. If this is the case, you will need to examine the <i>/var</i> and its subdirectories manually and remove enough files so uuclean.sh can be used to clean up the <i>/var</i> partition. If there are inodes available, seek technical assistance to determine why the file cannot be written by uucp.

* Sun, Sun Microsystems, the Sun logo, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Error Code Interpretation

Error Messages

This section lists the error messages associated with UUCP utilities. There are two types of error messages. ASSERT errors are recorded in the */var/uucp/.Admin/errors* file. STATUS errors are recorded in individual system files found in the */var/uucp/.Status* directory.

ASSERT Messages

When a process is aborted, ASSERT error messages are recorded in the */var/uucp/.Admin/errors* file. These messages include the filename, sccsid, line number, and the text listed. The following is an example format of the ASSERT message:

```
ASSERT ERROR (uucico) pid: 77660236 (6/6-11:16:56) CAN'T CHDIR /var/spool/uucp
(2) [FILE: ../com/cico.c, LINE: 245]
```

In most cases, these errors are the result of file system problems. The “errno” (when present) should be used to investigate the problem. If “errno” is present in a message, it is shown as () in the following list.

CAN'T OPEN	An open() or fopen() failed.
CAN'T WRITE	A write(), fwrite(), fprintf(), etc. failed.
CAN'T READ	A read(), fgets(), etc. failed.
CAN'T CREATE	A create() call failed.
CAN'T ALLOCATE	A dynamic allocation failed.
CAN'T LOCK	An attempt to make a LCK (lock) file failed. In some cases, this is a fatal error.
CAN'T STAT	A stat() call failed.
CAN'T CHMOD	A chmod() call failed.
CAN'T CHOWN	A chown() call failed.

CAN'T LINK	A <code>link()</code> call failed.
CAN'T CHDIR	A <code>chdir()</code> call failed.
CAN'T UNLINK	A <code>unlink()</code> call failed.
WRONG ROLE	This is an internal logic problem.
CAN'T MOVE TO CORRUPTDIR	An attempt to move bad C. or X. files to the <i>/var/spool/uucp/Corrupt</i> directory failed. The directory is probably missing or has wrong modes or owner.
CAN'T CLOSE	A <code>close()</code> or <code>fclose()</code> call failed.
FILE EXISTS	The creation of a C. or D. file is attempted, but the file exists. This occurs when there is a problem with the Sequence file access. Usually indicates a software error.
CAN'T FORK	An attempt to fork and exec failed. The current job should not be lost but will be attempted later (uuxqt). No action needs to be taken.
BAD UID	Could not find <i>/etc/passwd</i> entry for euid check <i>/etc/passwd</i> file.
BAD LOGIN_UID	Could not find <i>/etc/passwd</i> entry for uid check <i>/etc/passwd</i> file.
REMOTE SYSTEM CAN'T CREATE TEMP FILE	Error on remote system run command on remote with debug turned on and analyze output.
BAD LINE	Found bad line in the device file; check the device for file errors.
SYSLST OVERFLOW	Too many systems in the command line; command line too complex
TOO MANY SAVED C FILES	Too many C files required by the command; command too complex.
BAD SPEED	Invalid speed in systems file; check systems file for invalid speeds

RETURN FROM fixline ioctl	STTY failed on device; verify device is operational for standard <i>UNIX</i> System I/O.
PERMISSIONS file: BAD OPTION	Invalid keyword in <i>/etc/uucp/Permissions</i> ; correct <i>Permissions</i> file.
PKXSTART RET	Write failed on device during startup; verify device is operational for standard <i>UNIX</i> system I/O.
PKCGET READ	Read failed on device; verify device is operational for standard <i>UNIX</i> system I/O.
TOO MANY LOCKS	Too many locked files required by the command; command is too complex.
XMV ERROR	Could not move or copy a file; check file system space problems.

STATUS Messages

STATUS error messages are messages that are stored in the */var/spool/uucp/.Status* directory. This directory contains a separate file for each remote system that the 3B21D computer attempts to communicate with. These individual system files contain status information on the attempted communication whether it was successful or not.

The following is an example format of the STATUS message:

```
uucp jon0 (6/6-11:28:54,78970956,0) FAILED (BAD LOGIN/MACHINE COMBINATION)
```

The following is a list of the most common error messages that may appear in these files.

OK	Things are OK.
NO DEVICES AVAILABLE	There is currently no device available for the call. Check to see that there is a valid device in the <i>Devices</i> file for the particular system. Check the <i>Systems</i> file for the device to be used to call the system.

WRONG TIME TO CALL	A call was attempted to be placed to the system at a time other than what is specified in the <i>Systems</i> file.
TALKING	File transfer is in progress.
LOGIN FAILED	The login for the given system failed. It could be a wrong login/password, wrong number, a very slow system, or failure in getting through the Dialer-Token-Pairs script.
CONVERSATION FAILED	The conversation failed after successful startup. This usually means that one side went down, the program aborted, or the line (link) was dropped.
DIAL FAILED	The remote system never answered. It could be a bad dialer or the wrong phone number.
BAD LOGIN/MACHINE COMBINATION	The system called us with a login/system name that does not agree with the <i>Permissions</i> file. This could be an attempt by an unauthorized user to masquerade as a valid login or system!
DEVICE LOCKED	The calling device to be used is currently locked and in use by another process.
ASSERT ERROR	An ASSERT error occurred. Check the <i>/var/spool/uucp/.Admin/errors</i> file for the error message and refer to the section "ASSERT Messages".
SYSTEM NOT IN Systems	The system is not in the <i>Systems</i> file.
CAN'T ACCESS DEVICE	The device tried does not exist or the modes are wrong. Check the appropriate entries in the <i>Systems</i> and <i>Devices</i> files.
DEVICE FAILED	The open of the device failed.
WRONG MACHINE NAME	The called system is reporting a different name than expected.

CALLBACK REQUIRED	The called system requires that it calls the 3B21D computer.
REMOTE HAS A LCK FILE FOR ME	The remote site has an <i>LCK</i> file for the 3B21D computer. They could be trying to call a local system. An <i>LCK</i> file may be hung.
REMOTE DOES NOT KNOW ME	The remote system does not have the node name of the 3B21D computer in its <i>Systems</i> file.
BAD SEQUENCE CHECK	The remote system and local system sequence numbers are not in sync.
REMOTE REJECT AFTER LOGIN	The login used by the 3B21D computer to log in does not agree with what the remote system was expecting.
REMOTE REJECT, UNKNOWN MESSAGE	The remote system rejected the communication with the 3B21D computer for an unknown reason. The remote system may not be running a compatible version of UUCP utilities.
STARTUP FAILED	Login succeeded, but initial handshake failed.
CALLER SCRIPT FAILED	This is usually the same as "DIAL FAILED". However, if it occurs often, suspect the caller script in the <i>Dialers</i> file. Uutry is not supported to check the caller script. Try another method.

Error Codes and Meaning

When a UUCP process fails, it returns an error code to the parent shell. Error codes are also printed to standard error or the debug output file if the debugging level is set for the process. The debugging level for a particular process can be set using the *config* file. See section "Config File" for more information.

Table 23-3 describes the error code and its meaning. Some of the UUCP processes access the common software code. The error codes marked as "COMMON," describe the errors printed from the common code.

Table 23-3. Error Codes and Definitions

Process	Error Code	Meaning
COMMON	-1 to -127	Function returned with errno = (-n)
COMMON	0 to 32	Caught signal number n
COMMON	51	Cannot open STAT file for writing
COMMON	52	EUID not UUCPID
COMMON	53	Cannot access <i>Permissions</i> file
COMMON	54	Cannot get MNTSTAT
COMMON	55	Spool mount point mounted read-only
COMMON	56	Spool mount point not found
COMMON	57	EXECLE failed for UUCICO
COMMON	58	Bad node name
UUCHECK	101	Invalid login user
UUCHECK	102	Login user not ROOT or uucp
UUCHECK	103	Usage - invalid option
UUCHECK	104	Usage - too many parameters
UUCHECK	105	Cannot read <i>Permissions</i> file
UUCHECK	110	UUCHECK MALLOC failed for PUBDIR
UUCICO	81	-s system not in <i>Systems</i> file
UUCICO	82	No -s specified
UUCICO	83	Trying to uucico to this system
UUCICO	84	Invalid interface specified
UUCICO	85	Slave setup failed
UUCICO	86	Cannot create lock for remote system
UUCICO	87	Bad login for remote system
UUCICO	88	Exiting to setup call back
UUCICO	89	Sequence check previously failed for remote system

Table 23-3. Error Codes and Definitions (Contd)

Process	Error Code	Meaning
UUCICO	90	Sequence check failed for remote system
UUCICO	91	Cannot call remote system
UUCICO	92	Currently talking with the remote system
UUCICO	93	CONN() failed
UUCICO	94	Login failed
UUCICO	95	Wrong remote system
UUCICO	96	Waiting for IMMSG failed
UUCICO	97	Remote rejected handshake
UUCICO	98	Startup timed out
UUCICO	99	Startup failed
UUCICO	100	Remote system unknown
UUCICO	101	CNTRL() failed
UUCICO	102	Cannot create TM file
UUCICO	103	ACU write failed
UUCICO	104	Bad login ID
UUCICO	105	Cannot access -d <i>spool</i> directory
UUCICO	106	Invalid -r option
UUCICO	107	-s login name and remote system name security check failed
UUCICO	108	-u login user ID invalid
UUCICO	109	Usage error - unknown option
UUCICO	110	Login name when running as master invalid
UUCICO	111	3B21D computer interface not a UN582
UUCICO	112	IMMSG from remote failed
UUCICO	113	UUCICO timed out waiting for IMMSG from remote
UUCICO	114	Master login name and remote system name security check failed
UUCICO	115	Caught a signal whose value is 0
UUCICO	116	Callback keyword not valid
UUCICO	117	savstk() function call failed
UUCICO	118	Caught a Phase 1
UUCLEANUP	101	Login ID security check failed
UUCLEANUP	102	User not uucp or ROOT
UUCLEANUP	103	Usage - invalid option
UUCLEANUP	104	Usage - invalid option for CDWXo
UUCLEANUP	105	Usage - too many parameters
UUCLEANUP	106	Cannot cd to spooler directory
UUCLEANUP	107	Cannot open spooler directory

Table 23-3. Error Codes and Definitions (Contd)

Process	Error Code	Meaning
UUCP	101	Usage error
UUCP	102	Cannot cd to workspace directory
UUCP	103	Bad "TO" system
UUCP	104	Bad "FROM" system
UUCP	105	File permission access denied
UUCP	106	User login ID invalid
UUCP	107	Invalid source filename
UUCP	108	Source filename expansion failed
UUCP	109	Filename expansion failed for remote UUCP
UUCP	110	Filename expansion failed for remote forward
UUNAME	101	Invalid login ID
UUNAME	102	Invalid option -c
UUNAME	103	Usage - invalid option
UUNAME	104	Usage - options mutually exclusive (-m -l)
UUNAME	106	Cannot access <i>Systems</i> file
UUSTAT	101	Invalid login ID
UUSTAT	102	-s invalid system
UUSTAT	103	-u invalid user ID
UUSTAT	104	Usage - invalid option
UUSTAT	105	Usage - too many parameters
UUSTAT	106	Usage - invalid option combination
UUSTAT	107	Cannot open <i>STAT</i> directory
UUSTAT	108	Cannot open <i>LCK</i> directory
UUSTAT	109	Cannot open <i>spool</i> directory
UUSTAT	110	Invalid JOBID
UUSTAT	111	Cannot STAT JOB file
UUSTAT	112	Cannot READ JOB file
UUSTAT	113	JOB file has bad format
UUSTAT	114	Not owner, ROOT or uucp
UUSTAT	115	Cannot find <i>D.</i> file for JOBID
UUSTAT	116	JOB file removal or time stamp update failed
UUSTAT	117	Cannot open <i>LCK</i> directory for -p option

Table 23-3. Error Codes and Definitions (Contd)

Process	Error Code	Meaning
UUXQT	101	FORK() of command failed
UUXQT	102	Invalid login ID
UUXQT	103	Usage - invalid option
UUXQT	104	Usage - too many parameters
UUXQT	106	EXECL failed for <i>/bin/sh</i>
UUXQT	107	SAVSTK() call failed
UUXQT	108	Caught a Phase 1
UUX	101	Cannot cd to workspace directory
UUX	102	Bad remote system name for command
UUX	103	Bad system name for a file
UUX	104	Bad filename expansion for redirection
UUX	105	Bad filename expansion for local file - local command
UUX	106	Bad filename expansion for local file - remote command
UUX	107	User cannot get file status
UUX	108	User cannot access file
UUX	109	User cannot make local file copy
UUX	110	Bad filename expansion for remote file - local command
UUX	111	Bad filename expansion for receive file
UUX	112	Bad filename expansion for remote file - remote command
UUX	113	Invalid usage
UUX	114	Invalid login ID

uucheck Messages and Corrections

The **uucheck** messages for Errors, Warnings, Cautions, and Internal Errors are shown in the following tables.

- uucheck Error Messages - Table 23-4
- uucheck Warning Messages - Table 23-5
- uucheck Caution Messages - Table 23-6
- uucheck Internal Error Messages - Table 23-7.

Table 23-4. uuccheck Error Messages

uuccheck Error Message	Corrective Action
Cannot get UID=0 name — Should be "root"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Cannot get UID=5 name — Should be "uucp"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Can't open file-name	Seek technical assistance to find lost file.
Cannot open file-name for reading.	Seek technical assistance to find lost file.
eaio0 record contains an invalid system name "bad_string".	Add NODENAME keyword entry to <i>/etc/uucp/Config</i> .
<i>/etc/passwd</i> entry "login-names" has no PERMISSIONS file LOGNAME entry.	Either add login-names to <i>Permissions</i> file or remove it from <i>/etc/passwd</i> .
File name — Not found	Seek technical assistance to find lost file.
File name — Directory mode check failed (got 0xxx, expected 0yyyy)	Seek technical assistance to correct directory mode.
File-name — File format check failed (0xxx)	Seek technical assistance to correct incorrect file format.
File-name — File mode check failed (got 0xxx, expected 0yyy)	Seek technical assistance to correct file mode.
File-name — uid check failed (got nnn, expected yyy)	Seek technical assistance to correct uid.
Invalid GID=nnn for login user "login-names" - should be 5.	seek technical assistance to fix <i>/etc/passwd</i> file.
Invalid GID=nnn for login user "login-names"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Invalid Login Name "login-names" — Contains "uucp"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Invalid LOGNAME "login-names" — Contains "uucp"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Invalid UID=uid-value for login user "login-name"	Seek technical assistance to fix <i>/etc/passwd</i> file.

Table 23-4. uuccheck Error Messages (Contd)

uuccheck Error Message	Corrective Action
Login user "login-name" should not have login shell "bad_string"	Seek technical assistance to fix <i>/etc/passwd</i> file.
Login user "login-name" with UID=5 — Not "uucp"	Seek technical assistance to fix <i>/etc/passwd</i> file.
LOGNAME log-name — MYNAME rule "bad string" has invalid characters - using default node-name	Fix MYNAME rule in <i>Permissions</i> file.
LOGNAME log-name — UUCP cannot callout - CALLBACK rule invalid	Remove CALLBACK rule from the <i>Permissions</i> file.
LOGNAME log-name has no VALIDATE rule	Add VALIDATE rule to the <i>Permissions</i> file.
LOGNAME log-name incompatible with system name "node-name"	Choose another name for LOGNAME in the <i>Permissions</i> file.
LOGNAME log-name — keywordtype-name directory "dir-name" not found	Seek technical assistance.
LOGNAME log-name has invalid CALLBACK rule "bad-string" (yes/no)	Remove CALLBACK rule from the <i>Permissions</i> file.
LOGNAME log-name has invalid REQUEST rule "bad-string" (yes/no)	Fix REQUEST rule in the <i>Permissions</i> file.
LOGNAME log-name has invalid SENDFILES rule "bad-string" (yes/no)	Fix SENDFILES rule in the <i>Permissions</i> file.
LOGNAME log-name has more than one system name in VALIDATE rule	Fix the VALIDATE rule in the <i>Permissions</i> file.
MACHINE mch-name — MYNAME rule "bad-string" has invalid characters - using default "node-name"	Fix MYNAME rule in the <i>Permissions</i> file.
MACHINE mch-name — UUCP cannot callout - CALLBACK rule invalid	Remove CALLBACK rule from the <i>Permissions</i> file.
MACHINE mch-name has no VALIDATE rule	Add VALIDATE rule to the <i>Permissions</i> file.
MACHINE mch-name incompatible with system name "node-name"	Choose another name for LOGNAME in the <i>Permissions</i> file.
MACHINE mch-name — keywordtype-name directory "dir-name" not found	Seek technical assistance.
MACHINE mch-name has invalid CALLBACK rule "bad-string" (yes/no)	Remove CALLBACK rule from the <i>Permissions</i> file.

Table 23-4. uuccheck Error Messages (Contd)

uuccheck Error Message	Corrective Action
MACHINE mch-name has invalid REQUEST rule "bad-string" (yes/no)	Fix REQUEST rule in the <i>Permissions</i> file.
MACHINE mch-name has invalid SENDFILES rule "bad-string" (yes/no)	Fix SENDFILES rule in the <i>Permissions</i> file.
MACHINE mch-name has more than one system name in VALIDATE rule	Fix VALIDATE rule in the <i>Permissions</i> file.
MAILFLAG has invalid value "bad_string" in <i>Config</i> — Should be TRUE or FALSE	Fix entry in <i>/etc/uucp/Config</i> .
No login user with UID=5	Seek technical assistance to fix <i>/etc/passwd</i> file.
PERMISSIONS file LOGNAME entry "log-name" — Has no entry in <i>/etc/passwd</i>	Either add login-names to <i>/etc/passwd</i> or remove it from the <i>Permissions</i> file
SPOOLMOUNT has invalid value "bad_string" in <i>Config</i> — Should be TRUE or FALSE	Fix entry in <i>/etc/uucp/Config</i> .
User for UID=0 — Not "root" (bad login-name).	Seek technical assistance to fix <i>/etc/passwd</i> file.
User for UID=5 — Not "uucp" (bad login-name).	Seek technical assistance to fix <i>/etc/passwd</i> file.

Table 23-5. uucheck Warning Messages

uucheck Warning Message	Corrective Action
DEBUGpgm=nnn in <i>Config</i> out of range — Needs to be between xxx to yyy.	Fix DEBUGpgm=nnn entry in <i>/etc/uucp/Config</i> .
Invalid syntax in <i>Config</i> file line "bad-string"	Fix the <i>Config</i> file line.
Login user "login-name" has no password set.	Seek technical assistance to fix <i>/etc/passwd</i> file.
LOGNAME log-name COMMAND rule "ALL" will allow ANY command to be executed!! This could be a serious security problem!!	It is HIGHLY recommended to remove the ALL keyword from the COMMAND rule.
LOGNAME log-name COMMAND rule "cmd-string" not found or not executable by uucp	If "cmd-string" is incorrect, fix "cmd-string" in the <i>Permissions</i> file. Otherwise, if not found, seek technical assistance. If found but not executable by UUCP, verify that the login ID of the remote user can at least execute it.
LOGNAME log-name keywordtype-name "dir-name" not a directory	Seek technical assistance.
LOGNAME log-name keywordtype-name directory "dir-name" Causes a login directory (dir-name) to be writable by uucp jobs	It is STRONGLY recommended that the login directory for the login ID be changed to a directory not writable by uucp jobs. Either change the offending rule in the <i>Permissions</i> file or the login entry in <i>/etc/passwd</i> .
LOGNAME log-name keywordtype-name directory "dir-name" Causes a login directory (dir-name) to be readable by uucp jobs	It is STRONGLY recommended that the login directory for the login ID be changed to a directory not readable by uucp jobs. Either change the offending rule in the <i>Permissions</i> file or the login entry in <i>/etc/passwd</i> .
LOGNAME log-name keywordtype-name directory "dir-name" is a parent directory of a protected directory (dir-name).	Change the offending rule in the <i>Permissions</i> file.

Table 23-5. uucheck Warning Messages (Contd)

uucheck Warning Message	Corrective Action
LOGNAME log-name keywordtype-name directory "dir-name" is a protected directory (dir-name).	Change the offending rule in the <i>Permissions</i> file.
LOGNAME for user "log-name" — Is identical to user "uucp" login directory.	Seek technical assistance to fix <i>/etc/passwd</i> file.
MACHINE mch-name COMMAND rule "ALL" will allow ANY command to be executed!! This could be a serious security problem!!	It is HIGHLY recommended to remove the ALL keyword from the COMMAND rule.
MACHINE mch-name COMMAND rule "cmd-string" not found or not executable by uucp	If "cmd-string" is incorrect, fix "cmd-string" in the <i>Permissions</i> file. Otherwise, if not found, seek technical assistance. If found but not executable by UUCP, verify that the login ID of the remote user can at least execute it.
MACHINE mch-name keywordtype-name "dir-name" not a directory	Seek technical assistance.
MACHINE mch-name keywordtype-name directory "dir-name" Causes a login directory (dir-name) to be writable by uucp jobs	It is STRONGLY recommended that the login directory for the login ID be changed to a directory not writable by uucp jobs. Either change the offending rule in the <i>Permissions</i> file or the login entry in <i>/etc/passwd</i> .
MACHINE mch-name keywordtype-name directory "dir-name" Causes a login directory (dir-name) to be readable by uucp jobs	It is STRONGLY recommended that the login directory for the login ID be changed to a directory not readable by uucp jobs. Either change the offending rule in the <i>Permissions</i> file or the login entry in <i>/etc/passwd</i> .
MACHINE mch-name keywordtype-name directory "dir-name" is a parent directory of a protected directory (dir-name).	Change the offending rule in the <i>Permissions</i> file.

Table 23-5. uucheck Warning Messages (Contd)

uucheck Warning Message	Corrective Action
MACHINE mch-name keywordtype-name directory "dir-name" is a protected directory (dir-name).	Change the offending rule in the <i>Permissions</i> file.
MACHINE for user "mch-name" — Is identical to user "uucp" login directory.	Seek technical assistance to fix <i>/etc/passwd</i> file.
MAXDEBUGLEVEL=nnn in <i>Config</i> out of range — Needs to be between xxx to yyy.	Fix MAXDEBUGLEVEL=nnn entry in <i>/etc/uucp/Config</i> .
MAXSECURITYLEVEL=nnn in <i>Config</i> out of range — Needs to be between xxx to yyy.	Fix MAXSECURITYLEVEL=nnn entry in <i>/etc/uucp/Config</i> .

Table 23-6. uucheck Caution Messages

uucheck Caution Message	Corrective Action
Login user "login-name" has password aging enabled. This is likely to cause administrative complexity!!	Seek technical assistance to fix <i>/etc/passwd</i> file.
LOGNAME log-name NOWRITE "dir-name" in both WRITE and NOWRITE rules. The NOWRITE rule will override the WRITE rule!!	Change the offending rule in the <i>Permissions</i> file.
LOGNAME log-name NOREAD "dir-name" in both READ and NOREAD rules. The NOREAD rule will override the READ rule!!	Change the offending rule in the <i>Permissions</i> file.
LOGNAME log-name NOWRITE "dir-name" not a sub-directory of any WRITE rule	Change the offending rule in the <i>Permissions</i> file.
LOGNAME log-name NOREAD "dir-name" not a sub-directory of any READ rule	Change the offending rule in the <i>Permissions</i> file.
LOGNAME log-name keywordtype-name directory "dir-name" — NO UUCP read permissions	Verify that the login ID of the remote user can at least read the directory.
LOGNAME log-name keywordtype-name directory "dir-name" — NO UUCP write permissions	Verify that the login ID of the remote user can at least write the directory.
LOGNAME log-name keywordtype-name directory "dir-name" — not readable by uucp	Verify that the login ID of the remote user can at least read the directory.
LOGNAME log-name keywordtype-name directory "dir-name" — not writable by uucp	Verify that the login ID of the remote user can at least write the directory.
LOGNAME log-name has no MACHINE rule for the same entry	It is recommended that each LOGNAME entry have a MACHINE rule. Add one to the LOGNAME entry in the <i>Permissions</i> file.
LOGNAME log-name keywordtype-name directory "dir-name" — writable by others	It is HIGHLY recommended that only the PUBDIR directory be writable by others. Change the offending rule in the <i>Permissions</i> file.

Table 23-6. uucheck Caution Messages (Contd)

uucheck Caution Message	Corrective Action
MACHINE log-name NOWRITE "dir-name" in both WRITE and NOWRITE rules. The NOWRITE rule will override the WRITE rule!!	Change the offending rule in the <i>Permissions</i> file.
MACHINE log-name NOREAD "dir-name" in both READ and NOREAD rules. The NOREAD rule will override the READ rule!!	Change the offending rule in the <i>Permissions</i> file.
MACHINE log-name NOWRITE "dir-name" not a sub-directory of any WRITE rule	Change the offending rule in the <i>Permissions</i> file.
MACHINE log-name NOREAD "dir-name" not a sub-directory of any READ rule	Change the offending rule in the <i>Permissions</i> file.
MACHINE log-name keywordtype-name directory "dir-name" — NO UUCP read permissions	Verify that the login ID of the remote user can at least read the directory.
MACHINE log-name keywordtype-name directory "dir-name" — NO UUCP write permissions	Verify that the login ID of the remote user can at least write the directory.
MACHINE log-name keywordtype-name directory "dir-name" — not readable by uucp	Verify that the login ID of the remote user can at least read the directory.
MACHINE log-name keywordtype-name directory "dir-name" — not writable by uucp	Verify that the login ID of the remote user can at least write the directory.
MACHINE log-name has no LOGNAME rule for the same entry	It is recommended that each MACHINE entry have a LOGNAME rule. Add one to the MACHINE entry in the <i>Permissions</i> file.
MACHINE log-name keywordtype-name directory "dir-name" — writable by others	It is HIGHLY recommended that only the PUBDIR directory be writable by others. Change the offending rule in the <i>Permissions</i> file.

Table 23-7. uucheck Internal Error Messages

uucheck Internal Error Message	Corrective Action
PATH= field not found in "bad-path-string".	Seek technical assistance.
Cannot malloc for cmd path search full-cmd-path-name-string.	Seek technical assistance.
cmdOK check failed in permission.c	Seek technical assistance.
Malloc failed for PATH "path-string".	Seek technical assistance.
Malloc failed for "slash_dirname".	Seek technical assistance.
Malloc failed for VALIDATE "remote-name".	Seek technical assistance.
Internal Error: malloc failed for ent.	Seek technical assistance.

UUCP Administrative Files

Temporary (TM.) File

The *TM* data files are created under the *spool* directory; for example, */var/uucp/XXXX*) when receiving a file from another system. The directory “XXXX” has the same name as the remote system that is sending the file. The temporary data filenames have the format

TM.pid.ddd

where:

<i>pid</i>	is a process-id
<i>ddd</i>	is a sequential number starting at zero.

After the entire file is received, the *TM.* file is moved to the pathname specified in the command line. If processing is abnormally terminated, the *TM.* file may remain in the “XXXX” directory. These files should be periodically removed by **uuclean.sh**.

Lock (LCK) File

Lock files are created in the */var/locks* directory for each device in use. Lock files prevent duplicate conversations and multiple attempts to use the same calling device. The filename has the format

LCK.str

where *str* is either a device or computer name. The files may be left in the *spool* directory if runs abort. These files are periodically removed by **uuclean.sh**

Work (C.) File

Work files are created in a *spool* directory when transfers or remote command executions have been queued for a remote computer. Their names have the format

C.sysxxxx

where *sys* is the name of the remote computer, and *xxxx* is the four-digit job sequence number assigned by **uucp**. These files are periodically removed by **uuclean.sh**.

Work files contain the following information:

- Full pathname of the file to be sent or requested
- Full pathname of the destination or ~user/filename
- User login name
- List of options
- Mode bits of the source file
- Remote user's login name to be notified upon completion of the transfer.

Data (D.) File

Data files are created when it is specified in the command line to copy the source file to the *spool* directory. Their names have the format

D.sysxxxx

where *sys* is the name of the remote computer, and *xxxx* is the four-digit job sequence number assigned by **uucp**. The four-digit job sequence number may be followed by a subjob number which is used when there are several *D.* files created for a work (*C.*) file.

Execute (X.) File

Execute files are created in the *spool* directory prior to remote command executions. Their names have the format

X.sysxxxx

where *sys* is the name of the remote computer, and *xxxx* is the four-digit sequence number assigned by **uucp**.

Execute files contain the following information:

- Requester's login and computer name
- Name of file(s) required for execution
- Input to be used as the standard input to the command string
- Computer and filename to receive standard output from the command execution
- Command string
- Option lines for return status requests.

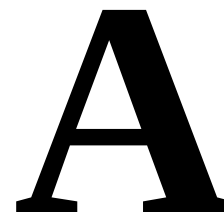
UUCP Manual Pages

The *UNIX* system manual pages supported by the UUCP implementation on the 3B21D computer are:

- **mail**
- **uuccheck**
- **uucleanup**
- **uucp**
- **uulog**
- **uuname**
- **uusetup**
- **uustat**
- **uux**

Descriptions for each of these UUCP commands/programs are provided in 235-700-200, *UNIX RTR Operating System Reference Manual*.

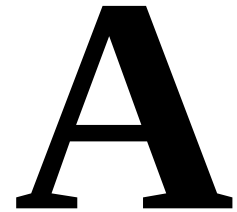
3B20D Computer References



Contents

Introduction	A-1
Control Unit Diagnostic Test Sequence	A-2
Diagnostic Hierarchy	A-4
■ CU Diagnostic Hierarchy	A-4
■ DFC Hardware Diagnostic Hierarchy	A-4
■ IOP Diagnostic Hierarchy	A-5
Test Procedures	A-6
■ CU Suggested Routine Test Procedure	A-6
■ DFC Suggested Routine Test Procedure	A-8
■ IOP Suggested Routine Test Procedure	A-9
Supplementary Phases	A-11
Main Store Failure Data	A-23
Processor Recovery Messages	A-29
Diagnostic Information Tables — CU, DFC, and IOP	A-30

3B20D Computer References



Introduction

This appendix contains 3B20D computer reference information on the following topics:

- Control Unit (CU) Diagnostic Test Sequence
- Diagnostic Hierarchy
- Test Procedures
- Supplementary Phases
- Main Store Failure Data
- Processor Recovery Messages (PRMs)
- Diagnostic Information Tables – CU, Disk File Controller (DFC), and Input/Output Processor (IOP).

Control Unit Diagnostic Test Sequence

The sequence of CU diagnostics tests for the 3B20D computer is shown in the following list:

- Maintenance Channel (MCH) communication
- Microcontrol clock
- Gating from/to microstore data bus
- Gating from/to microstore address bus
- Gating from/to bidirectional gating bus
- Gating from maintenance bus
- Initialization
- Microprogram control timing
- Microaddress sequencing
- Immediate data gating to bidirectional gating register
- Source to destination bus gating through the Arithmetic Logic Unit (ALU)/Remote Measurement Unit (RMU)
- Register gating (general registers)
- Register gating (temporary registers)
- Store address register
- Store data register
- Present address register
- Store instruction register
- Instruction register
- Microstore contents
- Store control and Main Store (MAS) contents
- RMU
- ALU
- Interrupt registers
- Timing registers
- Error response
- Emergency Action Interface (EAI)
- Store controller

- MAS contents
- Store error checker
- Isolate update and other store mode
- Double store read
- Address Translation Buffer (ATB) access logic
- ATB compare and protect logic
- ATB
- Cache access logic
- Cache compare logic
- Cache memory
- Central Control Input/Output (CCIO) bus
- Direct Memory Access (DMA)
- Input/Output (IO) channels
- Utility Circuit (UC).

Diagnostic Hierarchy

CU Diagnostic Hierarchy

1. Central Control (CC)
2. Main Store Controller (MASC), includes memory arrays
3. Store Address Translator (SAT)
4. Cache Storage Unit (CSU)
5. Direct Memory Access Controller (DMAC)
6. Direct Memory Access Channel (DMCH) and Programmed IO channel
7. Utility Circuit (UC).

DFC Hardware Diagnostic Hierarchy

DFC Hardware

1. DFC
2. Small Computer System Interface (SCSI) Bus (SBUS) (present for SCSI DFCs only)*
3. Moving Head Disk (MHD)

DFC 2 Hardware

1. DFC
2. SBUS (present for SCSI DFCs only)*
3. Moving Head Disk (MHD)
4. Magnetic Tape Unit (MTU).

* SBUS is a restorable unit but does not have a diagnostic.

IOP Diagnostic Hierarchy

IOP Hardware

1. IOP
2. Maintenance Teletypewriter Controller (MTTYC)
3. Teletypewriter Controller (TTYC)
4. Magnetic Tape Controller (MTC)
5. Synchronous Data Link Controller (SDLC)
6. Scan Point and Signal Distributor Controller (SCSDC)
7. Direct User Interface Controller (DUIC).

IOP 2 Hardware

1. IOP
2. MTTYC
3. TTYC
4. SDLC
5. SCSDC
6. DUIC.

Test Procedures

The following group of procedures is suggested as a complete routine manual diagnostic plan. These suggested test procedures should be performed once a week.

For the procedure to run routine demand diagnostics on the 3B20D computer, refer to the Non Trouble Procedure (NTP) for performing system demand diagnostics in one of the following information products (IPs):

- 254-302-831, *Routine Tasks (PDS Formats)*
- 254-302-841, *Routine Tasks (MML Formats)*.

CU Suggested Routine Test Procedure

The following is the routine test procedure for the 3B20D computer CU. The commands use the following variables *a*, *b*, and *d*:

a = Unit under test

b = Subunit under test

d = Member number of the helper unit.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, DGN:CU a,TLP! In MML, DGN:CU=a:DATA,TLP;
2	Enter: In PDS, DGN:CU a,CC 0:PH 1-92,TLP! In MML, DGN:CU=a,CC=0:DATA,PH=1&&92,TLP;
3	Enter: In PDS, DGN:CU a,CC 0:PH 93,TLP! In MML, DGN:CU=a,CC=0:DATA,PH=93,TLP; (Enter Phase 93 as a separate request because manual intervention is required.)

Step	Procedure
4	Check for the correct sequence of PRM patterns on the Emergency Action Page of the Maintenance Teletypewriter (MTTY) and on the Receive-Only Printer (ROP). (Refer to Table A-2, CC Phase 93.)
5	Check for the correct sequencing of lamps on the EAI display next to the power switch on the CU frame. (Refer to Table A-2, CC Phase 93.)
6	Enter: In PDS, RMV:DFC d! In MML, RMV:DFC=d; (An IOP may alternatively be used as the helper unit.)
7	Enter: In PDS, DGN:CU a,CH 11:PH 40,DFC d,TLP! In MML, DGN:CU=a,CH=11:DATA,PH=40,DFC=d,TLP;
8	Enter: In PDS, RST:DFC d;UCL! In MML, RST:DFC=d:UCL;
9	Enter: In PDS, RST:CU a;UCL! In MML, RST:CU=a:UCL;
10	Enter: In PDS, SW:CU! In MML, SW:CU;

DFC Suggested Routine Test Procedure

Step	Procedure
1	Enter: In PDS, RMV:CU d! In MML, RMV:CU=d;
2	Enter: In PDS, DGN:DFC a:PH 15,CU d,TLP! In MML, DGN:DFC=a:DATA,PH=15,CU=d,TLP;
3	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d:UCL;
4	Enter: In PDS, SW:CU! In MML, SW:CU;
5	Enter: In PDS, RMV:CU d! In MML, RMV:CU=d;
6	Enter: In PDS, DGN:DFC a:PH 15,CU d,TLP! In MML, DGN:DFC=a:DATA,PH=15,CU=d,TLP;
7	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d:UCL!
8	Enter: In PDS, RST:DFC a! In MML, RST:DFC=a;

IOP Suggested Routine Test Procedure

The following is the routine test procedure for the 3B20D computer CU. The commands use the following variables *a*, *b*, and *d*:

a = Unit under test

b = Subunit under test

d = Member number of the helper unit.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, RMV:CU d! In MML, RMV:CU=d;
2	Enter: In PDS, DGN:IOP a:PH 15,CU d,TLP! In MML, DGN:IOP=a:DATA,PH=15,CU=d,TLP;
3	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d:UCL;
4	Enter: In PDS, SW:CU! In MML, SW:CU;
5	Enter: In PDS, RMV:CU d! In MML, RMV:CU=d;

Step	Procedure
6	Enter: In PDS, DGN:IOP a:PH 15,CU d,TLP! In MML, DGN:IOP=a:DATA,PH=15,CU=d,TLP;
7	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d:UCL!
8	Enter: In PDS, RST:IOP a! In MML, RST:IOP=a;
9	For each MTC connected to this IOP, check each Magnetic Tape (MT) connected to the MTC by mounting a diagnostic test tape on the tape drive; enter: In PDS, DGN:MTC a:PH 5,MT d,TLP! In MML, DGN:MTC=a:DATA,PH=5,MT=d,TLP;
10	Enter: In PDS, RST:MTC a! In MML, RST:MTC=a;

Supplementary Phases

The following tables cover the supplementary phases:

Table A-1	Recommended Frequency of Selected Supplementary Phases
Table A-2	Supplementary Phases for CU
Table A-3	Supplementary Phases for SMD DFC
Table A-4	Supplementary Phases for SCSI DFC
Table A-5	Supplementary Phases for IOP

Table A-1. Recommended Frequency of Selected Supplementary Phases

Phase	Area Tested	Frequency
CC 59	Test special registers.	Weekly
CC 93	EAI display tests.	Weekly
MASC 33	Test memory with data patterns main store.	Weekly
MASC 34	Test memory with data patterns MASC, MASA.	Weekly
MASC 36	Test memory with data patterns MASC, MASA.	Weekly
MASC 40	Test memory read and half-word clear MASC.	Weekly
CSU 90	Long-term cache exercise for Cache Storage Unit (CSU). Read and write to cache.	Monthly (40-minute test)
MTC 5	Test tape transport and formatter. Use diagnostic test tape.	Weekly

Table A-2. Supplementary Phases for CU

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CC	32, 35, 38	These CC phases test various circuit packs in the writable microstore. If the circuit pack is not in the equipment, the test will not be run. Phase 32 tests circuit pack UN48 or UN248 at slot 1. Phase 35 tests circuit pack UN48 or UN248 at slot 2. Phase 38 tests circuit pack UN48 or UN248 at slot 3.	REX, DEX, Demand	In PDS, DGN:CU a,CC b;REX! DGN:CU a,CC b;DEX! DGN:CU a,CC b:PH 32! DGN:CU a,CC b:PH 35! DGN:CU a,CC b:PH 38! In MML, DGN:CU=a,CC=b:REX; DGN:CU=a,CC=b:DEX; DGN:CU=a,CC=b:DATA,PH=32; DGN:CU=a,CC=b:DATA,PH=35; DGN:CU=a,CC=b:DATA,PH=38;
CC	59	CC Phase 59 tests CC interrupt logic circuits connected to units other than the CC. Note: Any units that are causing external error signals should be repaired before CC Phase 59 is run. Do not execute these tests unless all CUs have been diagnosed or the CU has been in service since the last time power was restored to the frame.	Demand	In PDS, DGN:CU a,CC b:PH 59! In MML, DGN:CU=a,CC=b:DATA,PH=59;
CC	93	CC Phase 93 tests the EAI Light-Emitting Diodes (LEDs) display (located next to the power switch on the CU) and the capability of the EAI to transmit Processor Recovery Messages (PRMs). A series of PRMs appears on the emergency action page at the MTTY and on the Receive-Only Printer (ROP). Then the lamps on the EAI display are cycled through the specific pattern for that phase.	DEX, Demand — run when the user is available to watch the EAI LEDs and display on the circuit pack, MTTY, and ROP.	In PDS, DGN:CU a,CC b;DEX! DGN:CU a,CC b:PH 93! In MML, DGN:CU=a,CC=b:DEX; DGN:CU=a,CC=b:DATA,PH=93;

Table A-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CC	93	All forced states on the EAI must be cleared before CC Phase 93 is run. Do this by using Poke 14 on the Emergency Action Page. Check PRMs displayed for those expected. Refer to the "Processor Recovery Messages" section for further explanation of PRMs. Observe the following patterns on the EAI board (TN10, TN11) of the CU under test: First, digits 0 through F are sequenced on the hexadecimal digit display. The sequence starts with 0-F-0 to warn you that this part of the phase is beginning. Next, each LED is lighted and then extinguished in a pattern running down the display. Later, lamp flashes may occur and should be ignored. If you do not observe these patterns, the test failed. A lamp display failure indicates that the EAI (TN10 or TN11) is defective. A PRM failure may be because of a defective EAI or a defective MTTY controller (TN83 and TN983B) in the IOP to which the affected MTTY or ROP is connected. Note: Since this failure is indicated visually, the phase may run All Tests Pass (ATP) in spite of display faults.		

Table A-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CSU	18	This R1 (nonenlarged) CSU Phase 18 exercises the mechanism that writes values from the main store to the cache. A selection of memory addresses are accessed using various modes. Then the contents of main memory, ATBs, and cache are compared. If no differences are found, the test continues. Differences are reported as errors. If no errors are found, CSU Phase 18 runs for 40 seconds. Note: Always run CSU Phase 18 with the RAW option.	REX, DEX, Demand — run whenever CC, SAT, cache, and main store tests pass, but trouble is suspected in those areas.	In PDS, DGN:CU a,CSU b;RAW,REX! DGN:CU a,CSU b;RAW,DEX! DGN:CU a,CSU b;RAW:PH 18! In MML, DGN:CU=a,CSU=b:RAW,REX; DGN:CU=a,CSU=b:RAW,DEX; DGN:CU=a,CSU=b:RAW:DATA,PH=18;
CSU	26	Enlarged CSU Phase 26 exercises the mechanism that writes values from the main store to the enlarged cache. A selection of memory addresses are accessed using various modes. Then the contents of main memory, ATBs, and cache are compared. If no differences are found, the test continues. Differences are reported as errors. If no errors are found, CSU Phase 26 runs for 40 seconds. Note: Always run CSU Phase 26 with the RAW option.	REX, DEX, Demand — run whenever CC, SAT, cache, and main store tests pass but trouble is suspected in those areas.	In PDS, DGN:CU a,CSU b;RAW,REX! DGN:CU a,CSU b;RAW,DEX! DGN:CU a,CSU b;RAW:PH 26! In MML, DGN:CU=a,CSU=b:RAW,REX; DGN:CU=a,CSU=b:RAW,DEX; DGN:CU=a,CSU=b:RAW:DATA,PH=26;

Table A-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CSU	90	CSU Phase 90 is the same as Phase 18 of the R1 CSU and Phase 26 of the enlarged CSU except it will run for 40 minutes if no fault is found. Note: Always run CSU Phase 90 with the RAW option.	Demand — run whenever an intermittent or data-dependent fault is suspected.	In PDS, DGN:CU a,CSU b;RAW:PH 90! In MML, DGN:CU=a,CSU=b:RAW:DATA,PH=90;
CH	40	CH Phase 40 tests data communication between the DMA channel (in the CU) and the bus interface controller (in a DFC or IOP). Specify the DFC or IOP in the input message as a helper unit. The helper unit must be Out of Service (OOS). Note: A valid phase only if the helper unit is <i>UNIX</i> ® RTR operating system, IOP, or DFC.	Demand	In PDS, DGN:CU a,CH b:PH 40,DFC d! or DGN:CU a,CH b:PH 40,IOP d! In MML, DGN:CU=a,CH=b:DATA,PH=40,DFC=d; or DGN:CU=a,CH=b:DATA,PH=40,IOP=d;
CH	44	CH Phase 44 tests reading the normal bootstrap data from the disk into main store. Specify either DFC 0 or DFC 1 as the helper unit, which must be OOS.	Demand — run if it is suspected that a particular CU and DFC combination will not boot.	In PDS, DGN:CU a,CH 11:PH 44,DFC d! In MML, DGN:CU=a,CH=11:DATA,PH=44,DFC=d;
MASC	33, 34, 36, 40	Memory patterns and access tests to the main memory.	DEX, Demand	In PDS, DGN:CU a,MASC b;DEX! DGN:CU a,MASC b:PH 33! DGN:CU a,MASC b:PH 34! DGN:CU a,MASC b:PH 36! DGN:CU a,MASC b:PH 40! In MML, DGN:CU=a,MASC=b:DEX; DGN:CU=a,MASC=b:DATA,PH=33; DGN:CU=a,MASC=b:DATA,PH=34; DGN:CU=a,MASC=b:DATA,PH=36; DGN:CU=a,MASC=b:DATA,PH=40;

Table A-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
MASC	41	Update mode and refresh data parity tests to entire main memory.	Demand	In PDS, DGN:CU a,MASC b:PH 41! In MML, DGN:CU=a,MASC=b:DATA,PH=41;
MASC	93	Test main and duplicate refresh address counters.	REX, DEX, Demand	In PDS, DGN:CU a,MASC b:REX! DGN:CU a,MASC b:DEX! DGN:CU a,MASC b:PH 93! In MML, DGN:CU=a,MASC=b:REX; DGN:CU=a,MASC=b:DEX; DGN:CU=a,MASC=b:DATA,PH=93;
MASC	95	Memory patterns and access tests to the main memory. Phase 95 allows limits on an interactive diagnostic to be set at run time. (Refer to the "Requesting Interactive Diagnostics" section in Chapter 6.)	Demand	In PDS, DGN:CU a,MASC b:PH 95! In MML, DGN:CU=a,MASC=b:DATA,PH=95;
MASC	96	Update mode and refresh data parity tests to entire main memory. Phase 96 allows limits on an interactive diagnostic to be set at run time. (Refer to the "Requesting Interactive Diagnostics" section in Chapter 6.)	REX, DEX, Demand	In PDS, DGN:CU a,MASC b:REX! DGN:CU a,MASC b:DEX! DGN:CU a,MASC b:PH 96! In MML, DGN:CU a,MASC b:REX; DGN:CU a,MASC b:DEX; DGN:CU=a,MASC=b:DATA,PH=96;
UC	90	UC Phase 90 tests the external output of trigger function 0. It also tests the external inputs to trigger functions 1, 2, and 3. Clip pin 400 to pin 405 on the backplane at the UC slot to run this phase. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 90! In MML, DGN:CU=a,UC=b:DATA,PH=90;

Table A-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
UC	91	UC Phase 91 tests the external output of trigger function 1. It also tests the external inputs to trigger functions 0, 2, and 3. Clip pin 401 to pin 405 on the backplane at the UC slot to run this phase. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 91! In MML, DGN:CU=a,UC=b:DATA,PH=91;
UC	92	UC Phase 92 tests the external output of trigger function 2. It also tests the external inputs to trigger functions 0, 1, and 3. Clip pin 402 to pin 405 on the backplane at the UC slot to run this phase. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 92! In MML, DGN:CU=a,UC=b:DATA,PH=92;
UC	93	UC Phase 93 tests the external output of trigger function 3. It also tests the external inputs to trigger functions 0, 1, and 2. Clip pin 403 to pin 405 on the backplane at the UC slot to run this phase. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 93! In MML, DGN:CU=a,UC=b:DATA,PH=93;

Table A-3. Supplementary Phases for SMD DFC

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DFC	15	DFC Phase 15 tests data communication between the bus interface controller (in the DFC) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU used as a helper unit must be OOS when the diagnostic is run, and a manual remove of the CU must precede the supplementary phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:DFC a:PH 15,CU d! In MML, DGN:DFC=a:DATA,PH=15,CU=d;

Table A-4. Supplementary Phases for SCSI DFC

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DFC	14	DFC Phase 14 executes the bit-wise independence test for the First-In First-Out (FIFO) buffer memory of the Bus Interface Controller (BIC).	DEX, Demand	In PDS, DGN:DFC a:PH 14! In MML, DGN:DFC=a:DATA,PH=14;
DFC	15	DFC Phase 15 tests data communication between the bus interface controller (in the DFC) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU used as a helper unit must be OOS when the diagnostic is run, and a manual remove of the CU must precede the demand phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:DFC a:PH 15,CU d! In MML, DGN:DFC=a:DATA,PH=15,CU=d;
DFC	90	DFC Phase 90 tests the SCSI bus protocol by looping around data between the two buses. Note: Manual intervention is required to disconnect the existing cables from the DFC and to connect the looparound cable between the two buses where they connect to the DFC.	Demand	In PDS, DGN:DFC a:PH 90! In MML, DGN:DFC=a:DATA,PH=90;

Table A-5. Supplementary Phases for IOP

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
IOP	15	IOP Phase 15 tests data communication between the bus interface controller (in the IOP) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU must be OOS when the diagnostic is run, and a manual remove of the CU must precede the demand phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:IOP a:PH 15,CU d! In MML, DGN:IOP=a:DATA,PH=15,CU=d;
MTC	5	MTC Phase 5 verifies the operation of an MT transport. The MT connected to the MTC under test must be specified as the helper unit. A formatted diagnostic test tape with a Write-Enable Ring (WER) must be mounted on the MT, and the MT must be in the ON-LINE mode. A formatted diagnostic test tape can be obtained from Lucent Technologies, J-1P059AB-1 List 1M1. Any 200-foot reel of good quality tape certified at 6250 bits per inch can be used as a diagnostic test tape after it has been formatted at load point [Bottom of Tape (BOT)] on an In-Service (INS) MT. Tape reels longer than 200 feet will cause Conditional All Tests Pass (CATP) or Some Tests Failed (STF) results in MTC Phase 5.	Demand	In PDS, DGN:MTC a:PH 5,MT d! In MML, DGN:MTC=a:DATA,MT=d,PH=5;

Table A-5. Supplementary Phases for IOP (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
MTC	5	With prolonged use, the diagnostic test tape can become worn and cause STF results in MTC Phase 5. Sometimes moving the load point (BOT) a few feet farther into the tape reel and formatting the tape at the new load point will temporarily correct STF results caused by worn tape. Refer to Figure A-1 for BOT and End of Tape (EOT) marker placement on the back (shiny) side of the tape. To format a diagnostic test tape, use the following procedure: 1. Mount a 200-foot tape with a WER on an In-Service (INS) MT. 2. Press the LOAD button. This should move the tape to the load point (BOT). 3. Press the ON-LINE button. The ON-LINE light should go on. 4. Enter the recommended command ("d" is the MT unit number).	Demand	In PDS, COPY:TAPE:TEST, TD "/dev/mtd8" In MML, COPY:TAPE:DATA,TEST, TD="/dev/mtd8";
DUIC	5	DUIC Phase 5 tests the interface to Direct User Interface (DUI) using the loopback mode. This phase can only be performed if DUIs are equipped.	Demand	In PDS, DGN:DUIC a:PH 5! In MML, DGN:DUIC=a:DATA,PH=5;
DUIC	6	DUIC Phase 6 tests the send-frames mode to DUI from the controller. This phase can only be performed if DUIs are equipped.	Demand	In PDS, DGN:DUIC a:PH 6! In MML, DGN:DUIC=a:DATA,PH=6;

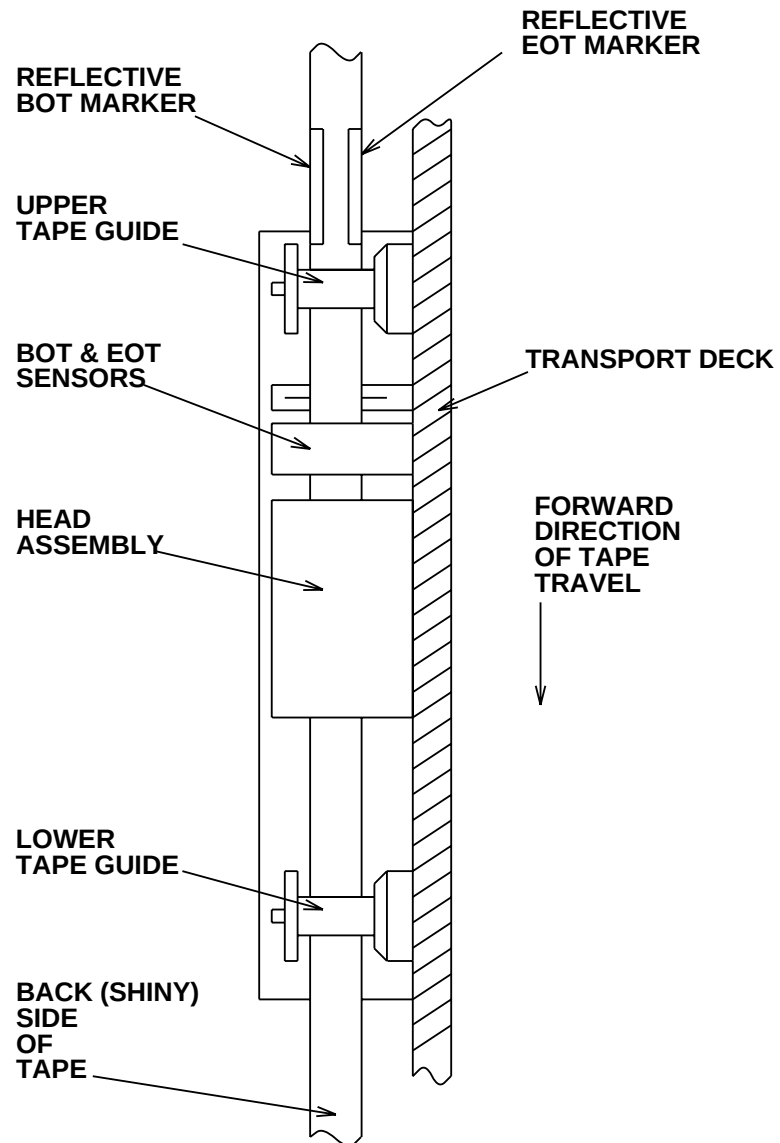


Figure A-1. BOT and EOT Marker Placement

Main Store Failure Data

Failure data for faults detected in main store may output in the form of a histogram. (Refer to Figure A-2 for an example.) A data pattern is 40 bits wide: 32 data bits, 4 hamming bits, and 4 parity bits. The address spectrum is divided among several memory arrays (circuit packs) as shown in Table A-6. When the data pattern read back from the store does not match the one that was written, a failure has occurred. Memory testing stops after the first 100 faults are detected.

A detailed failure analysis is output for the first fault. This analysis includes the failing address, expected value, received value, mismatch for both the data and parity bits, and a dump of three main store controller registers.

The number of addresses that failed and a list of the first eight failing addresses are followed by the memory failure histogram which shows the following:

- For each data bit, hamming bit, and parity bit, how many times a bit failed when a zero was expected and how many times a bit failed when a one was expected.
- For each address bit, how many times the address bit was a zero and how many times it was a one for each of the data bit, hamming bit, and parity bit failures.
- How many multibit errors were detected.
- How many logic errors were detected (single-bit error indication on a multibit error, or other error detection failures).

The numbers in these columns can be used to determine the location and extent of the problem. For example, in the DATA FAIL column, a 1 would indicate a single-memory cell failure, but a 100 would indicate a fault on a data signal path or a bad memory part. In the ADDRESS FAIL column, if the counts for a particular bit are approximately equal, the address bit is probably good. But, if on all failures a particular address bit always has the same value, then the address bit is stuck at that value.

⇒ NOTE:

The upper address bits select the memory array. They usually appear as stuck bits because the effect of a fault is usually isolated to one memory array circuit pack.

In the example shown in Figure A-2, data bit 24 is stuck at 0 but only when address bit 2 is 0. The address bit is the array half select. Therefore, since the main store under test contains TN14s, the fault is that the data bit 24 signal path is stuck at 0 for the A-half of memory array 2.

For TN14 memory arrays:

- Address bit 23 selects the MASC (0 or 1).
- Address bits 22-19 select the array (0-15).
- Address bit 2 selects the memory part (1 of 2).
- Address bits 24 and 25 are not used.

For TN28 memory arrays:

- Address bits 23-20 select the array (0-15).
- Address bits 19 and 2 select the memory part (1 of 4).
- Address bits 24 and 25 are not used.

For TN56 memory arrays:

- Address bits 24-21 select the array (0-15).
- Address bit 20 selects the memory part (1 of 2).
- Address bit 25 is not used.

For TN2012 memory arrays:

- Address bits 25-22 select the array (0-15).

DGN CU 1 MASC 0 PH 35 MEMORY FAILURE RESULTS
FIRST FAULT DATA

FAILING ARRAY	= TN14 at EQL 056-048
FAILING ADDRESS	= X'00100000
DATA MISMATCH	= X'01000000
ACTUAL DATA	= X'54555555
EXPECTED DATA	= X'55555555
PARITY MISMATCH	= X'00000000
ACTUAL PARITY	= X'00000000
EXPECTED PARITY	= X'00000000
MAS ERROR REG 2	= X'00003270
TRAPPED ADDRESS REG	= X'1B900001
MAS ERROR REG 1	= X'00000000
CC ERROR REG	= X'0FFFFFFF

NUMBER OF FAILING ADDRESSES = 100

FIRST 8 FAILING ADDRESS:

X'100000
X'100008
X'100010
X'100018
X'100020
X'100028
X'100030
X'100038

Figure A-2. Sample Memory Failure Histogram (Sheet 1 of 2)

MEMORY FAILURE HISTOGRAM				
DATA/ADR BITS	DATA FAIL EXP=1 RTN=0	COUNT EXP=0 RTN=1	ADDRESS FAIL AS 0	COUNT AS 1
00	0	0	0	0
01	0	0	0	0
02	0	0	100	0
03	0	0	50	50
04	0	0	50	50
05	0	0	52	48
06	0	0	52	48
07	0	0	52	48
08	0	0	64	36
09	0	0	64	36
10	0	0	100	0
11	0	0	100	0
12	0	0	100	0
13	0	0	100	0
14	0	0	100	0
15	0	0	100	0
16	0	0	100	0
17	0	0	100	0
18	0	0	100	0
19	0	0	100	0
20	0	0	0	100
21	0	0	100	0
22	0	0	100	0
23	0	0	100	0
24	100	0	100	0
25	0	0	0	100
26	0	0		
27	0	0		
28	0	0		
29	0	0		
30	0	0		
31	0	0		
H0	0	0		
H1	0	0		
H2	0	0		
H3	0	0		
PX	0	0		
PY	0	0		
PZ	0	0		
PA	0	0		
MULTI	0			
LOGIC	0			

Figure A-2. Sample Memory Failure Histogram (Sheet 2 of 2)

Table A-6. Partitioning of Address Spectrum Among Memory Arrays

MASA Type	MASA	Controller 0 Address Range in Hex	Controller 1 Address Range in Hex
TN14	00	000000 – 07ffffc	800000 – 87ffffc
TN14	01	080000 – 0fffffc	880000 – 8fffffc
TN14	02	100000 – 17ffffc	900000 – 97ffffc
TN14	03	180000 – 1fffffc	980000 – 9fffffc
TN14	04	200000 – 27ffffc	a00000 – a7ffffc
TN14	05	280000 – 2fffffc	a80000 – afffffc
TN14	06	300000 – 37ffffc	b00000 – b7ffffc
TN14	07	380000 – 3fffffc	b80000 – bfffffc
TN14	08	400000 – 47ffffc	c00000 – c7ffffc
TN14	09	480000 – 4fffffc	c80000 – cfffffc
TN14	10	500000 – 57ffffc	d00000 – d7ffffc
TN14	11	580000 – 5fffffc	d80000 – dfffffc
TN14	12	600000 – 67ffffc	e00000 – e7ffffc
TN14	13	680000 – 6fffffc	e80000 – efffffc
TN14	14	700000 – 77ffffc	f00000 – f7ffffc
TN14	15	780000 – 7fffffc	f80000 – ffffffc
TN28	00	000000 – 0fffffc	Not equipped
TN28	01	100000 – 1fffffc	Not equipped
TN28	02	200000 – 2fffffc	Not equipped
TN28	03	300000 – 3fffffc	Not equipped
TN28	04	400000 – 4fffffc	Not equipped
TN28	05	500000 – 5fffffc	Not equipped
TN28	06	600000 – 6fffffc	Not equipped
TN28	07	700000 – 7fffffc	Not equipped
TN28	08	800000 – 8fffffc	Not equipped
TN28	09	900000 – 9fffffc	Not equipped
TN28	10	a00000 – afffffc	Not equipped
TN28	11	b00000 – bfffffc	Not equipped
TN28	12	c00000 – cfffffc	Not equipped
TN28	13	d00000 – dfffffc	Not equipped
TN28	14	e00000 – efffffc	Not equipped
TN28	15	f00000 – ffffffc	Not equipped

Table A-6. Partitioning of Address Spectrum Among Memory Arrays (Contd)

MASA Type	MASA	Controller 0 Address Range in Hex	Controller 1 Address Range in Hex
TN56	00	000000 – 1fffffc	Not equipped
TN56	01	200000 – 3fffffc	Not equipped
TN56	02	400000 – 5fffffc	Not equipped
TN56	03	600000 – 7fffffc	Not equipped
TN56	04	800000 – 9fffffc	Not equipped
TN56	05	a00000 – bfffffc	Not equipped
TN56	06	c00000 – dfffffc	Not equipped
TN56	07	e00000 – ffffffc	Not equipped
TN56	08*	1000000 – 11fffffc	Not equipped
TN56	09*	1200000 – 13fffffc	Not equipped
TN56	10*	1400000 – 15fffffc	Not equipped
TN56	11*	1600000 – 17fffffc	Not equipped
TN56	12*	1800000 – 19fffffc	Not equipped
TN56	13*	1a00000 – 1bfffffc	Not equipped
TN56	14*	1c00000 – 1dfffffc	Not equipped
TN56	15*	1e00000 – 1fffffc	Not equipped
TN2012	00	000000 – 3fffffc	Not equipped
TN2012	01	400000 – 7fffffc	Not equipped
TN2012	02	800000 – bfffffc	Not equipped
TN2012	03	c00000 – ffffffc	Not equipped
TN2012	04*	1000000 – 13fffffc	Not equipped
TN2012	05*	1400000 – 17fffffc	Not equipped
TN2012	06*	1800000 – 1bfffffc	Not equipped
TN2012	07*	1c00000 – 1fffffc	Not equipped
TN2012	08†	2000000 – 23fffffc	Not equipped
TN2012	09†	2400000 – 27fffffc	Not equipped
TN2012	10†	2800000 – 2bfffffc	Not equipped
TN2012	11†	2c00000 – 2fffffc	Not equipped
TN2012	12†	3000000 – 33fffffc	Not equipped
TN2012	13†	3400000 – 37fffffc	Not equipped
TN2012	14†	3800000 – 3bfffffc	Not equipped
TN2012	15†	3c00000 – 3fffffc	Not equipped

* Requires Extended Main Memory (EMM) feature, or Very Large Main Memory (VLMM) feature.

† Requires VLMM feature.

Processor Recovery Messages

This section covers Processor Recovery Messages (PRMs) from Control Unit (CU) CC Phases 61 and 93. The CU under test produces PRMs while executing CC Phases 61 and 93. If CC Phases 61 and 93 execute all their tests (no Test Skipped Reason bits set in phase results) and the expected PRMs are not displayed, there may be a hardware problem with the EAI under test or in its connections to the MTTY Peripheral Controllers (PCs).

The first one or possibly two PRMs from CC Phases 61 and 93 are residual PRMs flushed from the EAI. They can be disregarded if they have a D or F in position 15:

```
PRM_z xxxx xxxx xxxD xx xx xx
PRM_z xxxx xxxx xxxF xx xx xx
```

CC Phase 61 expected PRM pattern:

```
PRM_z 0123 4567 89AB CDEF xx xx xx
PRM_z 0000 0000 0000 0000 xx xx xx
PRM_z FFFF FFFF FFFF FFFF xx xx xx
```

CC Phase 93 expected PRM pattern:

```
PRM_z 0123 4567 89AB CDEF xx xx xx
PRM_z AAAA AAAA AAAA AAAA xx xx xx
PRM_z 5555 5555 5555 5555 xx xx xx
PRM_z EEEE EEEE EEEE EEEE xx xx xx
PRM_z 1111 1111 1111 1111 xx xx xx
PRM_z 0000 0000 0000 0000 xx xx xx
PRM_z FFFF FFFF FFFF FFFF xx xx xx
```

Where:

x = any hexadecimal number.

z = number of the CU under test (0 or 1).

Refer to 254-303-103, *Processor Recovery Messages Guide*, for more information about PRMs.

Diagnostic Information Tables — CU, DFC, and IOP

The following are the diagnostic information tables found in this appendix.

Table A-7	Application Channel Interface
Table A-8	Central Control
Table A-9	Cache Storage Unit
Table A-10	Enlarged Cache Storage Unit
Table A-11	Disk File Controller
Table A-12	SCSI DFC
Table A-13	Direct Memory Access
Table A-14	Direct Memory Access Channel
Table A-15	Dual Serial Channel
Table A-16	Direct User Interface Controller
Table A-17	Input/Output Processor
Table A-18	Main Store Controller
Table A-19	Default Refresh Data Parity Check Times for MASC Phase 96
Table A-20	Moving Head Disk
Table A-21	Magnetic Tape Controller
Table A-22	Maintenance Teletypewriter Controller
Table A-23	Store Address Translator
Table A-24	Serial Channel
Table A-25	Scan Point and Signal Distributor Controller
Table A-26	Synchronous Data Link Controller
Table A-27	Teletypewriter Controller
Table A-28	Utility Circuit

Table A-7. Application Channel Interface

Phase	Description
achi01	Application Channel Interface (ACHI) — CCIO bus interface.
achi02	Data register looparound tests.
achi03	Miscellaneous function tests.

Table A-8. Central Control

Phase	Description
cc01	On-line MCH master tests.
cc02	MCH slave communication tests.
cc03	MCH slave command tests.
cc04	MCH microstore address and data register tests.
cc05	Microcontrol (MC) clock circuit tests, microinstruction register tests, and MC parity tests.
cc06	MC immediate data path, MCH access to test multiplexer, buffered bidirectional gating register parity bits, and path to central control error register.
cc07	MC address sequencer: verifies basic address path for unconditional branch and no branch.
cc08	MC address sequencer: conditional jump logic, opcode jam path, MC address stack, INT command to MCH, set interrupt set, and interrupt mask registers.
cc09	Address and memory verification of UN28 [Read-Only Memory (ROM)] in CU frame slot 0. Requires equipage bits in Equipment Configuration Database (ECD).
cc10	Microstore address bus parity tests.
cc11	Status registers — hardware status register, program status word, system status register.
cc12	Pulse point register.
cc13	Gate through data manipulation unit and byte rotate.
cc14	Data manipulation unit bit rotate.
cc15	Arithmetic logic unit matcher.
cc16	Data manipulation unit Q-register.
cc17	Data manipulation unit XOR, Z-flag, and ONES.
cc18	Data manipulation unit general registers.
cc19	Arithmetic logic unit and gating paths.

Table A-8. Central Control (Contd)

Phase	Description
cc20	24-bit mode.
cc21	Data manipulation unit looparound.
cc22	Temporary registers and firmware registers.
cc26	MC privileged Programmable Read-Only Memory (PROM) tests.
cc27	MC privileged PROM tests.
cc28	MC privileged PROM tests.
cc30	Writable microstore: bidirectional gating bus, microstore data bus, and microstore address bus access. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 1. Requires equipage bits in ECD.
cc31	Writable microstore control logic. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 1. Requires equipage bits in ECD.
cc32	Routine Exercise (REX) phase. Address and memory test of UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 1. Requires equipage bits in ECD.
cc33	Writable microstore: bidirectional gating bus, microstore data bus, and microstore address bus access. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 2. Requires equipage bits in ECD.
cc34	Writable microstore control logic. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 2. Requires equipage bits in ECD.
cc35	Routine Exercise (REX) phase. Address and memory test of UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 2. Requires equipage bits in ECD.
cc36	Writable microstore: bidirectional gating bus, microstore data bus, and microstore address bus access. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 3. Requires equipage bits in ECD.

Table A-8. Central Control (Contd)

Phase	Description
cc37	Writable microstore control logic. Tests UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 3. Requires equipage bits in ECD.
cc38	Routine Exercise (REX) phase. Address and memory test of UN48 (4 K writable microstore) or UN248 (16 K writable microstore) in CU frame slot 3. Requires equipage bits in ECD.
cc39	Data manipulation unit parity.
cc40	Store control register/store control field of microinstruction register to store control register.
cc41	Store data register, store instruction register, instruction buffer.
cc42	Store address register and program address register.
cc43	Instruction buffer parity circuit.
cc44	Address increment and parity predict.
cc45	Address increment (store address register only).
cc46	Store instruction register path to half-word multiplexer to instruction buffer.
cc47	Store instruction register path to half-word multiplexer to instruction buffer.
cc48	Store instruction register path to half-word multiplexer to instruction buffer.
cc49	Store instruction register path to half-word multiplexer to instruction buffer.
cc52	A and B select and rotate amount multiplexer.
cc53	Mask ROM.
cc54	Find-low-zero.
cc55	Interrupt registers and logic.
cc56	Timers (includes CL_PT MCH command check).
cc57	Channel data register.
cc58	Central Control Input/Output (CCIO) bus sanity.
cc59	Demand-only phase. Other unit interrupts.

Table A-8. Central Control (Contd)

Phase	Description
cc60	Error response logic.
cc61	EAI tests. Refer to the "Processor Recovery Messages" section in Chapter 6 for expected Processor Recovery Messages (PRMs).
cc62	Off-line MCH master side tests.
cc70	Address and memory verification of UN28 (ROM) in CU frame slot 2. Requires equipage bits in ECD.
cc71	Address and memory verification of UN28 (ROM) in CU frame slot 3. Requires equipage bits in ECD.
cc93	Demand exercise phase. EAI command and display tests. Requires manual observation. Refer to the "Processor Recovery Messages" section in Chapter 6 for expected PRMs.

Table A-9. Cache Storage Unit

Phase	Description
csu01	Access to cache memories and kernel/interrupt stack enable tests.
csu02	Cache memory match tests.
csu03	Hit/miss matchers (ATBA to tags A and B).
csu04	Hit/miss matchers (ATBB to tags A and B).
csu05	Hit/miss matchers (ATBA to tags C and D).
csu06	Hit/miss matchers (ATBB to tags C and D).
csu07	Select enable logic.
csu08	Write tag control logic (tags A and B).
csu09	Write tag control logic (tags C and D).
csu10	Store Address Translator (SAT) interface (ATBA to tags A and C).
csu11	SAT interface (ATBB to tags A and C).
csu12	Tag parity generators.
csu13	Tag parity checkers.
csu14	Hit and miss counters.
csu15	Cache store write control logic.
csu16	Cache store write control logic.
csu17	Error check circuits on cache memory board.
csu18	Routine Exercise (REX) phase. Cache exercise (40 seconds).
csu90	Demand-only phase. Cache exercise (40 minutes).

Table A-10. Enlarged Cache Storage Unit

Phase	VLMM Phase	Description
csue01	csux01	Access to cache memories and interrupt stack enable tests.
csue02	csux02	Cache tag and cache store memory match tests.
csue03	csux03	Hit/miss matchers (ATBA to tags A and B).
csue04	csux04	Hit/miss matchers (ATBB to tags A and B).
csue05	csux05	Hit/miss matchers (ATBA to tags C and D).
csue06	csux06	Hit/miss matchers (ATBB to tags C and D).
csue07	csux07	Hit/miss matchers (ATBA to tags E and F).
csue08	csux08	Hit/miss matchers (ATBA to tags E and F).
csue09	csux09	Hit/miss matchers (ATBA to tags G and H).
csue10	csux10	Hit/miss matchers (ATBA to tags G and H).
csue11	csux11	Select enable logic.
csue12	csux12	Cache tag memory write control logic (tags A and B).
csue13	csux13	Cache tag memory write control logic (tags C and D).
csue14	csux14	Cache tag memory write control logic (tags E and F).
csue15	csux15	Cache tag memory write control logic (tags G and H).
csue16	csux16	ATBA relocation address to tag memory mod A and E.
csue17	csux17	ATBA relocation address to tag memory mod B and F.
csue18	csux18	ATBA relocation address to tag memory mod C and G.
csue19	csux19	ATBA relocation address to tag memory mod D and H.
csue20	csux20	Address and data parity generators.
csue21	csux21	Address and data parity checkers (tags A, B, C, and D).
csue22	csux22	Address and data parity checkers (tags E, F, G, and H).
csue23	csux23	Cache store write control logic – stack write and write update operations (all tags: A, B, C, D, E, F, G, and H).
csue24	csux24	Cache store write control logic - normal read stack read and test cache mods updated at store complete time (all tags: A, B, C, D, E, F, G, and H).
csue25	csux25	Error check circuits on cache memory board.
csue26	csux26	Routine Exercise (REX) phase. Cache exercise (40 seconds).
csue90	csux90	Demand-only phase. Cache exercise (40 minutes).

Table A-11. Disk File Controller

Phase	Description
dfc01	Dual Serial Channel (DSCH) — Duplex Dual Serial Bus Selector (DDSBS) interface.
dfc02	DDSBS — BIC interface and BIC circuitry excluding Peripheral Interface Controller (PIC) interface.
dfc03	BIC — PIC interface using PIC resident firmware.
dfc04	BIC — PIC communication.
dfc05	PIC microstore data validity and microstore parity checker.
dfc06	PIC microsequencer.
dfc07	PIC register and ALU.
dfc08	PIC source/destination decoders.
dfc09	PIC internal registers.
dfc10	PIC interrupt circuitry.
dfc11	PIC Random Access Memory (RAM) memory and sequencer.
dfc12	Parallel/serial data interface and MHD control source and destination decoders.
dfc13	MHD control register.
dfc15	Demand-only phase. Tests the interface with the nonactive CU. The nonactive CU must be specified as the helper unit and must be OOS and ATP.

Table A-12. SCSI DFC

Phase	Description
dfcs01	Dual Serial Channel (DSCH) — Duplex Dual Serial Bus Selector (DDSBS) interface.
dfcs02	DDSBS — BIC interface and BIC circuitry excluding Host Adapter (HA) control interface. No BIC FIFO buffer memory testing.
dfcs03	BIC — HA control interface using HA resident firmware.
dfcs04	WE32100 Central Processing Unit (CPU) test.
dfcs05	Test ROM by verifying its content.
dfcs06	Test 1 MB Static Random Access Memory (SRAM).
dfcs07	HA register tests.
dfcs08	WE32104 Direct Memory Access Controller (DMAC) test.
dfcs09	Dual Universal Asynchronous Receiver/Transmitter (DUART) test.
dfcs10	System Timing Controller (STC) test.
dfcs11	SCSI Protocol Controllers (SPCs) test. Tests both SCSI buses.
dfcs12	Test (validate) only differential type devices attached to buses.
dfcs13	Test ability of SCSI devices to recognize and acknowledge bus reset.
dfcs14	Demand Exercise (DEX) phase. Bit-wise independence test for HA FIFO buffer memory.
dfcs15	Demand-only phase. Tests the interface with the nonactive CU. The nonactive CU must be specified as the helper unit and must be OOS and ATP.
dfcs90	Demand-only phase. SCSI bus loop test. (Requires craft interface and cable manipulation.)

Table A-13. Direct Memory Access

Model 1 Phase	Models 2 & 3 Phase	VLMM Phase	Description
dma01	dma01	dmav01	DMA — central control input/output bus interface.
dma02	dma02	dmav02	Status tests.
dma03	dma03	dmav03	RAM/register tests.
dma04	dma04	dmav04	Sequencer tests.
dma05	dma05	dmav05	Request priority circuit.
dma06	dma06	dmav06	Store address tests.
dma07	dma07	dmav07	Store data tests.
dma08	dma08	dmav08	DMA error tests.
dma09	—	—	DMA/cache interface tests.
—	dma09	dmav09	DMA/cache interface tests, Cache Memory Controller 0 (CAC0).
—	dma10	dmav10	DMA/cache interface tests, Cache Memory Controller 1 (CAC1).
dma10	dma11	dmav11	Main Store Update (MASU) priority and arbitration function tests.

Table A-14. Direct Memory Access Channel

Model 1 Phase	Models 2 & 3 Phase	Description
dmch01	dmch01	DSCH — direct input/output bus interface.
dmch02	dmch02	Data register (FIFO) tests.
dmch03	dmch03	DSCH error tests.
dmch08	dmch08	Port 0 selection, interrupts, and service requests.
dmch09	dmch09	Port 1 selection, interrupts, and service requests.
dmch10	dmch10	Port 2 selection, interrupts, and service requests.
dmch11	dmch11	Port 3 selection, interrupts, and service requests.
dmch12	dmch12	Port 4 selection, interrupts, and service requests.
dmch13	dmch13	Port 5 selection, interrupts, and service requests.
dmch14	dmch14	Port 6 selection, interrupts, and service requests.
dmch15	dmch15	Port 7 selection, interrupts, and service requests.
dmch16	dmch16	Port 8 selection, interrupts, and service requests.
dmch17	dmch17	Port 9 selection, interrupts, and service requests.
dmch18	dmch18	Port 10 selection, interrupts, and service requests.
dmch19	dmch19	Port 11 selection, interrupts, and service requests.
dmch20	dmch20	Port 12 selection, interrupts, and service requests.
dmch21	dmch21	Port 13 selection, interrupts, and service requests.
dmch22	dmch22	Port 14 selection, interrupts, and service requests.
dmch23	dmch23	Port 15 selection, interrupts, and service requests.

Table A-14. Direct Memory Access Channel (Contd)

Model 1 Phase	Models 2 & 3 Phase	Description
dmch24	dmch24	Port 0 DSCH — DBS communications tests.
dmch25	dmch25	Port 1 DSCH — DBS communications tests.
dmch26	dmch26	Port 2 DSCH — DBS communications tests.
dmch27	dmch27	Port 3 DSCH — DBS communications tests.
dmch28	dmch28	Port 4 DSCH — DBS communications tests.
dmch29	dmch29	Port 5 DSCH — DBS communications tests.
dmch30	dmch30	Port 6 DSCH — DBS communications tests.
dmch31	dmch31	Port 7 DSCH — DBS communications tests.
dmch32	dmch32	Port 8 DSCH — DBS communications tests.
dmch33	dmch33	Port 9 DSCH — DBS communications tests.
dmch34	dmch34	Port 10 DSCH — DBS communications tests.
dmch35	dmch35	Port 11 DSCH — DBS communications tests.
dmch36	dmch36	Port 12 DSCH — DBS communications tests.
dmch37	dmch37	Port 13 DSCH — DBS communications tests.
dmch38	dmch38	Port 14 DSCH — DBS communications tests.
dmch39	dmch39	Port 15 DSCH — DBS communications tests.
dmch40	dmch40	Demand-only phase. BIC — main store data transfer tests. Requires an out-of-service DFC or IOP as a helper unit.
dmch44	dmch44	Demand-only phase. Bootstrap exercise test. Requires an out-of-service DFC 0 or 1 as a helper unit.

Table A-15. Dual Serial Channel

Phase	Description
dsch01	DSCH — central control input/output bus interface.
dsch02	Data register (FIFO) tests.
dsch03	DSCH error tests.
dsch08	Port 0 selection, interrupts, and service requests.
dsch09	Port 1 selection, interrupts, and service requests.
dsch10	Port 2 selection, interrupts, and service requests.
dsch11	Port 3 selection, interrupts, and service requests.
dsch12	Port 4 selection, interrupts, and service requests.
dsch13	Port 5 selection, interrupts, and service requests.
dsch14	Port 6 selection, interrupts, and service requests.
dsch15	Port 7 selection, interrupts, and service requests.
dsch16	Port 8 selection, interrupts, and service requests.
dsch17	Port 9 selection, interrupts, and service requests.
dsch18	Port 10 selection, interrupts, and service requests.
dsch19	Port 11 selection, interrupts, and service requests.
dsch20	Port 12 selection, interrupts, and service requests.
dsch21	Port 13 selection, interrupts, and service requests.
dsch22	Port 14 selection, interrupts, and service requests.
dsch23	Port 15 selection, interrupts, and service requests.

Table A-16. Direct User Interface Controller

Phase	Description
duic01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
duic02	Read/write access of dual access memory.
duic03	Dual access memory parity checker.
duic04	Microdiagnostic.
duic05	Demand-only phase. Loopback mode. Special procedures.
duic06	Demand-only phase. Send-frames mode. Special procedures.

Table A-17. Input/Output Processor

Phase	Description
iop01	DSCH — DDSBS interface.
iop02	DDSBS — BIC interface and BIC circuitry excluding PIC interface.
iop03	BIC — PIC interface using PIC resident firmware.
iop04	BIC — PIC communication.
iop05	PIC microstore data validity and microstore parity checker.
iop06	PIC microsequencer.
iop07	PIC register and ALU.
iop08	PIC source/destination decoders.
iop09	PIC internal registers.
iop10	PIC interrupt circuitry.
iop11	PIC RAM memory and sequencer.
iop12	IO microprocessor interface hardware.
iop13	IO microprocessor interface power control circuitry peripheral controller community power converter/monitor.
iop15	Demand-only phase. Test the interface with the nonactive CU. The nonactive CU must be specified as the helper unit and must be OOS and ATP.

Table A-18. Main Store Controller

Phase	Description
masc01	Test the inactive state of errors B/C and basic MASC response.
masc02	Address looparound.
masc04	Data communication tests.
masc05	Parity circuit (data, address, control) tests and half-word and byte decoding.
masc06	Maintenance commands, my store time-out microinterrupt, and store go error tests.
masc07	MASC memory array response test.
masc08	Refresh rate tests.
masc09	MASC timing chain tests.
masc10	Special memory operations and array parity tests (main store array 0).
masc11	Destination and source wait states.
masc12	Single-bit hamming correction tests.
masc13	Double-bit error (hamming) tests.
masc14-28	Same as Phase 10 for memory arrays 1 – 15.
masc29	Address looparound through main store update (duplex mode).
masc30	Data communication tests through main store update (duplex mode).
masc31	Control lead parity and other store error tests through main store update (duplex mode).
masc32	My store error B/C tests.
masc33	Demand Exercise (DEX) phase. Memory data pattern tests.
masc34	Demand Exercise (DEX) phase. Memory data pattern tests.
masc35	Memory data pattern tests.
masc36	Demand Exercise (DEX) phase. Memory data pattern tests.
masc37	Test the capability to trap a failing address on errors A, C, and D.

Table A-18. Main Store Controller (Contd)

Phase	Description
masc38	Ensure the CC and MASC can fetch and execute instructions.
masc39	Update mode and refresh data parity check functional tests — part of memory.
masc40	Demand Exercise (DEX) phase. Memory data pattern tests, half-word read and clear.
masc41	Demand-only phase. Update mode and refresh data parity check tests — entire memory.
masc42	MASU priority circuit and bus arbitration tests.
masc93	Routine Exercise (REX) phase. Test the main and duplicate refresh address registers.
masc95	Demand-only phase. Memory data pattern tests with optional manual input parameters.
masc96	Routine Exercise (REX) phase. Update mode and refresh data parity check tests — entire memory — with optional manual input parameters.

Table A-19. Default Refresh Data Parity Check Times for MASC Phase 96

Memory Array	Standard	EMM Feature	VLMM Feature
TN14	34 sec	—	—
TN28	68 sec	—	—
TN56	68 sec	136 sec	136 sec
TN2012	68 sec	136 sec	272 sec

Table A-20. Moving Head Disk

SCSI Phase	SMD Phase	Description
mhds01*		Diagnoses the SCSI device identified in the command: - Runs device self-tests. - Verifies integrity of SCSI device controller buffer. - Verifies reads/writes to MHD maintenance area.
	mhd01	Parallel/serial data interface and MHD control source and destination decoders.
	mdh02	MHD control register.
	mhd03	Parallel/serial data interface.
	mhd04	MHD control, disk select, and driver enable.
	mhd05*	MHD clock.
	mhd06*	Read/write circuitry and error correction circuitry.
	mhd07*	Seek, seek error detection, servo offset, and fault detection circuitry of the drive.
	mhd08*	Data transfer between CU and the drive.

* Indicates that a fully formatted disk pack must be mounted and spinning on the drive under test.

Table A-21. Magnetic Tape Controller

Phase	Description
mtc01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
mtc02	Read/write access of dual access memory.
mtc03	Dual access memory parity checker.
mtc04	Microdiagnostic.
mtc05	Demand-only phase. Tape transport. A formatted diagnostic test tape must be mounted on the tape transport under test.

Table A-22. Maintenance Teletypewriter Controller

Phase	Description
mttyc01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
mttyc02	Read/write access of dual access memory.
mttyc03	Dual access memory parity checker.
mttyc04	Microdiagnostic.

Table A-23. Store Address Translator

Phase	VLMM Phase	Description
sat01	satv01	Store control register control flip-flops.
sat02	satv02	ATB access and parity.
sat03	satv03	ATB memory.
sat04	satv04	Counter and invalidate circuit.
sat05	satv05	Address multiplexer.
sat06	satv06	Write control circuit.
sat07	satv07	ATB hit and ATB miss logic.
sat08	satv08	Protection circuit.
sat09	satv09	Matchers.
sat10	satv10	Main store interface.

Table A-24. Serial Channel

Phase	Description
sch01	Serial channel — central control input/output bus interface.
sch02	Data register write and read tests.
sch03	Functional, maintenance, and error tests.
sch04	Device port selection tests.

Table A-25. Scan Point and Signal Distributor Controller

Phase	Description
scsdc01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
scsdc02	Read/write access of dual access memory.
scsdc03	Dual access memory parity checker.
scsdc04	Microdiagnostic.

Table A-26. Synchronous Data Link Controller

Phase	Description
sdlc01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
sdlc02	Read/write access of dual access memory.
sdlc03	Dual access memory parity checker.
sdlc04	Microdiagnostic.

Table A-27. Teletypewriter Controller

Phase	Description
ttyc01	Request isolate, control signal, and control signal acknowledge from the peripheral interface controller.
ttyc02	Read/write access of dual access memory.
ttyc03	Dual access memory parity checker.
ttyc04	Microdiagnostic.

Table A-28. Utility Circuit

Phase	VLMM Phase	Description
uc01	ucv01	Bidirectional gating bus access and enable.
uc02	ucv02	Access matcher, address leads, read/write test.
uc03	ucv03	Block address matcher, address leads, read/write test.
uc04	ucv04	Address matcher, address leads, read/write test.
uc05	ucv05	Unit identification matcher, address leads, read/write test.
uc06	ucv06	Data matcher, address leads, read/write test.
uc07	ucv07	Event and transfer trace counters, read/write test.
uc08	ucv08	Transfer trace access path unit identification buffers test.
uc09	ucv09	Trace memory, read/write test.
uc10	ucv10	Trigger functions in freeze mode tests.
uc11	ucv11	Block address matcher adder tests.
uc12	ucv12	Trigger functions in run mode without match conditions.
uc13	ucv13	Access matcher, run mode tests.
uc14	ucv14	Data matcher, run mode tests.
uc15	ucv15	Address matcher, virtual address bus (store address register), run mode tests.
uc16	ucv16	Address matcher, physical address bus (ATBA), run mode tests.
uc17	ucv17	Address matcher, physical address bus (ATBA), run mode tests.

Table A-28. Utility Circuit (Contd)

Phase	VLMM Phase	Description
uc18	ucv18	Unit identification matcher, run mode tests.
uc19	ucv19	Trigger functions in run mode with match conditions.
uc20	ucv20	Transfer trace, run mode tests.
uc90	ucv90	Demand-only phase with special procedure. External input, external output trigger function 0.*
uc91	ucv91	Demand-only phase with special procedure. External input, external output trigger function 1.*
uc92	ucv92	Demand-only phase with special procedure. External input, external output trigger function 2.*
uc93	ucv93	Demand-only phase with special procedure. External input, external output trigger function 3.*

* Refer to the "Supplementary Phases" section for special procedures before doing the test.

3B21D Computer References

B

Contents

Introduction	B-1
Control Unit Diagnostic Test Sequence	B-2
Diagnostic Hierarchy	B-4
■ CU Diagnostic Hierarchy	B-4
■ DFC Hardware Diagnostic Hierarchy	B-4
■ IOP Diagnostic Hierarchy	B-5
Test Procedures	B-6
■ CU Suggested Routine	B-6
■ DFC 2 and DFC 3 Suggested Routine	B-8
■ IOP 2 Suggested Routine	B-10
■ DCI Suggested Routine	B-12
Supplementary Phases	B-13
Main Store Failure Data	B-26
Processor Recovery Messages	B-29
Diagnostic Information Tables — CU, DFC, IOP, and DCI	B-30

3B21D Computer References

B

Introduction

This appendix contains 3B21D computer reference information on the following topics:

- Control Unit (CU) Diagnostic Test Sequence
- Diagnostic Hierarchy
- Test Procedures
- Supplementary Phases
- Main Store Failure Data
- Processor Recovery Messages (PRMs)
- Diagnostic Information Tables — CU, Disk File Controller (DFC), Input/Output Processor (IOP), and Dual Serial Channel Computer Interconnect (DCI).

Control Unit Diagnostic Test Sequence

The sequence of CU diagnostic tests for the 3B21D computer is shown in the following list:

- Maintenance Channel (MCH) communication
- Built-In Self-Test (BIST) boundary scan
- Microcontrol clock
- Gating from/to microstore data bus
- Gating from/to microstore address bus
- Gating from/to bidirectional gating bus
- Gating from maintenance bus
- Initialization
- Microprogram control timing
- Microaddress sequencing
- Immediate data gating to bidirectional gating register
- Source to destination bus gating through the Arithmetic Logic Unit (ALU)/Remote Measurement Unit (RMU)
- Register gating (general registers)
- Register gating (temporary registers)
- Store address register
- Store data register
- Present address register
- Store instruction register
- Instruction register
- Microstore contents
- Store control and Main Store (MAS) contents
- RMU
- ALU
- Interrupt registers
- Timing registers
- Error response
- Emergency Action Interface (EAI)

- Address Translator Buffer (ATB) access logic
- ATB compare and protect logic
- ATB
- Cache access logic
- Cache compare logic
- Cache memory
- Central Control Input/Output (CCIO) bus
- Store controller
- MAS contents
- Store error checker
- Isolate update and other store mode
- Double store read
- Direct Memory Access (DMA)
- Input/Output (IO) channels
- Utility Circuit (UC).

Diagnostic Hierarchy

CU Diagnostic Hierarchy

1. Central Control (CC) includes the Store Address Translator (SAT) and cache diagnostics
2. Main Store Controller (MASC) includes memory arrays
3. Direct Memory Access Controllers (DMACs)
4. Direct Memory Access Channels (DMCHs) and programmed IO channels
5. UC.

DFC Hardware Diagnostic Hierarchy

DFC Hardware

1. DFC
2. Small Computer System Interface (SCSI) Bus (SBUS) (present for SCSI DFCs only)*
3. Moving Head Disk (MHD).

DFC 2 Hardware

1. DFC
2. SBUS*
3. MHD
4. Magnetic Tape Unit (MTU).

DFC 3 Hardware

1. DFC
2. SBUS*
3. MHD
4. MTU.

* SBUS is a restorable unit but does not have a diagnostic.

IOP Diagnostic Hierarchy

IOP Hardware

1. IOP
2. Maintenance Teletypewriter Controller (MTTYC)
3. Teletypewriter Controller (TTYC)
4. Magnetic Tape Controller (MTC)
5. High-Speed Data Link Controller (HSDC)
6. Synchronous Data Link Controller (SDLC)
7. Scan Point and Signal Distributor Controller (SCSDC)
8. Direct User Interface Controller (DUIC).

IOP 2 Hardware

1. IOP
2. MTTYC
3. TTYC
4. HSDC
5. SDLC
6. SCSDC
7. DUIC.

Test Procedures

The following group of procedures is suggested as a complete routine manual diagnostic plan. These suggested test procedures should be performed once a week.

For the procedure to run demand diagnostics on a routine basis for the 3B21D computer, refer to the chapter on performing system demand diagnostics in 254-303-101, *Routine Tasks*.

CU Suggested Routine

The following is the routine test procedure for the 3B21D computer CU. The commands use the following variables *a*, *b*, and *d*:

a = Unit under test

b = Subunit under test

d = Member number of the helper unit.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, DGN:CU a,TLP! In MML, DGN:CU=a:DATA,TLP;
2	Enter: In PDS, DGN:CU a,CC 0:PH 60,TLP! In MML, DGN:CU=a,CC=0:DATA,PH=60,TLP;
3	Enter: In PDS, DGN:CU a,CC 0:PH 6,TLP! In MML, DGN:CU=a,CC=0:DATA,PH=6,TLP;
4	Enter: In PDS, DGN:CU a,CC 0:PH 65,TLP! In MML, DGN:CU=a,CC=0:DATA,PH=65,TLP; (Enter Phase 65 as a separate request because manual intervention is required.)

Step	Procedure
5	Check for the correct sequence of PRM patterns on the Emergency Action Page of the Maintenance Teletypewriter (MTTY) and on the Receive-Only Printer (ROP). (Refer to Table B-2, CC Phase 65.)
6	Check for the correct sequencing of lamps on the EAI display on KLW31. (Refer to Table B-2, CC Phase 65.)
7	Enter: In PDS, RMV:DFC d! In MML, RMV:DFC=d; (An IOP may alternatively be used as the helper unit.)
8	Enter: In PDS, RST:DFC d;UCL! In MML, RST:DFC=d:UCL;
9	Enter: In PDS, RST:CU a;UCL! In MML, RST:CU=a:UCL;
10	Enter: In PDS, SW:CU! In MML, SW:CU;

DFC 2 and DFC 3 Suggested Routine

The following is the routine test procedure for the DFC 2 and DFC 3. The commands use the following variables *a*, *b*, and *d*:

a = Unit under test

b = Subunit under test

d = Member number of the helper unit.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, RMV:CU <i>d</i>! In MML, RMV:CU=<i>d</i>;
2	Enter: In PDS, DGN:DFC <i>a</i>:PH 15,CU <i>d</i>,TLP! In MML, DGN:DFC=<i>a</i>:DATA,PH=15,CU=<i>d</i>,TLP;
3	Enter: In PDS, RST:CU <i>d</i>;UCL! In MML, RST:CU=<i>d</i>:UCL;
4	Enter: In PDS, SW:CU! In MML, SW:CU;
5	Enter: In PDS, RMV:CU <i>d</i>! In MML, RMV:CU=<i>d</i>;

Step	Procedure
6	Enter: In PDS, DGN:DFC a:PH 15,CU d,TLP! In MML, DGN:DFC=a:DATA,PH=15,CU=d,TLP;
7	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d;UCL!
8	Enter: In PDS, RST:DFC a! In MML, RST:DFC=a;
9	Check each MT connected to this DFC by: (a) Inserting a writable tape into the tape drive. (b) Entering: In PDS, DGN:MT a:PH 90,TLP! In MML, DGN:MT=a:DATA,PH=90,TLP; (c) Making the tape created from the previous substep write protected and inserting it into the tape drive. (d) Entering: In PDS, DGN:MT a:PH 91,TLP! In MML, DGN:MT=a:DATA,PH=91,TLP;
10	Enter: In PDS, RST:MT a! In MML, RST:MT=a;

IOP 2 Suggested Routine

The following is the routine test procedure for the IOP 2. The commands use the following variables *a*, *b*, and *d*:

a = Unit under test

b = Subunit under test

d = Member number of the helper unit.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, RMV:CU <i>d</i>! In MML, RMV:CU=<i>d</i>;
2	Enter: In PDS, DGN:IOP <i>a</i>:PH 16,CU <i>d</i>,TLP! In MML, DGN:IOP=<i>a</i>:DATA,PH=16,CU=<i>d</i>,TLP;
3	Enter: In PDS, RST:CU <i>d</i>;UCL! In MML, RST:CU=<i>d</i>:UCL;
4	Enter: In PDS, SW:CU! In MML, SW:CU;
5	Enter: In PDS, RMV:CU <i>d</i>! In MML, RMV:CU=<i>d</i>;
6	Enter: In PDS, DGN:IOP <i>a</i>:PH 16,CU <i>d</i>,TLP! In MML, DGN:IOP=<i>a</i>:DATA,PH=16,CU=<i>d</i>,TLP;

Step	Procedure
7	Enter: In PDS, RST:CU d;UCL! In MML, RST:CU=d:UCL!
8	Enter: In PDS, RST:IOP a! In MML, RST:IOP=a;

DCI Suggested Routine

The following is the routine test procedure for the DCI. The command uses the following variable *a*:

a = Unit under test.

All requests should be issued with the **TLP** option as indicated.

Step	Procedure
1	Enter: In PDS, DGN:DCI a:TLP! In MML, DGN:DCI=a:DATA,TLP;

Supplementary Phases

The following tables cover the supplementary phases:

Table B-1	Recommended Frequency of Selected Supplementary Phases
Table B-2	Supplementary Phases for CU
Table B-3	Supplementary Phases for DFC 2
Table B-4	Supplementary Phases for DFC 3
Table B-5	Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape
Table B-6	Supplementary Phases for IOP 2

Table B-1. Recommended Frequency of Selected Supplementary Phases

Phase	Area Tested	Frequency
CC 6	BIST and boundary scan test.	Weekly
CC 60	Test special registers.	Weekly
CC 65	Emergency Action Interface (EAI) display tests. EAI requires manual observation.	Weekly
MASC 45	Test memory with data patterns Main Store Controller (MASC) and Main Store Array (MASA).	Weekly
MASC 46	Test memory with data patterns MASC and MASA.	Weekly
MASC 50, 51, 52, 53	March pattern test involving writing/reading data patterns.	Weekly
MASC 48	Test memory read and half-word clear MASC and MASA.	Weekly
CC 90	Long-term cache exercise for Cache Storage Unit (CSU). Read and write to cache.	Monthly (20-minute test)
MTC 90	Test tape write/read capability.	Weekly
MTC 91	Test tape write-protect capability.	Weekly

Table B-2. Supplementary Phases for CU

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CC	6	CC Phase 6 tests BIST and boundary scan devices on the CC pack.	Demand	In PDS, DGN:CU a,CC b:PH 6! In MML, DGN:CU=a,CC=b:DATA,PH=6;
CC	34	CC Phase 34 tests the writable microstore. This phase runs for 5 minutes.	REX, DEX, Demand	In PDS, DGN:CU a,CC b:REX! DGN:CU a,CC b:DEX! DGN:CU a,CC b:PH 34! In MML, DGN:CU=a,CC=b:REX; DGN:CU=a,CC=b:DEX; DGN:CU=a,CC=b:DATA,PH=34;
CC	60	CC Phase 60 tests Central Control (CC) interrupt logic circuits connected to units other than the CC. Note: Any units that are causing external error signals should be repaired before CC Phase 60 is run. Do not execute these tests unless all CUs have been diagnosed or the CU has been in service since the last time power was restored to the frame.	Demand	In PDS, DGN:CU a,CC b:PH 60! In MML, DGN:CU=a,CC=b:DATA,PH=60;
CC	65	CC Phase 65 tests the EAI Light-Emitting Displays (LEDs) (located on KLW31) and the ability of the EAI to transmit Processor Recovery Messages (PRMs). A series of PRMs appears on the Emergency Action Page at the MTTY and on the ROP. Then the lamps on the EAI display are cycled through the specific pattern for that phase.	DEX, Demand — run when the user is available to watch the EAI LEDs and display on the circuit pack, MTTY, and ROP.	In PDS, DGN:CU a,CC b:DEX! DGN:CU a,CC b:PH 65! In MML, DGN:CU=a,CC=b:DEX; DGN:CU=a,CC=b:DATA,PH=65;

Table B-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CC	65	All forced states on the EAI must be cleared before CC Phase 65 is run. Do this by using Poke 14 on the Emergency Action Page. Check PRMs displayed for those expected. Refer to the "Processor Recovery Messages" section for further explanation of PRMs. Each LED is lighted and then extinguished in a pattern running down the display. Later, lamp flashes may occur and should be ignored. If you do not observe these patterns, the test failed. A lamp display failure indicates that the EAI (KLW31) is defective. A PRM failure may be due to a defective EAI or a defective MTTYC (TN983) in the IOP to which the affected MTTY or ROP is connected. Note: Since this failure is indicated visually, the phase may run All Tests Pass (ATP) in spite of display faults.		

Table B-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
CC	89	CC Phase 89 exercises the mechanism that writes values from the main store to the enlarged cache. A selection of memory addresses are accessed using various modes. Then the contents of main memory, ATBs, and cache are compared. If no differences are found, the test continues. Differences are reported as errors. If no errors are found, CC Phase 89 runs for 40 seconds. Note: Always run CC Phase 89 with the RAW option.	REX, DEX, Demand — run whenever CC and main store tests pass but trouble is suspected in the cache area.	In PDS, DGN:CU a,CC b;RAW,REX! DGN:CU a,CC b;RAW,DEX! DGN:CU a,CC b;RAW:PH 89! In MML, DGN:CU=a,CC=b:RAW,REX; DGN:CU=a,CC=b:RAW,DEX; DGN:CU=a,CC=b:RAW:DATA,PH=89;
CC	90	CC Phase 90 is the same as CC Phase 89 except it will run for up to 20 minutes if no fault is found. Note: Always run CC Phase 90 with the RAW option.	Demand — run whenever an intermittent or data-dependent fault is suspected.	In PDS, DGN:CU a,CC b;RAW:PH 90! In MML, DGN:CU=a,CC=b:RAW:DATA,PH=90;
DMA	10	DMA Phase 10 tests data communication between the DMA channel (in the CU) and the bus interface controller.	Demand	In PDS, DGN:CU a,DMA b:PH 10!

Table B-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DMCH	30	DMCH Phase 30 tests data communications between DSCH port 0 and DDSBS.	Demand	In PDS, DGN:CU a,CH b;RAW:PH 30! In MML, DGN:CU=a,CH=b:RAW:DATA,PH=30;
DMCH	31	DMCH Phase 31 tests data communications between DSCH port 1 and DDSBS.	Demand	In PDS, DGN:CU a,CH b;RAW:PH 31! In MML, DGN:CU=a,CH=b:RAW:DATA,PH=31;
DMCH	32	DMCH Phase 32 tests data communications between DSCH port 2 and DDSBS.	Demand	In PDS, DGN:CU a,CH b;RAW:PH 32! In MML, DGN:CU=a,CH=b:RAW:DATA,PH=32;
DMCH	33	DMCH Phase 33 tests data communications between DSCH port 3 and DDSBS.	Demand	In PDS, DGN:CU a,CH b;RAW:PH 33! In MML, DGN:CU=a,CH=b:RAW:DATA,PH=33;

Table B-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
MASC	45, 46, 48	Memory patterns and access tests to the main memory.	DEX, Demand	In PDS, DGN:CU a,MASC b;DEX! DGN:CU a,MASC b:PH 45! DGN:CU a,MASC b:PH 46! DGN:CU a,MASC b:PH 48! In MML, DGN:CU=a,MASC=b:DEX; DGN:CU=a,MASC=b:DATA,PH=45; DGN:CU=a,MASC=b:DATA,PH=46; DGN:CU=a,MASC=b:DATA,PH=48;
MASC	50, 51, 52, 53	Memory patterns and access tests to the main memory.	Demand	In PDS, DGN:CU a,MASC b:PH 50! DGN:CU a,MASC b:PH 51! DGN:CU a,MASC b:PH 52! DGN:CU a,MASC b:PH 53! In MML, DGN:CU=a,MASC=b:DATA,PH=50; DGN:CU=a,MASC=b:DATA,PH=51; DGN:CU=a,MASC=b:DATA,PH=52; DGN:CU=a,MASC=b:DATA,PH=53;
MASC	95	Memory patterns and access tests to the main memory. MASC Phase 95 allows limits on an interactive diagnostic to be set at run time. (Refer to the "Requesting Interactive Diagnostics" section in Chapter 6.)	Demand	In PDS, DGN:CU a,MASC b:PH 95! In MML, DGN:CU=a,MASC=b:DATA,PH=95;
MASC	96	Update mode and refresh data parity tests to entire main memory. MASC Phase 96 allows limits on an interactive diagnostic to be set at run time. (Refer to the "Requesting Interactive Diagnostics" section in Chapter 6.)	REX, DEX, Demand	In PDS, DGN:CU a,MASC b;REX! DGN:CU a,MASC b;DEX! DGN:CU a,MASC b:PH 96! In MML, DGN:CU a,MASC b:REX; DGN:CU a,MASC b:DEX; DGN:CU=a,MASC=b:DATA,PH=96;

Table B-2. Supplementary Phases for CU (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
UC	90	UC Phase 90 tests the external output of trigger function 0. It also tests the external inputs to trigger functions 1, 2, and 3. Clip pin 523 to pin 533 on the backplane at the UC slot to run UC Phase 90. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 90! In MML, DGN:CU=a,UC=b:DATA,PH=90;
UC	91	UC Phase 91 tests the external output of trigger function 1. It also tests the external inputs to trigger functions 0, 2, and 3. Clip pin 521 to pin 533 on the backplane at the UC slot to run UC Phase 91. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 91! In MML, DGN:CU=a,UC=b:DATA,PH=91;
UC	92	UC Phase 92 tests the external output of trigger function 2. It also tests the external inputs to trigger functions 0, 1, and 3. Clip pin 519 to pin 533 on the backplane at the UC slot to run UC Phase 92. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 92! In MML, DGN:CU=a,UC=b:DATA,PH=92;
UC	93	UC Phase 93 tests the external output of trigger function 3. It also tests the external inputs to trigger functions 0, 1, and 2. Clip pin 517 to pin 533 on the backplane at the UC slot to run UC Phase 93. For information on how to locate pins, refer to the "Locating Pins" section in Chapter 6.	Demand — run to test UC trigger function external inputs and outputs.	In PDS, DGN:CU a,UC b:PH 93! In MML, DGN:CU=a,UC=b:DATA,PH=93;

Table B-3. Supplementary Phases for DFC 2

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DFC	14	DFC 2 Phase 14 executes the bit-wise independence test for the First-In First-Out (FIFO) buffer memory of the bus interface circuit.	DEX, Demand	In PDS, DGN:DFC a;DEX! DGN:DFC a:PH 14! In MML, DGN:DFC=a:DEX; DGN:DFC=a:DATA,PH=14;
DFC	15	DFC 2 Phase 15 tests data communication between the bus interface controller (in the DFC) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU used as a helper unit must be OOS and ATP when the diagnostic is run, and a manual remove of the CU must precede the demand phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:DFC a:PH 15,CU d! In MML, DGN:DFC=a:DATA,PH=15,CU=d;
DFC	90	DFC 2 Phase 90 tests the SCSI bus protocol by looping around data between the two buses. Note: Manual intervention is required to disconnect the existing cables from the DFC and to connect the looparound cable between the two buses where they connect to the DFC.	Demand	In PDS, DGN:DFC a:PH 90! In MML, DGN:DFC=a:DATA,PH=90;

Table B-4. Supplementary Phases for DFC 3

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DFC	15	DFC 3 Phase 15 tests data communication between the bus interface controller (in the DFC) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU used as a helper unit must be OOS and ATP when the diagnostic is run, and a manual remove of the CU must precede the demand phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:DFC a:PH 15,CU d! In MML, DGN:DFC=a:DATA,PH=15,CU=d;

Table B-5. Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
MT	90	MT Phase 90 tests the read/write capability of the tape. Note: A writable tape must be installed.	Demand	In PDS, DGN:MT a:PH90! In MML, DGN:MT a:DATA,PH90!
MT	91	MT Phase 91 tests the write-protect capability of the tape. Note: A write-protected tape from MT Phase 90 must be installed.	Demand	In PDS, DGN:MT a:PH91! In MML, DGN:MT a:DATA,PH91!

Table B-6. Supplementary Phases for IOP 2

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
IOP	16	IOP 2 Phase 16 tests data communication between the bus interface controller (in the IOP) and a DMCH (in the nonactive CU). Specify the nonactive CU in the input message as the helper unit. Note: The CU must be OOS and ATP when the diagnostic is run, and a manual remove of the CU must precede the demand phase in order to ensure that the hardware has been properly initialized.	Demand	In PDS, DGN:IOP a:PH 16,CU d! In MML, DGN:IOP=a:DATA,PH=16,CU=d;

Table B-6. Supplementary Phases for IOP2 (Contd)

Sub-unit	Phase	Explanation	Type of Phase	Recommended Command Formats
DUIC	5	DUIC Phase 5 tests the interface to Direct User Interface (DUI) using the loop-back mode. This phase can only be performed if DUIs are equipped.	Demand	In PDS, DGN:DUIC a:PH 5! In MML, DGN:DUIC=a:DATA,PH=5;
DUIC	6	DUIC Phase 6 tests the send-frames mode to DUI from the controller. This phase can only be performed if DUIs are equipped.	Demand	In PDS, DGN:DUIC a:PH 6! In MML, DGN:DUIC=a:DATA,PH=6;

Main Store Failure Data

Failure data for faults detected in main store may output in the form of a histogram. (Refer to Figure B-1 for an example.) A data pattern is 39 bits wide: 32 data bits and 7 check bits. When the data pattern read back from the store does not match the one that was written, a failure has occurred. Memory testing stops after the first 100 faults are detected.

A detailed failure analysis is output for the first fault. This analysis includes the failing address, expected value, received value, mismatch for both the data and parity bits, and a dump of three main store controller registers.

The number of addresses that failed and a list of the first eight failing addresses is followed by the memory failure histogram which shows the following:

- For each data bit and check bit, how many times a bit failed when a zero was expected and how many times a bit failed when a one was expected.
- For each address bit, how many times the address bit was a zero and how many times it was a one for each of the data bit and check bit failures.
- How many multibit errors were detected.
- How many logic errors were detected (single-bit error indication on a multibit error, or other error detection failures).

The numbers in these columns can be used to determine the location and extent of the problem. For example, in the DATA FAIL column, a 1 would indicate a single-memory cell failure, but a 100 would indicate a fault on a data signal path or a bad memory part. In the ADDRESS FAIL column, if the counts for a particular bit are approximately equal, the address bit is probably good. But, if on all failures a particular address bit always has the same value, then the address bit is stuck at that value.

In the example shown in Figure B-1, data bit 24 is stuck at 0 but only when address bit 2 is 0.

DGN CU 1 MASC 0 PH 41 MEMORY FAILURE RESULTS
FIRST FAULT DATA

FAILING ARRAY	=
FAILING ADDRESS	= X'00100000
DATA MISMATCH	= X'01000000
ACTUAL DATA	= X'54555555
EXPECTED DATA	= X'55555555
PARITY MISMATCH	= X'00000000
ACTUAL PARITY	= X'00000000
EXPECTED PARITY	= X'00000000
MMR	= X'00003270
Error Addr Reg	= X'1B900001
Error Addr Reg	= X'00000000
CC ERROR REG	= X'0FFFFFFF

NUMBER OF FAILING ADDRESSES = 100

FIRST 8 FAILING ADDRESS:

X'100000
X'100008
X'100010
X'100018
X'100020
X'100028
X'100030
X'100038

Figure B-1. Sample Memory Failure Histogram (Sheet 1 of 2)

MEMORY FAILURE HISTOGRAM				
DATA/ADR BITS	DATA FAIL EXP=1 RTN=0	COUNT EXP=0 RTN=1	ADDRESS FAIL AS 0	COUNT AS 1
00	0	0	0	0
01	0	0	0	0
02	0	0	100	0
03	0	0	50	50
04	0	0	50	50
05	0	0	52	48
06	0	0	52	48
07	0	0	52	48
08	0	0	64	36
09	0	0	64	36
10	0	0	100	0
11	0	0	100	0
12	0	0	100	0
13	0	0	100	0
14	0	0	100	0
15	0	0	100	0
16	0	0	100	0
17	0	0	100	0
18	0	0	100	0
19	0	0	100	0
20	0	0	0	100
21	0	0	100	0
22	0	0	100	0
23	0	0	100	0
24	100	0	100	0
25	0	0	0	100
26	0	0		
27	0	0		
28	0	0		
29	0	0		
30	0	0		
31	0	0		
C0	0	0		
C1	0	0		
C2	0	0		
C3	0	0		
C4	0	0		
C5	0	0		
C6	0	0		
C7	0	0		
MULTI	0			
LOGIC	0			

Figure B-1. Sample Memory Failure Histogram (Sheet 2 of 2)

Processor Recovery Messages

This section covers Processor Recovery Messages (PRMs) from Control Unit (CU) CC Phases 64 and 65. The CU under test produces PRMs while executing CC Phases 64 and 65. If CC Phases 64 and 65 execute all their tests (no Test Skipped Reason bits set in phase results) and the expected PRMs are not displayed, there may be a hardware problem with the EAI under test or in its connections to the MTTY Peripheral Controllers (PCs).

CC Phase 64 expected PRM pattern:

```
PRM_z 0123 4567 89AB CDEF xx xx xx
PRM_z 0000 0000 0000 0000 xx xx xx
PRM_z FFFF FFFF FFFF FFFF xx xx xx
```

CC Phase 65 expected PRM pattern:

```
PRM_z 0123 4567 89AB CDEF xx xx xx
PRM_z AAAA AAAA AAAA AAAA xx xx xx
PRM_z 5555 5555 5555 5555 xx xx xx
PRM_z EEEE EEEE EEEE EEEE xx xx xx
PRM_z 1111 1111 1111 1111 xx xx xx
PRM_z 0000 0000 0000 0000 xx xx xx
PRM_z FFFF FFFF FFFF FFFF xx xx xx
```

Where:

x = any hexadecimal number

z = number of the CU under test (0 or 1)

Refer to 254-303-103, *Processor Recovery Messages Guide*, for more information about PRMs.

Diagnostic Information Tables — CU, DFC, IOP, and DCI

The following are the diagnostic information tables found in this appendix.

Table B-7	Central Control
Table B-8	Main Store Controller
Table B-9	Direct Memory Access
Table B-10	Direct Memory Access Channel
Table B-11	Disk File Controller 2
Table B-12	Disk File Controller 3
Table B-13	SCSI Magnetic Tape
Table B-14	Utility Circuit
Table B-15	Input/Output Processor 2
Table B-16	Dual Serial Channel Computer Interconnect

Table B-7. Central Control

Phase	Description
cc01	On-line Maintenance Channel (MCH) Master Side Tests.
cc02	MCH Slave Communication.
cc03	MCH Slave command Tests.
cc04	Microstore Address (MSA) and Microstore Data (MSD) registers in off-line MCH.
cc05	BIST and boundary scan test on CC pack.
cc06	Demand-only phase. Individual Application Specific Integrated Circuit (ASIC) BISTs on CC pack.
cc07	Microcontrol (MC) clock circuit tests, Microinstruction Register (MIR) tests, and MC parity tests.
cc08	MC immediate data path, MCH access to test multiplexer, buffered Bidirectional Gating Register (BGR) parity bits, and path to the CC error.
cc09	MC address sequencer.
cc10	MC address sequencer.
cc11	Read-Only Memory (ROM) in microstore.
cc12	Microstore address bus parity.
cc13	Status register.
cc14	Pulse point register.
cc17	Gate through Data Manipulation Unit (DMU) and barrel shifter to rotate bit function.
cc19	Arithmetic Logic Unit (ALU) matcher.
cc20	DMU Q register.
cc21	DMU X or Z flag and ONES.
cc22	DMU general registers.
cc23	ALU and gating paths.
cc24	24-bit mode.
cc25	DMU looparound.
cc26	Temporary and firmware registers.
cc29	MC privileged bits.
cc31	MC privileged bits.

Table B-7. Central Control (Contd)

Phase	Description
cc32	Bidirectional Gating Bus (BGB), microstore data bus, microstore address bus access, and writable microstore.
cc33	BGB, microstore data bus, microstore address bus access, and writable microstore.
cc34	Routine Exercise (REX) phase. BGB, microstore data bus, microstore address bus access, and writable microstore.
cc35	DMU parity check.
cc38	Store control register.
cc39	Store Data Register (SDR), Store Instruction Register (SIR), Instruction Buffer (IB).
cc40	Store Address Register (SAR), program address register.
cc41	IB parity circuit.
cc42	Address increment and parity.
cc44	SIR path to half-word multiplexer to IB.
cc45	SIR path to IB half-word multiplexer.
cc46	SIR path to path to half-word multiplexer to IB.
cc47	SIR path to half-word multiplexer to IB.
cc50	A and B select and rotate amount multiplexer.
cc51	Linecard register for masking out errors noted by the ERROR register (MASK).
cc52	Find-Low-Zero (FLZ) of DMU.
cc53	Interrupt registers and logic.
cc54	Timers.
cc55	Channel data register.
cc58	Central Control Input/Output (CCIO) bus sanity.
cc59	Off-line MCH master side tests.
cc60	Demand-only phase. Other unit interrupt.
cc61	Error response logic.
cc64	Emergency Action Interface.

Table B-7. Central Control (Contd)

Phase	Description
cc65	Demand Exercise (DEX) phase. EAI commands and display tests.
cc68	Store Control Register (SCR) Store Address Translator (SAT) control bits.
cc69	Address Translation Buffer (ATB) access and parity.
cc70	ATB memory.
cc71	Counter and invalidate circuit.
cc72	SAT address multiplier.
cc73	ATB write control logic.
cc74	ATB hit and miss logic.
cc75	ATB protection logic.
cc76	ATB matching of duplicate circuits.
cc77	Main store Interface.
cc80	Access to cache memories.
cc81	Cache tag and cache memory match.
cc82	Hit/miss matcher.
cc83	Write enable logic.
cc84	ATB relocation address tag memory.
cc85	Address and data parity generator.
cc86	Address and data parity checker.
cc87	Cache store write control logic.
cc88	Cache error check circuitry.
cc89	Routine Exercise (REX) phase. Cache exercise.
cc90	Demand-only phase. Phase Cache exercise.

Table B-8. Main Store Controller

Phase	Description
masc01	Execute the Memory Controller (MCERT) “power-on” initial test.
masc02	Boundary scan test.
masc03	Test basic response from Main Memory (MM) circuit.
masc04	Address looparound test.
masc05	Data path test.
masc06	Test store error A.
masc07	Maintenance command test.
masc08	My store B and C tests.
masc12	Single-bit hamming tests.
masc13	Double-bit error (hamming) tests.
masc14	Test address trap logic.
masc15	Arbiter Random Access Memory (RAM) testing.
masc20	Short march test to Bank 0.
masc21	Short march test to Bank 1.
masc22	Short march test to Bank 2.
masc23	Short march test to Bank 3.
masc24	Short march test to Bank 4.
masc25	Short march test to Bank 5.
masc26	Short march test to Bank 6.
masc27	Short march test to Bank 7.
masc30	Memory operations and parity tests to Bank 0.
masc31	Memory operations and parity tests to Bank 1.
masc32	Memory operations and parity tests to Bank 2.

Table B-8. Main Store Controller (Contd)

Phase	Description
masc33	Memory operations and parity tests to Bank 3.
masc34	Memory operations and parity tests to Bank 4.
masc35	Memory operations and parity tests to Bank 5.
masc36	Memory operations and parity tests to Bank 6.
masc37	Memory operations and parity tests to Bank 7.
masc38	Address looparound through main store update (duplex mode).
masc39	Data looparound through main store update (duplex mode).
masc40	Main store update control and error tests.
masc41	Memory data pattern tests.
masc42	Test of fetch/execute instruction from main store.
masc43	Function testing of update mode and refresh data parity checks (some memory locations).
masc45	Demand Exercise (DEX) phase. Memory data — March pattern tests.
masc46	Demand Exercise (DEX) phase. Memory data — March pattern tests.
masc48	Demand Exercise (DEX) phase. Memory data pattern tests using byte, half, read-modify-write commands.
masc50	Demand-only phase. Memory data — March pattern tests
masc51	Demand-only phase. Memory data — March pattern tests
masc52	Demand-only phase. Memory data — March pattern tests
masc53	Demand-only phase. Memory data — March pattern tests
masc95	Demand-only phase. Run User-Specified — March pattern tests
masc96	Routine Exercise (REX) phase. Function testing of update mode and refresh data parity checks (all memory locations). User may specify test parameters.

Table B-9. Direct Memory Access

Phase	Description
dma01	BIST/boundary scan tests.
dma02	DMA/CCIO bus interface.
dma03	Status tests.
dma04	RAM register tests.
dma05	Sequencer tests.
dma06	Request priority circuit.
dma07	Store address tests.
dma08	Store data tests.
dma09	DMA error tests.
dma10	Demand-only phase. DMA/CCIO bus interface
dma11	DMA/Cache interface tests.
dma13	Main Store Update (MASU) priority and arbitration tests.

Table B-10. Direct Memory Access Channel

Phase	Description
dmch01	Direct IO bus interface.
dmch02	Data register FIFO tests.
dmch03	Error tests.
dmch10	Port 0 selection, interrupt, and service requests.
dmch11	Port 1 selection, interrupt, and service requests.
dmch12	Port 2 selection, interrupt, and service requests.
dmch13	Port 3 selection, interrupt, and service requests.
dmch20	Port 0 Dual Serial Channel (DSCH)/Duplex Dual Serial Bus Selector (DDSBS) communication tests.
dmch21	Port 1 DSCH/DDSBS communication tests.
dmch22	Port 2 DSCH/DDSBS communication tests.
dmch23	Port 3 DSCH/DDSBS communication tests.
dmch30	Demand Only - Port 0 DSCH/DDSBS communication tests.
dmch31	Demand Only - Port 1 DSCH/DDSBS communication tests.
dmch32	Demand Only - Port 2 DSCH/DDSBS communication tests.
dmch33	Demand Only - Port 3 DSCH/DDSBS communication tests.

Table B-11. Disk File Controller 2

Phase	Description
dfcs02	DDSBS interface, DDSBS/Bus Interface Controller (BIC) interface.
dfcs03	BIC — Peripheral Interface Controller (PIC) interface.
dfcs04	Central Processing Unit (CPU) Test 1.
dfcs05	Verify ROM contents.
dfcs06	Test 1 — MB Static Random Access Memory (SRAM).
dfcs07	Host Adapter (HA) register test.
dfcs08	WE32104 Direct Memory Access Controller (DMAC) test.
dfcs09	Dual Universal Asynchronous Receiver/Transmitter (DUART) test.
dfcs10	System Timing Controller (STC).
dfcs11	SCSI protocol controllers test both SCSI buses.
dfcs12	Test that there are NO single-ended SCSI type devices attached to bus.
dfcs13	Test bus reset capability.
dfcs14	Demand Exercise (DEX) phase. Bit wise independent test for HA FIFO.
dfcs15	Demand-only phase. Test the interface to nonactive CU
dfcs90	Demand-only phase. Bus loop test.

Table B-12. Disk File Controller 3

Phase	Description
dfcs02	DDSBS interface, DDSBS/BIC interface.
dfcs03	BIC — HA Control interface.
dfcs04	CPU test.
dfcs05	Verify 1 — byte ROM and FLASH memory contents.
dfcs06	Test the SRAM.
dfcs07	HA register test.
dfcs08	DMAC test.
dfcs10	Test system timers.
dfcs11	SCSI protocol controllers test both SCSI buses.
dfcs12	Test that there are NO single-ended SCSI type devices attached to bus.
dfcs13	Test bus reset capability.
dfcs15	Demand-only phase. Test the interface to nonactive CU.

Table B-13. SCSI Magnetic Tape

Phase	Description
mts01	Diagnose the SCSI device identified in the command: ▪ Reset and verify reset condition. ▪ Execute the BIST of the tape unit.
mts90	Demand-only phase. Verify read/write of blocked data to tape. (Writable tape must be installed.)
mts91	Demand-only phase. Verify write-protection mechanism of the tape unit. (Write-protected tape must be installed.) Phase mts90 must be successfully executed before Phase mts91.

Table B-14. Utility Circuit

Phase	Description
uc01	Bidirectional gating bus access and enable.
uc02	Access matcher, address leads, read/write test.
uc03	Block address matchers, address leads, read/write test.
uc04	Address matcher, address leads, read/write test.
uc05	Unit identification matcher, address leads, read/write test.
uc06	Data matcher, address leads, read/write test.
uc07	Event and trace memory addresses counters, read/write test.
uc08	Transfer trace memory access path unit identification buffers test.
uc09	Trace memory, read/write test.
uc10	Trigger functions in freeze mode tests.
uc11	Trigger functions in run mode without match conditions.
uc12	Access matcher, run mode tests.
uc13	Data matcher, run mode tests.
uc14	Address matcher, virtual address bus (store address register), run mode tests.
uc15	Address matcher, physical address bus, run mode tests.
uc16	Unit identification matcher, run mode tests.
uc17	Trigger functions in run mode with match conditions.
uc18	Transfer trace, run mode tests.
uc19	Trace memory read/write indicator bit test.
uc90	Demand-only phase with special procedure. External input, external output trigger function 0.*
uc91	Demand-only phase with special procedure. External input, external output trigger function 1.*
uc92	Demand-only phase with special procedure. External input, external output trigger function 2.*
uc93	Demand-only phase with special procedure. External input, external output trigger function 3.*

* Refer to the "Supplementary Phases" section for special procedures before doing the test.

Table B-15. Input/Output Processor 2

Phase	Description
iop02	DSCH — duplex dual serial bus selector interface.
iop03	DDSBS — BIC interface and BIC circuitry excluding PIC interface.
iop04	BIC — PIC interface using PIC resident firmware.
iop05	BIC — PIC communication.
iop06	PIC microstore data validity and microstore parity checker.
iop07	PIC microsequencer.
iop08	PIC register and ALU.
iop09	PIC source/destination decoders.
iop10	PIC internal registers.
iop11	PIC interrupt circuitry.
iop12	PIC RAM and sequencer.
iop13	Input/output microprocessor interface hardware.
iop15	Input/Output Microprocessor Interface (IOMI) power control circuitry peripheral controller community power converter/monitor.
iop16	Demand-only phase. Test the interface with the nonactive CU. The nonactive CU must be specified as the helper unit and must be OOS and ATP.

Table B-16. Dual Serial Channel Computer Interconnect

Phase	Description
dci01	DSCH — duplex dual serial bus selector interface.
dci02	DDSBS — BIC interface.
dci03	BIC — PIC interface.

Glossary

A

ACHI

Application Channel Interface.

ACK

Acknowledge.

ACKDB

Acknowledgment Database.

ACO/T

Alarm Cutoff/Test.

Active (ACT)

A unit is currently in service.

ADP

Automatic Diagnostic Process.

AIM

Application Integrity Monitor.

ALU

Arithmetic Logic Unit.

AM

Administrative Module.

AMA

Automatic Message Accounting.

ANSI

American National Standards Institute.

ASCII

American Standard Code for Information Interchange.

ASIC

Application Specific Integrated Circuit.

ASW

All Seems Well.

Asynchronous

A communications method in which time intervals between transmitted characters may be of unequal length and no clock is included with the transmission. Transmission is controlled by start and stop bits at the beginning and end of each character.

ATB

Address Translation Buffer.

ATP

All Tests Pass.

AUD

Audit.

Audit

Validity checks that are performed to assure the proper operation of diagnostic control structure and the accuracy of message and data.

Audit Control Record

Information that the System Integrity Monitor (SIM) needs to schedule and dispatch the audit, such as a unique name and member number for each audit. There are three audit record forms (*audrec*, *auderr*, and *audinst*).

Audit Error Recovery Form (auderr)

An RC/V form used to enter information about one of six error categories in an audit. It is identified by the key field's audit name, member number, and category number. One of these forms must be entered for each error category that applies.

Audit Instance Form (audinst)

An RC/V form used to enter information about one of four possible instances of an audit. It is identified by the key field's audit name, member number, and instance name (which may be NULL for an audit that will never have more than one instance). One of these forms must be entered for each instance of an audit. At least one *audinst* form must be entered for each *audrec* form that is entered.

Audit Interface Library

A series of functions which audits use to communicate with SIM.

Audit Record Form (audrec)

An RC/V form used to enter instance-independent information about an audit. It is identified by the key field's audit name and member number.

Audit Requests

Requests from other processes to run audits.

AUTODRST

Automatic Delayed Restoration.

B

Backplane

A large Printed Wiring Assembly (PWA) that distributes DC power and logic signals between hardware modules. Most circuit pack modules plug into the backplane.

Baud Rate

A measure of the speed of data transfer from a computer to a peripheral device (such as a terminal) or from one device to another. Common baud rates are 300, 1200, 4800, and 9600. As a general guide, divide a baud rate by 10 to get the approximate number of English characters transmitted each second.

BDU

Bus Display Unit.

BGB

Bidirectional Gating Bus.

BGR

Bidirectional Gating Register.

BIC

Bus Interface Controller.

BIST

Built-In Self-Test.

Bitmap

Records the status of each block in the file system, using one bit per block, excluding the blocks used for the super block and inode list.

BOM

Beginning of Media.

Boot

Starts the operating system. The term refers to the fact that the kernel must bootstrap itself from secondary storage into an empty machine.

BOP

Beginning of Partition.

BOT

Beginning of Tape.

BOW

Binary Overwrite.

BPI

Bytes per inch.

Broadcast Warning Message (BWM)

A field update package containing the information needed to make a change in the system software.

BTDB

Boot Time Database.

Bus

One or more conductors that transmit control signals, data, or power. A bus usually acts as a common connection between several locations.

BWM

Broadcast Warning Message.

BWMINT

Broadcast Warning Message Initialization.

Byte Replacement

An update that changes a set of contiguous bytes in the text or data of an updatable object file or its memory image.

C

C/D

Control/Display.

CAC

Cache Memory Controller.

Cache Memory

A high-speed, low-capacity memory filled from a slower-speed main memory. Cache memory provides the CPU with very fast access to a working copy of the currently executing instruction and data stream. The 3B21D computer uses 64 Kb of direct mapped data and instruction cache memory.

Cache Store Unit

The optional system unit that provides a high-speed random access memory used to reduce the apparent Main Store (MAS) access and cycle time.

CAM

Cache Memory.

CAS

Column Address Strobe.

Cathode-Ray Tube (CRT) Terminal

A terminal that uses a cathode-ray tube and a screen with a phosphor coating to display data as opposed to ink and paper.

CATP

Conditional All Tests Passed.

CCIO

Central Control Input/Output.

Central Control (CC)

The control circuitry arithmetic logic unit, rotate and mask unit, general and special registers, maintenance channel, store address translator circuit, and optional cache memory unit that compose the CC.

Central Processing Unit (CPU)

A 32-bit R2000A RISC microprocessor clocked at 16.67 MHz in the 3B20D/3B21D computer with the circuitry to control the interpretation and execution of instructions. The term CPU also refers to the logical processor consisting of the three CPU modules.

CFTMTC

Craft Field Maintenance Command.

CFTSHL

Craft Shell.

CH

Channel.

Clock Daemon (CRON)

A command that creates a daemon whose purpose is to invoke commands at specified dates and times.

Clock Diagnostic Monitor

A clock daemon program that executes commands at specified dates and times according to the instructions in file */usr/lib/crontab*.

CMGR

Capability Manager.

CN

Change Notice.

Command

An instruction to the *UNIX*® system shell program ended by a carriage return. A command is entered to instruct the computer system to perform one or more tasks.

CONFIG

Configuration Manager.

Configuration Control Process

A process that keeps track of the number of instances and amount of time a unit is in the nonactive state. It also records faults and errors reported against specific units.

Contiguous Files

Contain data stored in single multiple extent files.

Control Unit (CU)

That portion of the computer which is switched in and out of service as a unit. A CU includes a CC, a main store, a Direct Memory Access (DMA) unit (when equipped), the input/output channels, and the power control unit.

CPDP

Common Processor Display Page.

CPH

Communications Protocol Handler.

CPID

Circuit Pack Identification.

CPU

Central Processing Unit.

Craft Field Maintenance Commands

A software package that controls, maintains, and administers the software in a format consistent with Program Documentation Standard (PDS) and Man-Machine Language (MML) syntax.

Craft Interface

The hardware, firmware, and software that allows the user access to the 3B20D/3B21D computers and provides information about the status of the machine.

Craft Shell

A process that reads input from the bottom line of the user interface terminal, checks it for proper PDS/MML syntax, and then creates appropriate child processes to execute the desired command.

Crash

A condition in which the system takes itself out of service. Such a condition occurs when the system encounters a hardware or software error that it cannot handle.

CRC

Cyclic Redundancy Check.

CRON

Clock Daemon.

CRT

Cathode-Ray Tube.

CSCANS

Customer Service Computer Access Network System.

CSOP

Coordinator of Spooler Output Processes.

CSU

Cache Storage Unit.

CTIP

Customer Training and Information Products organization.

CU

Control Unit.

CUMEM

Control Unit Memory.

CUSTAT

Control Unit Status.

D

Daemon

A background process, often perpetual, that performs a system-wide public function; for example, *CRON*.

DAP

Display Administration Process.

DAT Unit

Digital Audio Tape Unit.

Data Manipulation Unit

The circuit that provides the Remote Measurement Unit (RMU), the ALU, and logic functions.

Daughterboard

A small Printed Wiring Assembly (PWA) that mounts on another large PWA.

DBS

Duplex Bus Selector.

DCE

Data Communications Equipment.

DCI

Dual Serial Channel (DSCH) Computer Interconnect.

dcn

Device Control Number.

DCS

Diagnostic Control Structure.

DCT

Dispatcher Control Table.

DCTE

Dispatcher Control Table Entry.

DDS

Digital Data Storage.

DDSBS

Duplex Dual Serial Bus Selector.

Defensive Check

A means to guard against run-time errors in the operating system or in application processes.

Demand Exercise Phase

A diagnostic phase run when specified in the command line or when the Demand Exercise (DEX) option is used.

Demand Phase

A phase that has a very long execution time and/or requires manual action to complete successfully. To execute a demand phase, the phase number must be specified or included in a range of phases, and the Routine Exercise (REX) or DEX options must NOT be specified.

DEX

Demand Exercise.

DFC

Disk File Controller.

DFSA

Disk File System Access.

DGN

Diagnostic.

DIAGC

Diagnostic Control Process.

DIAMON

Diagnostic Monitor.

Differential SCSI

A receiver/driver design for the Small Computer System Interface (SCSI) bus that provides for a maximum bus length of 25 meters (versus 6 meters for the single-ended SCSI bus). The differential SCSI bus is used with the optional 9-track tape drive.

DIO

Direct Memory Access Input/Output.

DIOP

Disk Independent Operation.

Direct Memory Access (DMA)

A method used to perform high-speed data transfers between memory locations without CPU involvement.

Direct Memory Access Controller (DMAC)

The system unit that provides Main Store (MAS) access control for Input Output (IO) devices without requiring direct control from the CC.

Directory

A file that is a catalog of filenames, providing the organizing principle of the file system. A directory forms a node of the directory tree and consists of entries that specify further files.

Disk File Controller (DFC)

A unit that controls the operation of a portion of the disk storage facility of the *UNIX* Real-Time Reliable (RTR) operating system. This unit interfaces to an IO channel and to a Moving Head Disk (MHD) drive.

Disk Image

A software entity stored on disk; for example, a file or a process.

Display Administration Process

A process that controls the graphic displays on the user Maintenance CRT (MCRT) terminal. It accepts requests from other processes via messages, changes its internal representation of the displayed page, and then displays it on the terminal.

DKDRV

Disk Driver.

DLGSHL

Dialogue Shell.

DMA

Direct Memory Access.

DMAC

Direct Memory Access Controller.

DMCH

Direct Memory Access Channel.

DML

Data Manipulation Language.

DMQ

Deferred Maintenance Queue.

DMU

Data Manipulation Unit.

DP

Data Path.

DRAM

Dynamic Random Access Memory.

Dual Serial Channel (DSCH)

A system component that provides serial interfaces for up to 16 devices.

DTE

Data Terminal Equipment.

DUART

Dual Universal Asynchronous Receiver/Transmitter.

DUC

Dual Utility Circuit.

DUI

Direct User Interface.

DUIC

Direct User Interface Controller.

DUP

Disk Unit Package.

Dynamic Random Access Memory (DRAM)

A type of memory that requires constant refreshing to maintain its contents.

E

EAI

Emergency Action Interface.

ECD

Equipment Configuration Database.

ECD Common Access Functions (ECDFUNC)/Low-Level Access (LLA)

The ECDFUNC and LLA commands are used together to provide kernel or read only user access to the ECD.

ECDFUNC

Equipment Configuration Database Common Access Functions.

ECDOWN

Equipment Configuration Database Owner.

EGRASP

Enhanced Generic Access Package.

EIA

Electronics Industries Association.

EIA-232-D

An Electronics Industries Association standard for serial binary communications. The January 1987 revision of this standard replaces the EIA recommended standard RS-232-C. (Refer to RS-232-C.)

EIH

Error Interrupt Handler.

EIR

Enhanced Information Report.

Electronic Programmable Read-Only Memory (EPROM)

A Programmable Read-Only Memory (PROM) device whose contents can be reprogrammed after being exposed to ultraviolet light.

Emergency Action Interface (EAI)

The EAI provides an interface between a CU and a serial data link for effecting various manual override functions such as "force on-line," "stop and switch," and so forth. The EAI is part of the maintenance interface.

EMI filter

Electromagnetic interference filter used in the power distribution unit and bulk power supplies to reduce transient electrical noise found on power lines.

EMM

Extended Main Memory.

EOM

End of Medium.

EOT

End of Tape (marker).

EPROM

Electronic Programmable Read-Only Memory.

EQL

Equipment Location.

Equipment Configuration Database (ECD)

A database that describes the physical and logical configurations of the 3B20D/3B21D computers and peripheral hardware. It also contains configuration and control information for the audit and operator interface software systems.

ER

Error Register.

Error Interrupt Handler (EIH)

A process that keeps track of the source and level of all system initializations. It also maintains the record of each level of processor error interrupts.

ESAC

Electronic Systems Assistance Center.

Essential System Image

The volatile state of the kernel, user processes, and hardware devices. The system saves the essential system image on the power-fail partition if a power failure or overheating condition occurs and if the power-fail/automatic restart resume option is in effect.

Ethernet

A widely-used base-band local area network specification developed by Xerox, Intel, and DEC. It utilizes coaxial cable and transceivers to interconnect computer equipment.

Event

An interprocess communication mechanism for user processes used to synchronize processes.

EWEarly Warning.

Executable Process (pfile)

A program that can be run on a computer.

Expansion Slots

The Expansion slots (EX) provide an interface for new (future) hardware features to access the Main Store (MAS) and CC via the Central Control Input/Output (CCIO) bus and the Main Store Bus (MASB). Each CU has two Expansion slots, EX 0 and EX 1.

Extended Main Memory (EMM)

A feature consisting of 3B20D/3B21D computer CUs with 32-megabyte memory addressing capability in the main store only, and only 16 megabytes of memory available for virtual addressing, CSU, and DMA.

Extent

A range of contiguous blocks in a file system specified by starting address and size.

F

FDS

Force Disk Selection.

Field Update

A method that allows fixes (also called patches or overwrites) to a *UNIX* operating system without interfering with the running system.

FIFO

First-In First-Out.

File Replacement

An update that adds or changes the entire file on disk.

File System

A collection of files that can be mounted on a block special file, typically a disk partition. Each file of a file system appears exactly once in the inode list of the file system and is accessible through some path from the root directory of the file system. Also, the collection of all files on a computer system.

FLZ

Find-Low-Zero.

FMGR

File Manager.

FSB

File-System Body.

FSBLK

File-System Block.

FSCMPT

File-System Disk Space Compaction.

FSLINK

File-System Link.

FTS

Field Test Set.

Function Replacement

An update to an (updatable) object file and/or its memory image based on a specific set of changes to the source, in which functions or data are changed.

G

Generic

A release of hardware or software.

GRASP

Generic Access Package.

Grow

The state of a unit while it is being added to the system (grown).

H

HA

Host Adapter.

HDR

Header.

Host

A computer system that is configured to share resources, such as disks and printers, in a networked environment.

HSD

High-Speed Data Link.

HSDC

High-Speed Data Link Controller.

HSR

Hardware Status Register.

HV

Hardware Version.

I

IB

Instruction Buffer.

ID

Identification.

IM/OM

Input Messages/Output Messages.

IMCTLG

Input Message Catalogue.

IMR

Interrupt Mask Register.

Incore Database

A database read from a disk that is resident (incore) in the machine.

Indirect Block

A block that contains pointers. There are three types of indirect blocks:

- Single indirect block – a block that contains pointers to data blocks.
- Double indirect block – a block that contains pointers to single indirect blocks.
- Triple indirect block – a block that contains pointers to double indirect blocks.

Initialization

An action taken to provide the system with a known good and operating configuration.

Initialization Level

A value of initialization level counter. The severity of the action taken to recover the system integrity is dependent on the counter value. The higher the value, the more drastic the action taken to recover the system.

Inode

An element of a file system; an inode specifies all properties of a particular file and locates the contents of a file, if any.

Input/Output Processor (IOP)

A front-end processor that permits the connection of a variety of data terminals utilizing a variety of line protocols. The IOP contains a peripheral interface controller, and IOP interface, and two microprocessor communities. Each microprocessor community connects to a number of data terminals.

INS

In-Service.

Interrupt

A signal, caused by a break or an interrupt character, that normally terminates a process. It is also a signal generated by a hardware condition or a peripheral device.

IO

Input/Output.

IODRV

Input/Output Processor Driver.

IOMI

Input/Output Microprocessor Interface.

IOP

Input/Output Processor.

IS

Interrupt Source.

K

Killable Kernel Process Update

Provides the addition of a capability to field update which allows the memory image of killable, kernel processes to be updated via function or byte replacement with full functionality.

Killable Process

A process that may leave main store without disrupting service.

KDAP

Kernel Display Application.

KDAPDRVR

Kernel Display Application Driver.

KPUPDATE

Kernel Process Update.

L

LDFT

Load Disk from Tape.

LIBC

Library C.

Light-Emitting Diode (LED)

A semiconductor that emits visible light.

Link

An entry for an existing file to a directory.

LLA

Low-Level Access.

LSB

Least Significant Bit.

M

Main Memory Image

A software entity located in main store; for example, a process.

Main Store (MAS)

The system unit that provides storage for program instructions and data. It is composed of solid-state memory devices and provides up to 4 megabytes of storage.

Main Store Array (MASA)

The circuit that provides for the storage of words.

Main Store Controller (MASC)

The circuit that provides control interface between the main store bus and the memory devices.

Maintenance Channel (MCH)

The CC component that provides a serial access between CUs.

Maintenance Input Request Administrator (MIRA)

The mechanism by which requests are made to remove a unit from service and to restore a unit to service.

Maintenance Terminal

The terminal that is used for exchange of maintenance information/commands between the maintenance person and the computer. It is also referred to as Maintenance CRT (MCRT) or Maintenance Teletypewriter (MTTY).

MAN

Manual.

Manual Input Commands

Program Documentation Standard (PDS) and Man-Machine Language (MML) format input commands that allow the user to query and request system services.

MASB

Main Store Bus.

MASK

A line card register for masking out errors noted by the ERROR register.

MASU

Main Store Update.

MASUB

Main Store Update Bus.

MC

Microcontrol or Microcontroller.

MCC

Master Control Center.

MCERT

Memory Controller.

MCH

Maintenance Channel.

MCRT

Maintenance CRT.

MCS

Microcontrol Store.

MDCT or mdct

Minor Device Chain Table.

MER

Memory Error Register.

Message

The mechanism that transfers data from one process to another.

MHD

Moving Head Disk.

MHDC

Moving Head Disk Controller.

MHDDC

Moving Head Disk Data Clock.

Microcontrol (MC)

The portion of the CC that controls the sequencing of microstore and decoding of microinstructions. MC provides control signals for the CC circuitry.

Microstore (MIS)

The portion of the CC that provides memory containing the microprogram used for instruction decoding and overall control of the CC.

MIR

Microinstruction Register.

MIRA

Maintenance Input Request Administrator.

Mirroring

A procedure that associates a special device file (*dev/dsk/mir n* , where *n* is an integer) with a pair of disk partitions.

MIS

Microstore or Microinstruction Store.

MLTS

Micro Level Test Set.

MLTSI

Micro Level Test Set Interface.

MMGR

Memory Manager.

MML

Man-Machine Language.

MMR

Main Memory Register.

MOR

Manual Override.

MPC

Multi-Purpose Peripheral Controller.

MPU

Microprogram Unit.

MRF

Maintenance Reset Function.

MSA

Microstore Address.

MSB

Most Significant Bit.

MSACK

Message Acknowledge.

MSD

Microstore Data.

MSDEAD

Message Dead.

MSGBUF

Message Buffer.

MSGs file

An ordered list of executable commands used to install changes to the system using field update.

MT

Magnetic Tape.

MTC

Magnetic Tape Controller.

MTTY

Maintenance Teletypewriter.

MTTYC

Maintenance Teletypewriter Controller.

MTU

Magnetic Tape Unit.

Multitextent File

Contains contiguous data stored in up to six separate extents. Each extent is specified by its starting address and size.

N

N/A

Not Applicable.

Network

The hardware and software that connect computer systems, permitting electronic communications between the systems and associated peripherals.

NOG

Number of Generic.

Nonkillable Process

A process that may not die or leave main store without disrupting service. Nonkillable processes have permanent residence in main store.

Nonsegmented Audit

An audit that runs to completion and reports the amount of CPU time used. All *UNIX* system and supervisor-level audits are nonsegmented.

NPH

Network Protocol Handler.

NTP

Non Trouble Procedure

NTR

No Tests Run.

O

Object File

The file that contains the native mode machine language instructions (object code).

ODD

Office Dependent Data.

Off-Line (OFL)

The unit which is not in active control of the system configuration and execution but which may be actively performing off-line functions; for example, executing diagnostics.

OFL

Off-Line.

OFLBOOT

Off-Line Boot.

OMDB

Output Message Database.

On-Line

The unit which is in active control of the system configuration and execution. Its mate, the off-line CU, may be active (executing diagnostics) but is not in control.

Operating system

A set of programs (or software) that controls the computer system. The nucleus of the *UNIX* Real-Time Reliable (RTR) operating system is the kernel, which manages the resources of the computer. The operating system takes care of such things as IO procedures, process scheduling, and the file system which relieves user programs of these tasks.

OOS

Out of Service.

OSPS

Operator Services Position System.

OST

Operating System Trap.

P

PA

Program Address.

Page

A fixed length, 4-Kb block that has a virtual address and can be transferred between main and secondary storage.

Page Descriptor Table

Contains administrative information about each page of physical memory. There is a one-to-one correspondence between a physical page and a Page Descriptor Entry (PDE).

Page Table

A list of physical pages in main memory that belong to a segment. Each member of the list is referred to as a Page Entry (PGE). The MMGR administers an array of Page Tables (PGTs) and allocates a PGT to every segment that resides in main memory.

Paging

The process by which programs are divided into pages and transferred between main and secondary.

Panic

A *UNIX* system reaction to a problem that it encounters and cannot fix. A panic halts the system.

Parse

To resolve data into a collection of predefined tokens based on syntactical relationships. For example, a sentence can be parsed into words. Parsing is an operation in the executions of terminal messages.

PAS

Protected Application Segment.

Patch Space

Reserved address space where functions are added or replaced.

PC

Peripheral Controller.

PCB

Process Control Block.

PCPAUD

Processor Control Process Audit.

PCPEIH

Processor Control Process Error Interrupt Handler.

PCPMD

Processor Control Process Maintenance Driver.

PD

Peripheral Device.

PDE

Page Descriptor Entry.

PDF

Power Distribution Frame.

PDI

Peripheral Disk Interface.

PDS

Program Documentation Standard.

PDT

Physical Disk-to-Tape.

PECC

Product Engineering Control Center.

pfile

3B20D/3B21D computers executable process file.

PGT

Page Table.

PIC

Peripheral Interface Controller.

PID

Program Identification Number.

PINIT

Processor Initialization.

PIO

Programmed Input/Output.

PIR

Programmed Interrupt Request.

Plant Measurements System (PMS) Database

The database that stores system-performance information. Counts of audit attempts and failures are stored in the PMS database.

PMDB

Plant Measurements Database.

PMGR

Process Manager.

PMS

Plant Measurements System.

POA

Priority of Action.

Poke

The activation of a virtual key that is a portion of a virtual panel display on a Maintenance CRT (MCRT) or Maintenance Teletypewriter (MTTY). A poke is performed via cursor position or menu mode keyboard input. It is a numbered command (entered from display pages) that performs desired procedures.

PPR

Pulse Point Register.

Printed Wiring Assembly (PWA)

A flat board made of multiple layers that holds chips and other electronic components and contains the conductive pathways between them.

PRM

Processor Recovery Message.

PROAD

Process Administration.

Process

An executing program with its own input, output, and "parent" program that initiated it.

Process Control Block (PCB)

The segment of a process that contains its controlling parameters and virtual memory administration structures.

PROM

Programmable Read-Only Memory.

PSBR

Primary Segmentation Base Register.

PSDI

Parallel Serial Data Interface.

PSM

Power Switch Monitor.

PSS

Programmer Support System.

PSW

Processor Status Word.

PTMP

Pending Temp.

PWA

Printed Wiring Assembly.

Q

Quasi-killable Process

A monitored killable process. A “watchdog” process restarts a quasi-killable process if it dies.

Queue

A waiting list usually consisting of processes or tasks waiting for processor time to execute.

R

Random Access Memory (RAM)

A type of memory that can be written to or read from in any address location in any order. The contents of RAM are lost when power is removed.

RAS

Row Address Strobe.

RDY

Ready.

Read-Only Memory (ROM)

A type of memory in which data is stored permanently. Data can be read from but not written to ROM. Instructions and data used repeatedly without change are often kept in ROM.

Real-Time

Pertains to computation performed while the related physical process is taking place. Results of the computation can be used in guiding the physical process.

Receive-Only Printer (ROP)

Output device printer.

Recent Change and Verify (RC/V)

A procedure used for creating or modifying records in the ECD.

Reduced Instruction Set Computer (RISC)

A microprocessor architecture that implements a smaller instruction set and, thereby, turns much of the low-level control of the machine over to software. A RISC processor, by using only the simplest instructions that predominate in system kernels and object modules, improves performance by reducing cycles per instruction and time per cycle.

REG

Register.

Regular File

Collection of individual blocks which may be scattered throughout a file system. Each block is specified only by its address.

Regular Phase

Diagnostic phase which will be executed by default when the simple **dgn: unit** command is entered.

REX

Routine Exercise.

RID

Record Identification.

RMF

Recovery Message Formatter.

RMU

Remote Measurement Unit.

Robust

In reference to a program, the ability to gracefully handle error or failure situations and continue to run.

Root

The directory that constitutes the origin of the directory hierarchy in a file system. Specifically, the origin for the file system with the pathname.

ROS

Request Out of Service.

Routine Exercise Phase

The diagnostic phase which is run automatically when the requesting process is REX, when specified in the command line, or when REX or DEX options are used.

RQIP

Request In Progress.

RS-232-C

A recommended standard published by the Electronics Industries Association (EIA) that specifies the mechanical and electrical characteristics of the interface for connecting DTE and DCE. It defines interface circuit functions and their corresponding connector pin assignments. The standard applies to both asynchronous and synchronous serial, binary data transmission at speeds up to 20 kbps in full- or half-duplex mode. This recommended standard was replaced by the EIA standard EIA-232-D and is functionally compatible with the Consultative Committee International Telegraph and Telephone (CCITT) Recommendation V.24.

RST

Restore.

RTR

Real-Time Reliable.

S

SAC

Store Address Controller.

SAI

Store Address Interface.

SAR

Store Address Register.

SAS

Stop-and-Switch.

SAT

Store Address Translator.

SBUS

Small Computer System Interface Bus.

SC

Scan Point.

SC/SD

Scanner and Signal Distributor.

SCANS

Software Change Administration and Notification System.

SCC

Switching Control Center.

SCCS

Switching Control Center System.

SCR

Store Control Register.

SCSD

Scan Point and Signal Distributor.

SCSDA

Scan Point and Signal Distributor Administrator.

SCSDBODY

An ECD form that gives information for one logical group of an SCSD option block.

SCSDC

Scan Point and Signal Distributor Controller.

SCSDH

Scan Point and Signal Distributor Handler.

SCSDMAP

An ECD form that is review-only data from the SCSD option block.

SCSDOPT

An ECD database record that gives the option name for the SCSD PC in the IOP.

SCSI

Small Computer System Interface.

SD

Signal Distributor.

SDC

Store Data Control.

SDE

Segment Descriptor Entry.

SDF

Special Device File.

SDI

Store Data Interface.

SDL

Synchronous Data Link.

SDLC

Synchronous Data Link Controller.

SDLRT

Synchronous Data Link Restore Tool.

SDP

Software Demand Paging.

SDR

Store Data Register.

Segment

A logically related portion of main memory with consecutive virtual addresses.

Segment Descriptor Table (SDT)

The table that contains information that is used to administer segments of memory. Each entry in the table is referred to as an SDE. An SDE is allocated on the creation of a segment and remains allocated until it is no longer referenced by any process. An SDE exists for every segment regardless of whether the segment is in main memory or swapped out.

Segmented Audits

Relinquish the CPU after a predetermined amount of time. Most kernel-level audits execute in a segmented mode.

Serial Channel

A system component that provides a 16- or 32-bit serial data interface to IO devices.

SG

System Generation.

Shadow File

An always-contiguous file that exists to preserve contiguous file space for future updates.

Shell

The program which causes other programs to be executed on command. The shell is usually started on a user's behalf when the user logs in.

Shell Script

An executable file of commands taken as input by the shell.

SIM

System Integrity Monitor.

Single-Ended SCSI

Receiver/driver design for the SCSI bus that allows extremely fast data transfer rates over the bus. The total bus length is limited to 6 meters, including PWA wiring and cables. This is the standard SCSI bus used to control disk and cartridge tape modules.

SIOF

System Integrity Output Formatter.

SIR

Store Instruction Register.

SIV

Software Integrity Verification.

Small Computer System Interface (SCSI)

An interface standard (ANSI X3.131-1986) for an IO bus used to connect peripheral devices and processors (SCSI devices). A SCSI device requests access to (and gains control of) the SCSI bus in order to perform data transfers with other SCSI devices using a well-defined set of commands.

SMD

Storage Module Drive.

SMI

Selectable Microprocessor Interface.

SMM

Standard Main Memory.

Software Integrity Subsystem

A system feature designed to handle all aspects of software reliability. The software integrity subsystem is composed of SIM, the ECD, audits, the system-integrity-output formatter, manual input commands, audit requests, and a PMS interface.

SPC

SCSI Protocol Controller.

Special Device Files

The software associated with input/output devices.

SRAM

Static Random Access Memory.

SREG

Special Registers.

SSA

Status Summary Area.

SSBR

Secondary Segmentation Base Register.

SSR

System Status Register.

Standard Error

A file descriptor that is normally associated with the terminal on which a parsed input line is entered. The standard error output should be used for returning PDS/MML command acknowledgments such as "OK" or interactive responses for multiline input. It should never be used for other kinds of output to the craft interface terminal. These outputs should be routed through the output spooler.

Standard Input

Normally the keyboard of a terminal. Most PDS/MML commands should not read the standard input since the parsed input line has been passed to the command as **argv** parameters. However, if the input line is terminated by a slash, which is used to indicate multiline input, the command may read the standard for succeeding lines.

Standard Output

A PDS/MML command will normally be attached to the output spooler. The standard output should be used for all the usual output from the PDS/MML command, diagnostic dumps, audit output, and so forth.

Standby (STBY)

The state of a unit that is operational and available for immediate service. It is either not communicating or is not the active half of a duplex pair of devices.

Static Random Access Memory (SRAM)

A type of memory that requires power to maintain its contents but does not require constant refreshing.

STBY

Standby.

STC

System Timing Controller.

STF

Some Tests Fail.

Stop-and-Switch

The signal that causes the standby CU to become the active CU, and the active CU to become the standby CU. It is sent from one CU to the other.

SU

Software Update.

SUPR

System Update Procedure.

Synchronous

A communications method in which there is a constant time between successive bits, characters, or events. The timing is achieved by the sharing of a clock signal between the transmitter and receiver.

System Console

A terminal connected to the system console port that is typically used for administrative, control, and maintenance functions for the system.

System Integrity Monitor (SIM)

Software integrity and audit-control process. SIM is responsible for scheduling all audits.

System Integrity Monitor Control Record

Information stored in the ECD and used by SIM to regulate the percentage of system time to be spent running audits and the scheduling of manually requested audits. A master-inhibit state in this record is used to turn off the scheduling of all routine audits.

System Integrity Output Formatter

A program that provides a consistent output message format when reporting audit results to the user.

System Memory

The combination of main store memory and all other memory for the system.

System Update

The procedure used to introduce an entire new generic, as opposed to selected fixes, on 3B20D/3B21D computers.

System Update Procedure (SUPR)

A collection of processes that install new versions of the *UNIX* system and application software into 3B20D/3B21D computers.

T

TA

Trapped Address.

TAR

Trapped Address Register.

Terminator Plug

A plug installed at the physical end of a bus. A terminator usually is installed on each line of a bus to prevent signal distortion.

TLP

Trouble-Locating Procedure.

TMR

Timer Register.

Token

A defined character or sequence of characters that represent a particular action, object, or a type of delimiter. For the computer, most tokens are either verbs, identifiers, or delimiters.

TOP

Tape-Only Program.

TTY

Teletypewriter.

TTY Controller (TTYC)

The control circuit that interfaces to an IOP and provides TTY communication.

TTYC

Teletypewriter Controller.

TTYPC

TTY Peripheral Controller.

U

UC

Utility Circuit.

UCB

Unit Control Block.

UCL

Unconditional.

UER

Microinterrupt Error Register.

UID

Utility Identification.

ULARP

User-Level Automatic Restart Process.

UNAV

Unavailable.

UNEQ

Unequipped.

Utility Circuit (UC)

The optional system unit that provides address and data matchers and transfer trace capability. This unit is used for software debugging.

V

V.35

A Consultative Committee International Telegraph and Telephone (CCITT) standard for data transmission at 48 kbps per second using 60 through 108 kHz group band circuits. The V.35 interface is typically used on DTE or DCE interfacing to a high-speed digital carrier.

VCP

Volume Copy.

Very Large Main Memory (VLMM)

Feature consisting of 3B20D/3B21D computer CUs with 64-megabyte memory address spectrum.

VLMM

Very Large Main Memory.

VORA

Virtual Address Out of Range.

VTOC

Volume Table of Contents.

W

WER

Write-Enable Ring.

WMS

Writable Microstore.

WPA

Words Per Array.

Writable Microstore

The CC component uses readable and writable random access memory to store microprograms.

Z

ZDP

Zero Disk Partition.

Index

- 160-MB Minimodule Disk Frame, 1-33
 - Disk Inverters, 1-33
 - Moving Head Disk Units, 1-33
- 300 V A Inverter Synthesizer, 1-29
- 300-MB Disk, 1-12, 1-29
- 300-MB MHD Frame, 1-27
 - 300-MB Disk, 1-29
 - Control Panel, 1-28
 - Disk Inverter, 1-28
- 340-MB Disk, 1-29, 1-47
- 3B20D Computer, 3-81
- 3B20D Computer Hardware Overview, 1-1
- 3B20D Computer References,
 - Control Unit Diagnostic Test Sequence, A-2
 - Diagnostic Hierarchy, A-4
 - Diagnostic Information Tables CU, DFC, and IOP, A-30
 - Main Store Failure Data, A-23
 - Processor Recovery Messages, A-29
 - Supplementary Phases, A-11
 - Test Procedures, A-6
- 3B20D Computer Supplementary Phases,
 - Recommended Frequency of Selected Supplementary Phases, A-11
 - Supplementary Phases for CU, A-11
 - Supplementary Phases for IOP, A-11
 - Supplementary Phases for SCSI DFC, A-11
 - Supplementary Phases for SMD DFC, A-11
- 3B20D Model 1 Computer Hardware, 1-4
 - 300-MB Disk, 1-12
 - Central Control Unit, 1-8
 - Control Unit Frame, 1-6
 - Cooling Unit, 1-9, 1-16
 - Direct Memory Access Input/Output Unit, 1-6
 - Disk Inverter, 1-12
 - Format Control Unit, 1-14
 - Input/Output Processor Unit, 1-11
 - IOP and SMD Type DFC Growth Frames, 1-14
 - Magnetic Tape Unit, 1-14
 - Main Store Unit, 1-9
 - MHD Control Panel, 1-12
 - Moving Head Disk Frame, 1-12
 - Peripheral Control Frame, 1-10
 - Power Distribution Frame, 1-16
 - Power Distribution Unit, 1-16
 - Power Units, 1-9
- 3B20D Model 1 Computer Hardware (Continued)
 - SCSI DFC Units, 1-16
 - SCSI Disk Cabinet (Optional), 1-14
 - SCSI DUP Units, 1-16
 - Storage Module Drive Disk File Controller Unit, 1-11
 - Tape Unit Frame, 1-14
- 3B20D Model 2 Computer Hardware, 1-18
 - 160-MB Minimodule Disk Frame, 1-33
 - 300 V A Inverter Synthesizer, 1-29
 - 300-MB Disk, 1-29
 - 300-MB MHD Frame, 1-27
 - 340-MB Disk, 1-29
 - Central Processor Unit, 1-23
 - Control Panel, 1-28
 - Cooling Unit, 1-31
 - Disk Inverter, 1-28
 - Disk Inverters, 1-33
 - Disk Power Supply, 1-30
 - Disk Power Switch, 1-30
 - Growth Unit, 1-27
 - IOP Basic Unit, 1-26
 - Magnetic Tape Unit, 1-29
 - MAS-IO-DFC Unit, 1-24
 - Moving Head Disk Units, 1-33
 - Power Distribution Frame, 1-33
 - Power Distribution Unit, 1-31
 - Processor Control Frame, 1-18
 - SCSI DFC Units, 1-31
 - SCSI Disk Cabinet (Optional, Found in Release 6 Only), 1-31
 - SCSI DUP Units, 1-31
 - Tape Drive, 1-29
 - Tape Unit Frame, 1-29
 - Tape/Disk Cabinet, 1-29
- 3B20D Model 3 Computer Hardware, 1-36
 - 340-MB Disk, 1-47
 - Central Processor Unit, 1-42
 - Cooling Unit, 1-49
 - Disk Power Supply, 1-47
 - Disk Power Switch, 1-47
 - Growth Unit, 1-46
 - IOP Basic Unit, 1-45
 - MAS-IO-DFC Unit, 1-43
 - Power Cabinet (Optional), 1-51
 - Power Distribution Unit, 1-49
 - Processor Cabinet, 1-36
 - SCSI DFC Units, 1-49
 - SCSI Disk Cabinet (Optional), 1-49
 - SCSI DUP Units, 1-50

- 3B20D Model 3 Computer
 - Hardware (Continued)
 - Tape Drive, 1-46, 1-49
 - Tape/Disk Cabinet, 1-46
- 3B20D/3B21D Computer Product
 - Installation, 22-5
- 3B20D/3B21D Computers TOP Tape, 9-11
- 3B21D Computer, 3-83
 - Recommended Frequency of Selected Supplementary Phases, B-13
 - Supplementary Phases for CU, B-13
 - Supplementary Phases for DFC 2, B-13
 - Supplementary Phases for DFC 3, B-13
 - Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape, B-13
 - Supplementary Phases for IOP 2, B-13
- 3B21D Computer Hardware Overview, 1-2
- 3B21D Computer Processor Cabinet, 1-52
 - Cache Storage Unit, 1-54
 - Central Control, 1-54
 - Control Unit, 1-53
 - Direct Memory Access, 1-54
 - Disk File Controller Unit, 1-55
 - Expansion Slots, 1-55
 - Input/Output Processor Unit, 1-55
 - Main Store and Main Store Update, 1-54
 - Peripheral Growth Cabinet, 1-56
 - Utility Circuit, 1-55
- 3B21D Computer References,
 - Control Unit Diagnostic Test Sequence, B-2
 - Diagnostic Hierarchy, B-4
 - Diagnostic Information Tables CU, DFC, IOP, and DCI, B-30
 - Main Store Failure Data, B-26
 - Processor Recovery Messages, B-29
 - Supplementary Phases, B-13
 - Test Procedures, B-6
- 9-Track Drive Tape Capacity, 19-28
- 9-Track Drive Tape Density, 19-28
- 9-Track Drives, 19-27
- 9-Track Tape Loading/Unloading, 19-30

A

- Abort or Cancel a Request, 6-21
- Access to Scan and Distribute Points, 16-62
- Action Indicators, 13-25
- Activation, 21-9, 21-11
- Add UCL Argument to Specified Command Line Argument, 11-51

- Adding to the Defect Table of a SCSI Disk Drive, 10-23
- Alarms, 16-56
 - Detailed Design Description, 16-57
 - Fan Alarms, 16-56
 - Procedure to Respond to Alarms, 2-15
 - Responding to Alarms, 2-6
- All Terminals Do Not Function, 3-34
- Allow REX Command, 5-5
- Allowable Scope/Action Combinations, 13-27
- Allowing Audits, 14-16
- Analyzing Data for File-System Audits, 14-20
- Application Dependent Processing
 - Example, 11-52
- Application Recovery Procedure, 3-31
- AUD Output Messages, 14-22
- auderr* form for CUMEM 1, category 0, 14-56
- auderr* form for CUMEM 1, category 1, 14-56
- auderr* form for CUSTAT 1, category 0, 14-69
- auderr* form for CUSTAT 1, category 1, 14-69
- auderr* form for CUSTAT 1, category 2, 14-70
- auderr* form for CUSTAT 1, category 3, 14-70
- auderr* form for ECD 1, category 3, 14-88
- auderr* form for ECD 1, category 4, 14-88
- auderr* form for ECD 1, category 5, 14-89
- auderr* form for ECD 2, category 3, 14-89
- auderr* form for ECD 2, category 4, 14-90
- auderr* form for ECD 2, category 5, 14-90
- auderr* form for ECD 3, category 3, 14-91
- auderr* form for ECD 3, category 4, 14-91
- auderr* form for ECD 3, category 5, 14-92
- auderr* form for ECDOWN 1, category 1, 14-99
- auderr* form for ECDOWN 1, category 2, 14-99
- auderr* form for FMGR 1, category 0, 14-117
- auderr* form for FMGR 1, category 5, 14-117
- auderr* form for FMGR 2, category 0, 14-118
- auderr* form for FMGR 2, category 2, 14-118
- auderr* form for FMGR 2, category 3, 14-119
- auderr* form for FMGR 2, category 5, 14-119
- auderr* form for FMGR 3, category 2, 14-120
- auderr* form for FMGR 3, category 4, 14-120
- auderr* form for FMGR 4, category 0, 14-121
- auderr* form for FMGR 4, category 2, 14-121
- auderr* form for FMGR 5, category 1, 14-122
- auderr* form for FMGR 5, category 2, 14-122
- auderr* form for FMGR 6, category 0, 14-123
- auderr* form for FMGR 6, category 1, 14-123
- auderr* form for FMGR 6, category 2, 14-124
- auderr* form for FMGR 6, category 4, 14-124
- auderr* form for FMGR 6, category 5, 14-125
- auderr* form for FMGR 7, category 0, 14-125

- auderr* form for FMGR 7, category 1, 14-126
- auderr* form for FMGR 7, category 2, 14-126
- auderr* form for FMGR 7, category 3, 14-127
- auderr* form for FMGR 8, category 2, 14-127
- auderr* form for FMGR 9, category 5, 14-128
- auderr* form for FSBLK 1, category 1, 14-139
- auderr* form for FSBLK 1, category 3, 14-139
- auderr* form for FSCMPT 1, category 1, 14-164
- auderr* form for FSCMPT 1, category 3, 14-164
- auderr* form for FSCMPT 1, category 5, 14-165
- auderr* form for FSLINK 1, category 1, 14-150
- auderr* form for FSLINK 1, category 2, 14-150
- auderr* form for MMGR 1, category 1, 14-178
- auderr* form for MMGR 1, category 2, 14-179
- auderr* form for MMGR 1, category 3, 14-179
- auderr* form for MMGR 1, category 4, 14-180
- auderr* form for MMGR 1, category 5, 14-180
- auderr* form for MMGR 2, category 1, 14-181
- auderr* form for MMGR 2, category 2, 14-181
- auderr* form for MMGR 2, category 3, 14-182
- auderr* form for MMGR 2, category 4, 14-182
- auderr* form for MMGR 3, category 1, 14-183
- auderr* form for MMGR 3, category 2, 14-183
- auderr* form for MMGR 3, category 3, 14-184
- auderr* form for MMGR 3, category 4, 14-184
- auderr* form for MMGR 4, category 3, 14-185
- auderr* form for MMGR 5, category 3, 14-185
- auderr* form for MMGR 9, category 0, 14-186
- auderr* form for MSGBUF 1, category 0, 14-200
- auderr* form for MSGBUF 1, category 1, 14-200
- auderr* form for MSGBUF 1, category 2, 14-201
- auderr* form for MSGBUF 1, category 3, 14-201
- auderr* form for MSGBUF 1, category 5, 14-202
- auderr* form for MSGBUF 2, category 0, 14-202
- auderr* form for MSGBUF 2, category 1, 14-203
- auderr* form for MSGBUF 2, category 2, 14-203
- auderr* form for MSGBUF 2, category 3, 14-204
- auderr* form for PMS 1, category 1, 14-213
- auderr* form for PMS 1, category 2, 14-213
- auderr* form for PMS 1, category 3, 14-214
- auderr* form for PMS 1, category 4, 14-214
- auderr* form for PMS 1, category 5, 14-215
- auderr* form for PMS 2, category 0, 14-215
- auderr* form for PMS 2, category 3, 14-216
- auderr* form for PMS 2, category 5, 14-216
- auderr* form for PROAD 1, category 1, 14-223
- auderr* form for PROAD 1, category 2, 14-223
- Audience, xxxi
- audinst* form for CUMEM 1, 14-57
- audinst* form for CUSTAT 1, 14-71
- audinst* form for ECD 1, 14-92
- audinst* form for ECD 2, appecd, 14-93
- audinst* form for ECD 2, ecd, 14-93
- audinst* form for ECD 3, 14-94
- audinst* form for ECDOWN 1, 14-100
- audinst* form for FMGR 1, 14-128
- audinst* form for FMGR 2, 14-129
- audinst* form for FMGR 3, 14-129
- audinst* form for FMGR 4, 14-130
- audinst* form for FMGR 5, 14-130
- audinst* form for FMGR 6, 14-131
- audinst* form for FMGR 7, 14-131
- audinst* form for FMGR 8, 14-132
- audinst* form for FMGR 9, 14-132
- audinst* form for FSBLK 1, /dev/bdb, 14-142
- audinst* form for FSBLK 1, /dev/betc, 14-142
- audinst* form for FSBLK 1, /dev/broot, 14-143
- audinst* form for FSBLK 1, /dev/db, 14-140
- audinst* form for FSBLK 1, /dev/etc, 14-140
- audinst* form for FSBLK 1, /dev/root, 14-141
- audinst* form for FSBLK 1, /dev/tmp, 14-141
- audinst* form for FSCMPT 1, /dev/db, 14-165
- audinst* form for FSCMPT 1, /dev/etc, 14-166
- audinst* form for FSCMPT 1, /dev/root, 14-166
- audinst* form for FSLINK 1, /dev/bdb, 14-153
- audinst* form for FSLINK 1, /dev/betc, 14-153
- audinst* form for FSLINK 1, /dev/broot, 14-154
- audinst* form for FSLINK 1, /dev/db, 14-151
- audinst* form for FSLINK 1, /dev/etc, 14-151
- audinst* form for FSLINK 1, /dev/root, 14-152
- audinst* form for FSLINK 1, /dev/tmp, 14-152
- audinst* form for MMGR 1, 14-186
- audinst* form for MMGR 2, 14-187
- audinst* form for MMGR 3, 14-187
- audinst* form for MMGR 4, 14-188
- audinst* form for MMGR 5, 14-188
- audinst* form for MMGR 9, 14-189
- audinst* form for MSGBUF 1, 14-204
- audinst* form for MSGBUF 2, 14-205
- audinst* form for PMS 1, 14-217
- audinst* form for PMS 2, 14-217
- audinst* form for PROAD 1, 14-224
- Audit Control and Scheduling, 14-9
 - Audit Dispatching, 14-10
 - Audit Execution Modes, 14-10
 - Audit Execution Scenario, 14-15
 - Audit Scheduling and Time
 - Consumption, 14-11
 - Audit Segmentation, 14-10
 - Audit-Control Subsystem Initialization, 14-9
 - Blocking Audits, 14-11
 - Error Reporting and Recovery, 14-14

Audit Control and Scheduling (Continued)

- Frequency Groups, 14-13
- Inhibiting the Routine Execution of Audits, 14-13
- Scheduling Routine Audits, 14-12
- Timed Audits, 14-14

Audit Control Subsystem, 14-5

- Audit Control and Scheduling, 14-9
- Audit Identification, 14-6
- Software Integrity Records, 14-6

Audit Dispatching, 14-10

Audit Execution Modes, 14-10

Audit Execution Scenario, 14-15

Audit Identification, 14-6

Audit Message, Respond to Diagnostic, 6-41

Audit Output Messages and Error Reports, 14-22

- AUD Output Messages, 14-22
- OP AUD Output Message, 14-23
- OP AUDERR Output Message, 14-23
- OP FNAME Output Message, 14-24
- Processor Recovery Messages, 14-24
- REPT AUDSTAT Output Message, 14-26
- REPT SIMCHK Output Message, 14-26

Audit Scheduling and Time Consumption, 14-11

Audit Segmentation, 14-10

Audit User Input Commands, 14-16

- Allowing Audits, 14-16
- Analyzing Data for File-System Audits, 14-20
- Extracting Error Count Information for Audits, 14-19
- Inhibiting Audits, 14-18
- Reporting Status of Audits, 14-19
- Requesting Audits, 14-17
- Stopping Audits, 14-21

Audit-Control Subsystem Initialization, 14-9

- audrec* form for CUMEM 1, 14-55
- audrec* form for CUSTAT 1, 14-68
- audrec* form for ECD 1, 14-85
- audrec* form for ECD 2, 14-86
- audrec* form for ECD 3, 14-87
- audrec* form for ECDOWN 1, 14-98
- audrec* form for FMGR 1, 14-108
- audrec* form for FMGR 2, 14-109
- audrec* form for FMGR 3, 14-110
- audrec* form for FMGR 4, 14-111
- audrec* form for FMGR 5, 14-112
- audrec* form for FMGR 6, 14-113
- audrec* form for FMGR 7, 14-114
- audrec* form for FMGR 8, 14-115

- audrec* form for FMGR 9, 14-116
- audrec* form for FSBLK 1, 14-138
- audrec* form for FSCMPT 1, 14-163
- audrec* form for FSLINK 1, 14-149
- audrec* form for MMGR 1, 14-174
- audrec* form for MMGR 2, 14-175
- audrec* form for MMGR 3, 14-175
- audrec* form for MMGR 4, 14-176
- audrec* form for MMGR 5, 14-177
- audrec* form for MMGR 9, 14-177
- audrec* form for MSGBUF 1, 14-198
- audrec* form for MSGBUF 2, 14-199
- audrec* form for PMS 1, 14-211
- audrec* form for PMS 2, 14-212
- audrec* form for PROAD 1, 14-222

Automatic Diagnostics, 6-7

Automatic Maintenance Requests, Inhibit and Allow, 6-19

Automatic Postmortem Dump Analysis, 4-54

Automatic Postmortem Dumps, 4-4

- BOOT DEVICES, 4-10
- CACHE REGISTERS (3B21D Computer Only), 4-15
- EMERGENCY ACTION INTERFACE BUFFER, 4-10
- FAULTY CU REGISTERS, 4-14
- GENERAL REGISTERS, 4-13
- Heading, 4-5
- INITIALIZATION MESSAGE, 4-6
- INTERRUPT STACK SAVED STATE, 4-14
- MAIN STORE REGISTERS, 4-16
- OFF-LINE REGISTERS, 4-15
- REAL-TIME CLOCK, 4-16
- REQUESTED INITIALIZATION PARAMETERS, 4-9
- TIMER, 4-13

Automatic Recovery Actions, 2-3

Autonomous Report, 16-28

Available Information After OFLBOOT, 7-17

Available Information During OFLBOOT, 7-17

B

- Back Out Last Official SUs, 21-33
- Back Out SUs, 21-32
- Backout, 21-10, 21-11
- Backout Procedures, 16-75
- BACKOUT** Request, 11-6
- Backplane Pin Set, 6-58

- Backup Partition Method, 11-3
- Blocking Audits, 14-11
- BOOT DEVICES, 4-10
- Boot File Verification Through OFLBOOT, 9-10
- Boot Image Component Processes, 13-3
- Boot Procedure, 3-74
- Boot Progress PRMs, 10-10
- Boot to New Generic, 11-40
- Boot to Old Generic, 11-45
- Bootstrap Options, 3-13
- Broadcast Warning Messages, 13-6
 - BWM Components, 13-8
 - BWM File Structure, 13-7
 - Update Package Rules, 13-15
 - Update Package States, 13-18
- BWM Components, 13-8
 - HEADER File, 13-11
 - MSGs File, 13-12
 - SCANS File, 13-9
 - Update Files, 13-15
- BWM File Structure, 13-7

C

- CACHE REGISTERS (3B21D Computer Only), 4-15
- Cache Storage Unit, 1-54
- Call Up the Field Update Page, 13-36
- Canceled and Denied Requests, 6-6
- Capabilities and User Requirements, 16-18
 - Alarms, 16-56
 - Client Processes, 16-18
 - Interface Requirements, 16-30
 - Power Switch Monitor, 16-39
 - SCSD Features and Limitations, 16-23
 - SCSD Handler, 16-23
 - SCSD System and Alarm Requirements and Assignments, 16-51
 - SCSDA Functions, 16-18
 - UN33/UN933 Circuit Pack (SCSDC Board), 16-19
- Capability Modes, 12-122
- Central Control, 1-54
- Central Control Unit, 1-8
- Central Processor Unit, 1-23, 1-42
- Channel Board Installation Procedure, 22-5
- Check/Make Official Temporary SUs, 21-26
- Checking File-System Resources, 9-1
- Checking OFLBOOT Results, 7-17
- Checking OFLBOOT Results (Continued)
 - Available Information After OFLBOOT, 7-17
 - Available Information During OFLBOOT, 7-17
- Checking Operating System Resources, 10-1
- chopt* Form, 22-15
- Circuit Pack Log, 17-2
- classdef* Form, 22-17
- Cleaning,
 - Cleaning KS-23909, L10 9-Track Tape Drives, 19-13
 - Cleaning KS-23909, L21 9-Track Tape Drives, 19-16
 - KS-24367, L1/Comcode 407771260 DAT Drives, 19-19
 - NCR006-3300608/Comcode 40745243 DAT Drives, 19-19
 - NCR006-3503341 DAT Drives, 19-19
- Cleaning Cassette, KS-23860, L800, 19-20
- Cleaning KS-23909, L10 9-Track Tape Drives, 19-13
- Cleaning KS-23909, L21 9-Track Tape Drives, 19-16
- Cleaning SCSI Tape Drives, 19-13
- Clear Trouble Using the Diagnostic Listing, 6-38
- Clear Trouble Using the TLP Circuit Pack List, 6-36
- Clear Trouble, Interpret Diagnostic Output and, 6-25
- Client Process Message Interface, 16-31
- Client Processes, 16-18
- Command Response, 16-27
- Command, Filename, and Display Notations, xxxvi
- COMMIT** Process Terminates, 11-42
- COMMIT** Request, 11-6
- Common TOP Tape, 9-12
- Compare Operation, 9-40
- Comparing and Correcting Disk Data, 9-39
 - Limitations of Disk Compare and Correct Commands, 9-43
 - Operations for Comparing and Correcting Disk Data, 9-39
- Concurrent Maintenance Requests, 6-4
- Conditional and Unconditional Execution Mode, 11-8
- Conditional Execution Mode, 11-8
- Configuration Indicators, 13-24
- Continue Update for Disk 770, 11-33
- Continue Update for Disk 772, 11-35
- Continue Update for Disk 775, 11-37

CONTINUE Utility, 11-7
Control Panel, 1-28
Control Unit, 1-53, B-4
 Cache Storage Unit, 1-54
 Central Control, 1-54
 Direct Memory Access, 1-54
 Expansion Slots, 1-55
 Main Store and Main Store Update, 1-54
 Utility Circuit, 1-55
Control Unit Diagnostic Test Sequence, A-2, B-2
Control Unit Error Messages, 4-2
 Error Interrupt Handler Log File, 4-3
 Memory History Log File, 4-3
Control Unit Frame, 1-6
 Central Control Unit, 1-8
 Cooling Unit, 1-9
 Direct Memory Access Input/Output Unit, 1-6
 Main Store Unit, 1-9
 Power Units, 1-9
Control Unit Hardware Status Audit, 14-58
 CUSTAT Recent Change Forms, 14-68
 Interpreting AUD CUSTAT Output Message and Error Reports, 14-59
 Requesting the CUSTAT Audit, 14-58
 Responding to AUD CUSTAT Error Reports, 14-66
Control Unit Memory Comparison Audit, 14-51
 CUMEM Recent Change Forms, 14-55
 Interpreting AUD CUMEM Output Messages and Error Reports, 14-52
 Requesting the CUMEM Audit, 14-51
 Responding to AUD CUMEM Error Reports, 14-54
Conventions Used,
 Command, Filename, and Display Notations, xxxvi
 Equipment Locations, xxxviii
 Hexadecimal Notation, xxxvii
 Safety Labels, xxxviii
 Signal Designations, xxxviii
Cooling Unit, 1-9, 1-16, 1-31, 1-49
Copying Files to Nonactive Disks, 9-37
Copying to a Spare Disk Drive, 9-26
Copying to Backup Partitions, 9-13
Core Dump Administration, 8-5
Correction Operation, 9-41
Corrective Action for Error Code 141, 14-36
Corrective Action for Error Code 146, 14-36
Corrective Action for Error Code 159, 14-37

Corrective Action for Error Code 190, 14-37
Corrective Action for Error Codes 101, 103, 104, 126, 127, 156, 158, and 183, 14-35
Corrective Action for Error Codes 147, 152, and 154, 14-36
Craft Initialization, 3-42
Craft Interface,
 Maintenance Interface, 12-1
 Maintenance Procedures, 12-4
Craft TTY Interface, 16-6
Creating a Dump Tape, 3-43
CU Diagnostic Hierarchy, A-4
CU Suggested Routine, B-6
CU Suggested Routine Test Procedure, A-6
CUMEM Recent Change Forms, 14-55
 auderr form for CUMEM 1, category 0, 14-56
 auderr form for CUMEM 1, category 1, 14-56
 audinst form for CUMEM 1, 14-57
 audrec form for CUMEM 1, 14-55
CUSTAT Recent Change Forms, 14-68
 auderr form for CUSTAT 1, category 0, 14-69
 auderr form for CUSTAT 1, category 1, 14-69
 auderr form for CUSTAT 1, category 2, 14-70
 auderr form for CUSTAT 1, category 3, 14-70
 audinst form for CUSTAT 1, 14-71
 audrec form for CUSTAT 1, 14-68

D

DAT Drive Cartridge Tapes and Status Indicators, 19-39
DAT Drive Tape Capacity, 19-29
DAT Drive Tape Density, 19-29
DAT Drives, 19-25, 19-27
DAT Loading/Unloading, 19-37
Database Modification Procedures, 10-9
DCI Suggested Routine, B-12
 dciopt Form, 22-15
Definition of Logical Groups, 16-11
Degrowth Procedure for Removing a UN33/UN933 Board, 16-70
Degrowth Procedure for Removing Scan Points on a UN33/UN933 Board, 16-68

- Denied Request, Responding, 6-16
- Description of Error Reports, 4-2
 - Automatic Postmortem Dumps, 4-4
 - Control Unit Error Messages, 4-2
 - Error Log Files, 4-2
 - Processor Recovery Messages, 4-16
- Detailed Design Description, 16-57
 - Fan Alarms and PSM Operation
 - Features, 16-58
 - Power Switch Monitor, 16-57
- device* Form, 22-17
- Device Translation, 16-5
- DFC 2 and DFC 3 Suggested Routine, B-8
- DFC error completion codes, 3-79, 3-84
- DFC Hardware Diagnostic Hierarchy, A-4
- DFC Suggested Routine Test Procedure, A-8
- DGN** Command, Using, 6-8
- Diagnosing SCSI Tape Drives, 19-10
- Diagnosing the DCI Board, 22-11
- Diagnostic Audit Message, Respond to, 6-41
- Diagnostic Commands, 6-4
- Diagnostic Hierarchy, A-4, B-4
 - Control Unit, B-4
 - CU Diagnostic Hierarchy, A-4
 - DFC Hardware Diagnostic Hierarchy, A-4
 - Disk File Controller Hardware, B-4
 - Input/Output Processor, B-5
 - IOP Diagnostic Hierarchy, A-5
- Diagnostic Information Tables CU, DFC, and IOP, A-30
- Diagnostic Information Tables CU, DFC, IOP, and DCI, B-30
- Diagnostic Listing, Clear Trouble Using, 6-38
- Diagnostic Output Message, 6-25
- Diagnostic Processes, 13-5
- Diagnostic Program, Inform of Hardware Changes, 6-23
- Diagnostics,
 - Automatic Diagnostics, 6-7
 - Locating Pins, 6-57
 - Maintenance Input Request Administrator, 6-3
 - Performing Diagnostics, 6-8
 - Processor Recovery Messages, 6-59
 - Synchronous Data Link Restore Tool, 6-54
- DIOP Options, 3-18
 - Faulty Unit Repair, 3-18
 - System Restoration, 3-18
- Direct Memory Access, 1-54
- Direct Memory Access Input/Output Unit, 1-6
- Disk File Controller Hardware, B-4

- Disk File Controller Unit, 1-55
- Disk Inverter, 1-12, 1-28
- Disk Inverters, 1-33
- Disk Power Supply, 1-30, 1-47
- Disk Power Switch, 1-30, 1-47
- Display Enter Arguments Page 2 Screen, 11-23
- Display Page 2 of the APPLY section, 13-40
- Display REX Inhibits, 5-5
- Display System Update Menu Page, 11-16
- Display the APPLY Section, 13-37, 13-39
- Display the PERM Section, 13-41, 13-65
- Display the SOAK Section, 13-59
- Distribute Operation Requests, 16-6
- Documentation, 2-17
- Drive Identification, 19-9
- DSCH Computer Interconnect Feature,
 - Installation Procedure, 22-5
 - Maintenance Commands for DCI, 22-10
 - SG/ECD Setup, 22-12

E

- ECD Audits, 14-72
- ECD Changes for the Dialogue Shell, 12-124
- ECD Forms, 22-12
 - chopt* Form, 22-15
 - classdef* Form, 22-17
 - dciopt* Form, 22-15
 - device* Form, 22-17
 - errthld* Form, 22-17
 - logdev* Form, 22-14
 - mdct* Form, 22-14
 - ucb* Form, 22-13
- ECD Owner Audit, 14-95
- ECD Recent Change Forms, 14-84
 - auderr* form for ECD 1, category 3, 14-88
 - auderr* form for ECD 1, category 4, 14-88
 - auderr* form for ECD 1, category 5, 14-89
 - auderr* form for ECD 2, category 3, 14-89
 - auderr* form for ECD 2, category 4, 14-90
 - auderr* form for ECD 2, category 5, 14-90
 - auderr* form for ECD 3, category 3, 14-91
 - auderr* form for ECD 3, category 4, 14-91
 - auderr* form for ECD 3, category 5, 14-92
 - audinst* form for ECD 1, 14-92
 - audinst* form for ECD 2, appecd, 14-93
 - audinst* form for ECD 2, ecd, 14-93
 - audinst* form for ECD 3, 14-94
 - audrec* form for ECD 1, 14-85

ECD Recent Change Forms (Continued)

audrec form for ECD 2, 14-86

audrec form for ECD 3, 14-87

ECD Requirements, 16-7

ECDOWN Recent Change Forms, 14-98

auderr form for ECDOWN 1, category 1, 14-99

auderr form for ECDOWN 1, category 2, 14-99

audinst form for ECDOWN 1, 14-100

audrec form for ECDOWN 1, 14-98

EGRASP Capabilities, 18-4

Ejecting the DAT Cassette, 19-38

EMERGENCY ACTION INTERFACE

BUFFER, 4-10

Emergency Fix, 21-10

Activation, 21-11

Backout, 21-11

SCCS and MCC Interface, 21-11

End Operation, 7-2

Enhanced SU Capabilities, 21-14

Enter Arguments Page 1 Screen, 11-12

Enter Arguments Page 2 Screen, 11-13

Enter BWM Number, 13-38

ENTER Process Terminates, 11-38

ENTER Request, 11-5

Equipment Configuration Database Audits,

ECD Recent Change Forms, 14-84

Interpreting AUD ECD Output Messages and Error Reports, 14-73

Requesting ECD Audits, 14-72

Responding to AUD ECD Error

Reports, 14-84

Equipment Configuration Database Owner Audit,

ECDOWN Recent Change Forms, 14-98

Interpreting AUD ECDOWN Output

Messages and Error Reports, 14-96

Requesting the ECDOWN Audit, 14-95

Responding to AUD ECDOWN Error Reports, 14-97

Equipment Locations, xxxviii

Error Analysis,

Description of Error Reports, 4-2

Error Report Analysis, 4-18

Error Conditions, 21-12

Error Interrupt Handler File, 4-47

Off-Line Hardware Faults, 4-52

On-Line Hardware Faults, 4-49

Software Faults, 4-53

Error Interrupt Handler Log File, 4-3

Error Log Files, 4-2

Error Report Analysis, 4-18

Automatic Postmortem Dump Analysis, 4-54

Error Interrupt Handler File, 4-47

Interpreting Memory Failures, 4-74

Memory Management Exceptions, 4-73

Procedure 1 Memory History File, 4-26

Procedure 2 On-Line Hardware DMA Detected, 4-40

Procedure 3 On-Line Hardware DMA Detected, 4-41

Procedure 4 On-Line Hardware DMA Detected, 4-42

Procedure 5 On-Line Hardware Fault, 4-43

Procedure 6 Off-Line Hardware

Failures, 4-45

Procedure 7 Software Fault, 4-46

Error Reporting, 10-12

Error Reporting and Recovery, 14-14

errthld Form, 22-17

Example 1, Read-Only Operation, 9-39

Example 10, Correction Operation, 9-42

Example 11, Stop Operation, 9-42

Example 2, Read-Only Operation, 9-40

Example 3, Read-Only Operation, 9-40

Example 4, Compare Operation, 9-40

Example 5, Compare Operation, 9-41

Example 6, Compare Operation, 9-41

Example 7, Compare Operation, 9-41

Example 8, Correction Operation, 9-41

Example 9, Correction Operation, 9-42

Examples of ECD Forms, 22-18

Examples of Performing OFLBOOT, 7-20

Performing OFLBOOT With One IOP Selected, 7-23

Performing OFLBOOT Without IOP Selected, 7-20

Execute **BACKOUT** Command, 11-44

Execute **COMMIT** Command, 11-41

Execute **ENTER** Command, 11-31

Execute Message 6 of the SOAK Section, 13-62

Execute **PROCEED** Command, 11-39

Execute **READLOG** Command, 11-48

Execute **REX** Command, 5-4

Execute **STOP** Command, 11-47

Execute the APPLY Section, 13-43

Execute the BKOUT Section, 13-51

Execute the PERM Section, 13-42, 13-67

Execute the **RESTORE** Command, 11-45

Execute the SOAK Section by Line Number, 13-60

Expansion Slots, 1-55
Extracting Error Count Information for
Audits, 14-19

F

Fan Alarms, 16-56
Fan Alarms and PSM Operation
Features, 16-58
FAULTY CU REGISTERS, 4-14
Faulty Unit Repair, 3-18
Field Support, 2-18
Field Test Set, 18-1
Field Updatable Processes, 13-3
Boot Image Component Processes, 13-3
Diagnostic Processes, 13-5
Killable Kernel Processes, 13-5
Killable Processes, 13-4
Nonkillable Processes, 13-3
Peripheral Microcode File, 13-5
Quasi-Killable Processes, 13-4
Field Update,
Broadcast Warning Messages, 13-6
Field Update Session Example, 13-34
Input Messages, 13-73
Messages File Execution and Display, 13-22
System Generation Database, 13-74
Update Methods and Processes, 13-1
Field Update Command Execution
Locations, 13-29
Message File Commands, 13-30
Scrolling Area Only Commands, 13-29
Scrolling Area/Poke Messages, 13-30
Field Update Page, 13-22
Field Update Procedures for ULARP and Its
Child Processes, 10-17
Field Update Session Example, 13-34
Call Up the Field Update Page, 13-36
Display Page 2 of the APPLY section, 13-40
Display the APPLY Section, 13-37, 13-39
Display the PERM Section, 13-41, 13-65
Display the SOAK Section, 13-59
Enter BWM Number, 13-38
Execute Message 6 of the SOAK
Section, 13-62
Execute the APPLY Section, 13-43
Execute the BKOUT Section, 13-51
Execute the PERM Section, 13-42, 13-67
Execute the SOAK Section by Line
Number, 13-60

Field Update Session Example (Continued)
Print the PERM Section, 13-66
Redisplay the APPLY Section, 13-53, 13-64
Reexecute the APPLY Section, 13-49, 13-54
Terminate the Field Update Session, 13-72
File Corruption Detection (SIV Only), 9-9
File Replacement, 13-2
File-Manager Audits, 14-100
FMGR Audit Error Categories, 14-106
FMGR Recent Change Forms, 14-108
Interpreting AUD FMGR Output Messages
and Error Reports, 14-103
Requesting FMGR Audits, 14-102
Responding to AUD FMGR Error
Reports, 14-106
File-System Block and Link Audits, 9-5
File-System Block Audit, 14-133
FSBLK Recent Change Forms, 14-138
Interpreting AUD FSBLK Output Messages
and Error Reports, 14-135
Requesting the FSBLK Audit, 14-134
Responding to AUD FSBLK Error
Reports, 14-137
File-System Body Form, 13-75
File-System Compaction Audit, 9-8, 14-155
FSCMPT Recent Change Forms, 14-163
How to Use the Compaction Audit, 14-158
Interpreting AUD FSCMPT Output
Messages and Error Reports, 14-160
Requesting the FSCMPT Audit, 14-159
Responding to AUD FSCMPT Error
Reports, 14-162
When to Use the Compaction Audit, 14-156
File-System Link Audit, 14-144
FSLINK Recent Change Forms, 14-149
Interpreting AUD FSLINK Output Messages
and Error Reports, 14-146
Requesting the FSLINK Audit, 14-145
Responding to AUD FSLINK Error
Reports, 14-148
File-System Maintenance,
Boot File Verification Through
OFLBOOT, 9-10
Checking File-System Resources, 9-1
Comparing and Correcting Disk Data, 9-39
Copying Files to Nonactive Disks, 9-37
Copying to Backup Partitions, 9-13
File Corruption Detection (SIV Only), 9-9
Initialize Disk Partition with Zeros, 9-36
Making a Backup Disk, 9-26
Making a Backup Tape Copy of a System
Disk, 9-34

File-System Maintenance (Continued)

Manipulating Files, 9-2

Procedure for Handling Disk Errors, 9-44

System Backup, 9-11

Using File-System Audits, 9-5

Firmware vs. Pumpcode, 19-2

FMGR Audit Error Categories, 14-106

FMGR Recent Change Forms, 14-108

auderr form for FMGR 1, category 0, 14-117

auderr form for FMGR 1, category 5, 14-117

auderr form for FMGR 2, category 0, 14-118

auderr form for FMGR 2, category 2, 14-118

auderr form for FMGR 2, category 3, 14-119

auderr form for FMGR 2, category 5, 14-119

auderr form for FMGR 3, category 2, 14-120

auderr form for FMGR 3, category 4, 14-120

auderr form for FMGR 4, category 0, 14-121

auderr form for FMGR 4, category 2, 14-121

auderr form for FMGR 5, category 1, 14-122

auderr form for FMGR 5, category 2, 14-122

auderr form for FMGR 6, category 0, 14-123

auderr form for FMGR 6, category 1, 14-123

auderr form for FMGR 6, category 2, 14-124

auderr form for FMGR 6, category 4, 14-124

auderr form for FMGR 6, category 5, 14-125

auderr form for FMGR 7, category 0, 14-125

auderr form for FMGR 7, category 1, 14-126

auderr form for FMGR 7, category 2, 14-126

auderr form for FMGR 7, category 3, 14-127

auderr form for FMGR 8, category 2, 14-127

auderr form for FMGR 9, category 5, 14-128

audinst form for FMGR 1, 14-128

audinst form for FMGR 2, 14-129

audinst form for FMGR 3, 14-129

audinst form for FMGR 4, 14-130

audinst form for FMGR 5, 14-130

audinst form for FMGR 6, 14-131

audinst form for FMGR 7, 14-131

audinst form for FMGR 8, 14-132

audinst form for FMGR 9, 14-132

audrec form for FMGR 1, 14-108

audrec form for FMGR 2, 14-109

audrec form for FMGR 3, 14-110

audrec form for FMGR 4, 14-111

audrec form for FMGR 5, 14-112

audrec form for FMGR 6, 14-113

audrec form for FMGR 7, 14-114

audrec form for FMGR 8, 14-115

audrec form for FMGR 9, 14-116

Force Disk Selection Options, 3-16

Forced Configuration Options, 3-9

Format Control Unit, 1-14

Frequency Groups, 14-13

FSBLK Recent Change Forms, 14-138

auderr form for FSBLK 1, category 1, 14-139

auderr form for FSBLK 1, category 3, 14-139

audinst form for FSBLK 1, */dev/bdb*, 14-142

audinst form for FSBLK 1, */dev/betc*, 14-142

audinst form for FSBLK 1,

/dev/broot, 14-143

audinst form for FSBLK 1, */dev/db*, 14-140

audinst form for FSBLK 1, */dev/etc*, 14-140

audinst form for FSBLK 1, */dev/root*, 14-141

audinst form for FSBLK 1, */dev/tmp*, 14-141

audrec form for FSBLK 1, 14-138

FSCMPT Recent Change Forms, 14-163

auderr form for FSCMPT 1, category

1, 14-164

auderr form for FSCMPT 1, category

3, 14-164

auderr form for FSCMPT 1, category

5, 14-165

audinst form for FSCMPT 1, */dev/db*, 14-165

audinst form for FSCMPT 1,

/dev/etc, 14-166

audinst form for FSCMPT 1,

/dev/root, 14-166

audrec form for FSCMPT 1, 14-163

FSLINK Recent Change Forms, 14-149

auderr form for FSLINK 1, category

1, 14-150

auderr form for FSLINK 1, category

2, 14-150

audinst form for FSLINK 1, */dev/bdb*, 14-153

audinst form for FSLINK 1,

/dev/betc, 14-153

audinst form for FSLINK 1,

/dev/broot, 14-154

audinst form for FSLINK 1, */dev/db*, 14-151

audinst form for FSLINK 1, */dev/etc*, 14-151

audinst form for FSLINK 1, */dev/root*, 14-152

audinst form for FSLINK 1, */dev/tmp*, 14-152

audrec form for FSLINK 1, 14-149

Function and Byte Replacement, 13-2

Functional Description, 12-84

Functional Messages, 16-26

Autonomous Report, 16-28

Command Response, 16-27

SCSD Error Handler, 16-29

Software Errors, 16-30

G

- General Guidelines, 17-1
- General Information, 16-1
- GENERAL REGISTERS, 4-13
- General Requirements, 16-51
- GRASP/EGRASP, 18-3
 - EGRASP Capabilities, 18-4
 - GRASP/EGRASP Capabilities, 18-3
- GRASP/EGRASP Capabilities, 18-3
- Growth and Degrowth Procedures, 16-64
 - Degrowth Procedure for Removing a UN33/UN933 Board, 16-70
 - Degrowth Procedure for Removing Scan Points on a UN33/UN933 Board, 16-68
 - Growth Procedure for Adding a UN33/UN933 Board, 16-65
 - Growth Procedure for Adding Scan Points on a UN33/UN933 Board, 16-66
 - Input/Output Processor Growth and Degrowth, 16-74
 - Preparation and Requirements for Growth and Degrowth Procedures, 16-64
 - Procedure for Adding Logical Groups, 16-70
 - Procedure for Deleting Logical Groups, 16-72
- Growth Procedure for Adding a UN33/UN933 Board, 16-65
- Growth Procedure for Adding Scan Points on a UN33/UN933 Board, 16-66
- Growth Requirements, 16-75
- Growth Unit, 1-27, 1-46
- Guidelines for MTC Diagnostics, 15-5
- Guidelines for Preventive Maintenance, 15-5
- Guidelines for Tape Transport Operation, 15-4

H

- Hardware Changes, Inform Diagnostic Program of, 6-23
- Hardware Identification,
 - 3B20D Model 1 Computer Hardware, 1-4
 - 3B20D Model 2 Computer Hardware, 1-18
 - 3B20D Model 3 Computer Hardware, 1-36
 - 3B21D Computer Processor Cabinet, 1-52
- HEADER File, 13-11

- Heading, 4-5
- Hexadecimal Notation, xxxvii
- How to Abort Commands or Messages, 12-129
- How to Abort or Cancel a Request, 6-21
- How to Access and Interpret the DFC Index Page, 12-88
- How to Access and Interpret the Disk File System Access Index Page, 12-105
- How to Access the Common and Application System Display Pages, 12-36
- How to Access the Common Processor Display Page, 12-44
 - Major and Minor State Display Field Values, 12-55
 - Page Description, 12-44
- How to Access the Control/Display Update Page, 12-57
- How to Access the Craft Field Maintenance Command Display Page, 12-63
- How to Access the DCI Status Page, 12-75
 - Major and Minor State Display Field Values for DCI, 12-80
- How to Access the DFC Performance Page, 12-101
- How to Access the DFC Status Page, 12-90
- How to Access the DFSA Performance Page, 12-117
- How to Access the DFSA Status Page, 12-107
- How to Access the Emergency Action Pages, 12-19
- How to Access the Field Update Page, 12-80
 - Functional Description, 12-84
 - Page Description, 12-81
- How to Access the Operating System Status Page, 12-59
- How to Access the Status Summary Area Page, 12-42
- How to Access the System Update Menu Page, 12-73
- How to Change Audit Forms, 14-43
- How to Clear Trouble Using the Diagnostic Listing, 6-38
- How to Clear Trouble Using the TLP Circuit Pack List, 6-36
- How to Comment on This Information Product, xl
- How to Execute CFTMTC Commands via Menu Numbers, 12-64
- How to Field Update the Input Message Catalog, 12-125
- How to Field Update the Output Message Database, 12-128

- How to Inform Diagnostic Program of Hardware Changes, 6-23
- How to Inhibit/Allow Automatic Maintenance Requests, 6-19
- How to Insert Audit Forms, 14-47
- How to Interpret Diagnostic Output and Clear Trouble, 6-25
 - Diagnostic Output Message, 6-25
 - Output Message Format, 6-26
- How to Interpret Output from TLP Option, 6-30
- How to Interpret the DFC Performance Page, 12-104
- How to Interpret the DFC Status Page, 12-96
- How to Interpret the DFSA Performance Page, 12-120
- How to Interpret the DFSA Status Page, 12-113
- How to Interpret the MTTY Display, 12-17
- How to Interpret the Operating System Status Page, 12-61
- How to Invoke RC/V, 14-39
- How to Invoke the CFTSHL Help Facility, 12-126
- How to Maintain System from User Operating Position, 12-4
- How to Modify Audit Records, 14-38
- How to Obtain Status of Requests, 6-20
- How to Operate the ROP in a High Humidity Environment, 12-11
- How to Reinitialize the Craft interface, 12-130
- How to Remove and Restore Units, 6-22
- How to Request Diagnostics, 6-8
 - Requesting Interactive Diagnostics, 6-12
 - Requesting Supplementary Phases, 6-10
 - Suggested Routine Test Procedures, 6-11
 - Using the **DGN** Command, 6-8
- How to Respond to a Denied Request, 6-16
- How to Respond to a Diagnostic Audit Message, 6-41
- How to Retire Alarms, 12-129
- How to Review Audit Forms, 14-42
- How to Set Options on Maintenance Terminals, 12-5
- How to Set Options on the Receive-Only Printer, 12-11
- How to Side Switch During OFLBOOT, 7-13
 - Stepping Through the **SW** Command, 7-15
 - Using the **SW** Command, 7-15
- How to Specify the Incore Size of the Message Spooler's Output Message Database in the ECD, 12-127
- How to Specify the Initial Display Pages for C/D Terminal in the ECD, 12-40

- How to Start OFLBOOT, 7-5
 - Stepping Through the **EXC** Command, 7-10
 - Using the **EXC** Command, 7-5
- How to Stop OFLBOOT, 7-12
 - Stepping Through the **STOP** Command, 7-13
 - Using the **STOP** Command, 7-12
- How to Switch Devices to Alternate MTTYC, 12-131
- How to Use the Compaction Audit, 14-158
- How to Use the Craft Shell and Dialogue Shell with the Input Message Catalog, 12-121
 - Capability Modes, 12-122
 - ECD Changes for the Dialogue Shell, 12-124
 - Input Message Catalog Database, 12-121
 - Pathname File, 12-124
 - Shell Configuration, 12-123
 - splinfo* Record in the ECD, 12-123
- How to Use the Flexible Input Message Formats, 12-126
- How to Use This Information Product, xxxiii

I

- Inform Diagnostic Program of Hardware Changes, 6-23
- Inhibit and Allow Automatic Maintenance Requests, 6-19
- Inhibit Options, 3-11
- Inhibit REX Command, 5-4
- Inhibiting Audits, 14-18
- Inhibiting the Routine Execution of Audits, 14-13
- Initialization and Recovery Analysis Guidelines, 3-6
- Initialization Causes, 3-3
- INITIALIZATION MESSAGE, 4-6
- Initialization Options, 3-12
- Initialization Procedures, 10-9
 - Boot Progress PRMs, 10-10
 - System Bootstrap, 10-9
- Initialize Disk Partition with Zeros, 9-36
- Input Message Catalog Database, 12-121
- Input Messages, 11-59, 13-73
- Input/Output Processor, B-5
- Input/Output Processor Growth and Degrowth, 16-74
 - Backout Procedures, 16-75
 - Growth Requirements, 16-75

Input/Output Processor Magnetic Tape Unit,
 Operating, Maintaining, and Diagnosing the
 Magnetic Tape System, 15-4
 Using the Magnetic Tape System, 15-3
 Writing a File from a Disk, 15-7
Input/Output Processor Unit, 1-11, 1-55
Install SUs, 21-25
Installation Procedure, 22-5
 Channel Board Installation Procedure, 22-5
 Software Installation Procedure, 22-5
Installing New Pumpcode, 19-5
Instance name, 14-17, 14-18, 14-20
Interactive Diagnostics, Requesting, 6-12
Interface Requirements, 16-30
 Client Process Message Interface, 16-31
 SCSDA ECD Requirements, 16-35
 SCSDH Message Interface, 16-30
Interfaces, 16-8
 SCSDH Message Interface, 16-8
Interpret Diagnostic Output and Clear
 Trouble, 6-25
Interpret Output from TLP Option, 6-30
Interpreting AUD CUMEM Output Messages
 and Error Reports, 14-52
Interpreting AUD CUSTAT Output Message
 and Error Reports, 14-59
Interpreting AUD ECD Output Messages and
 Error Reports, 14-73
Interpreting AUD ECDOWN Output Messages
 and Error Reports, 14-96
Interpreting AUD FMGR Output Messages and
 Error Reports, 14-103
Interpreting AUD FSBLK Output Messages and
 Error Reports, 14-135
Interpreting AUD FSCMPT Output Messages
 and Error Reports, 14-160
Interpreting AUD FSLINK Output Messages
 and Error Reports, 14-146
Interpreting AUD MMGR Output Messages and
 Error Reports, 14-169
Interpreting AUD MSGBUF Output Messages
 and Error Reports, 14-191
Interpreting AUD PMS Output Messages and
 Error Reports, 14-207
Interpreting AUD PROAD Output Messages
 and Error Reports, 14-218
Interpreting Memory Failures, 4-74
INTERRUPT STACK SAVED STATE, 4-14
IOP 2 Suggested Routine, B-10
IOP and SMD Type DFC Growth Frames, 1-14
IOP Basic Unit, 1-26, 1-45

IOP Diagnostic Hierarchy, A-5
IOP Magnetic Tape *logdev* Records (3B20D
 Computer Only), 20-2
IOP Suggested Routine Test Procedure, A-9
IP Organization, xxxiii
Isolating Emergency Action Page
 Problems, 3-45

K

Key Points About Menu Indicators, 13-25
Killable Kernel Processes, 13-5
Killable Processes, 13-4
KS-22921 Video Terminal, 3-39
KS-23554, L5 and L6 Video Terminal, 3-37
KS-23860, L800 Cleaning Cassette, 19-20
KS-23909, L10 9-Track Tape, 19-21
 Loading the Tape Reel (Auto Load), 19-30
 Manually Loading, 19-31
KS-23909, L10 9-Track Tape Reel,
 Removing Tape During a Power
 Failure, 19-33
 Removing Tape with Power On, 19-32
KS-23909, L21 9-Track Tape, 19-24
KS-23909, L21 9-Track Tape Reel,
 Loading, 19-34
 Manually Rewind, 19-36
 Unloading, 19-36
KS-23996, L1 and L5 Video Terminal, 3-38
KS-24367, L1/Comcode 407771260 DAT Drive
 Status Indicators, 19-40
KS-24367, L1/Comcode 407771260 DAT
 Drives, 19-19, 19-37

L

Limitations of Disk Compare and Correct
 Commands, 9-43
Listing Logical Groups Defined in ECD, 16-17
Load Disk from Tape Procedure, 3-67
Load SUs from SCANS or Tape, 21-40
Loading, 19-34
Loading Disk from Tape, 3-85
Loading the DAT Cartridge
 Tape/Cassette, 19-37
Loading the Tape Reel (Auto Load), 19-30

Loading/Unloading, 19-30
 9-Track Tape Loading/Unloading, 19-30
 DAT Loading/Unloading, 19-37
 Ejecting the DAT Cassette, 19-38
 KS-24367, L1/Comcode 407771260 DAT
 Drives, 19-37
 Loading the DAT Cartridge
 Tape/Cassette, 19-37
 NCR006-3300608/Comcode 40754524 DAT
 Drives, 19-37
 NCR006-3503341 DAT Drives, 19-37
Locating Pins, 6-57
Log File Review,
 Core Dump Administration, 8-5
 Using Log Files, 8-3
logdev Form, 22-14
Logical Addressing, 16-14

M

Magnetic Tape Special Device Filenames,
 IOP Magnetic Tape *logdev* Records (3B20D
 Computer Only), 20-2
 SCSI *logdev* Records, 20-5
 SCSI/IOP Backward Compatible *logdev*
 Records, 20-4
 Tape Device *logdev* Records, 20-1
Magnetic Tape Unit, 1-14, 1-29
Main Store and Main Store Update, 1-54
Main Store Failure Data, 6-30, A-23, B-26
 How to Clear Trouble Using the Diagnostic
 Listing, 6-38
 How to Clear Trouble Using the TLP Circuit
 Pack List, 6-36
 How to Interpret Output from TLP
 Option, 6-30
 How to Respond to a Diagnostic Audit
 Message, 6-41
MAIN STORE REGISTERS, 4-16
Main Store Unit, 1-9
Maintaining Software Integrity,
 Adding to the Defect Table of a SCSI Disk
 Drive, 10-23
 Checking Operating System
 Resources, 10-1
 Maintenance Guide for ULARP, 10-7
 Manipulating Processes, 10-6
 Using CRON, 10-18
 Using Operating System Audits, 10-4

Maintaining Software Integrity (Continued)
 Writing SMD Disk Drive Defect
 Tables, 10-19
Maintenance Commands for DCI, 22-10
 Diagnosing the DCI Board, 22-11
 Removing DCI from Service, 22-10
 Restoring DCI to Service, 22-10
 Terminating the Running DCI Driver, 22-10
Maintenance Guide for ULARP, 10-7
 Database Modification Procedures, 10-9
 Error Reporting, 10-12
 Field Update Procedures for ULARP and Its
 Child Processes, 10-17
 Initialization Procedures, 10-9
 ULARP Records in the ECD, 10-7
 Use of the **OP:ULARP** Command, 10-15
Maintenance Input Request Administrator, 6-3
Maintenance Interface, 12-1
Maintenance Plan,
 Automatic Recovery Actions, 2-3
 Documentation, 2-17
 Field Support, 2-18
 Manual Recovery Actions, 2-4
 Preventive Maintenance, 2-4
 Responding to Alarms, 2-6
 Site Maintenance Plan, 2-2
 Switching Control Center Interface, 2-18
 Training, 2-18
 Using Automatically Generated Data for
 Troubleshooting, 2-17
Maintenance Procedures, 12-4
 How to Abort Commands or
 Messages, 12-129
 How to Access and Interpret the DFC Index
 Page, 12-88
 How to Access and Interpret the Disk File
 System Access Index Page, 12-105
 How to Access the Common and
 Application System Display Pages, 12-36
 How to Access the Common Processor
 Display Page, 12-44
 How to Access the Control/Display Update
 Page, 12-57
 How to Access the Craft Field Maintenance
 Command Display Page, 12-63
 How to Access the DCI Status Page, 12-75
 How to Access the DFC Performance
 Page, 12-101
 How to Access the DFC Status Page, 12-90
 How to Access the DFSA Performance
 Page, 12-117

Maintenance Procedures (Continued)

- How to Access the DFSA Status
Page, 12-107
- How to Access the Emergency Action
Pages, 12-19
- How to Access the Field Update
Page, 12-80
- How to Access the Operating System Status
Page, 12-59
- How to Access the Status Summary Area
Page, 12-42
- How to Access the System Update Menu
Page, 12-73
- How to Execute CFTMTC Commands via
Menu Numbers, 12-64
- How to Field Update the Input Message
Catalog, 12-125
- How to Field Update the Output Message
Database, 12-128
- How to Interpret the DFC Performance
Page, 12-104
- How to Interpret the DFC Status
Page, 12-96
- How to Interpret the DFSA Performance
Page, 12-120
- How to Interpret the DFSA Status
Page, 12-113
- How to Interpret the MTTY Display, 12-17
- How to Interpret the Operating System
Status Page, 12-61
- How to Invoke the CFTSHL Help
Facility, 12-126
- How to Maintain System from User
Operating Position, 12-4
- How to Operate the ROP in a High Humidity
Environment, 12-11
- How to Reinitialize the Craft
interface, 12-130
- How to Retire Alarms, 12-129
- How to Set Options on Maintenance
Terminals, 12-5
- How to Set Options on the Receive-Only
Printer, 12-11
- How to Specify the Incore Size of the
Message Spooler's Output Message
Database in the ECD, 12-127
- How to Specify the Initial Display Pages for
C/D Terminal in the ECD, 12-40
- How to Switch Devices to Alternate
MTTYC, 12-131
- How to Use the Craft Shell and Dialogue
Shell with the Input Message
Catalog, 12-121

Maintenance Procedures (Continued)

- How to Use the Flexible Input Message
Formats, 12-126
- Maintenance Requests, Automatic, Inhibit and
Allow, 6-19
- Maintenance Tools,
 - Field Test Set, 18-1
 - GRASP/EGRASP, 18-3
 - Software Debugging Tools, 18-4
- Major and Minor State Display Field
Values, 12-55
- Major and Minor State Display Field Values for
DCI, 12-80
- Make SUs Official, 21-29
- Making a Backup Disk, 9-26
 - Copying to a Spare Disk Drive, 9-26
 - Using a Warm Spare Disk Drive, 9-31
- Making a Backup Disk, Method 1, Procedure
A, 9-26
- Making a Backup Disk, Method 1, Procedure
B, 9-29
- Making a Backup Tape Copy of a System
Disk, 9-34
- Mandatory Customer Information, xl
- Manipulating Files, 9-2
- Manipulating Processes, 10-6
- Manual Recovery Actions, 2-4
- Manual Stop and Start of SU Expansion, 21-34
- Manually Loading, 19-31
- Manually Rewind, 19-36
- MAS-IO-DFC Unit, 1-24, 1-43
- mdct* Form, 22-14
- Memory History Log File, 4-3
- Memory Management Exceptions, 4-73
- Memory-Manager Audits, 14-167
 - Interpreting AUD MMGR Output Messages
and Error Reports, 14-169
 - MMGR Recent Change Forms, 14-174
 - Requesting the MMGR Audits, 14-168
 - Responding to MMGR Audit
Reports, 14-173
- Menu Indicators, 13-24
 - Action Indicators, 13-25
 - Allowable Scope/Action
Combinations, 13-27
 - Configuration Indicators, 13-24
 - Field Update Command Execution
Locations, 13-29
 - Key Points About Menu Indicators, 13-25
 - Scope Indicators, 13-24
- Message File Commands, 13-30

Message-Buffer Audits, 14-190
 Interpreting AUD MSGBUF Output
 Messages and Error Reports, 14-191
MSGBUF Recent Change Forms, 14-198
 Requesting MSGBUF Audits, 14-190
 Responding to AUD MSGBUF Error Reports, 14-195
Messages File Execution and Display, 13-22
Messages Files Execution and Display,
 Field Update Page, 13-22
 Menu Indicators, 13-24
 Troubleshooting Hints, 13-31
 Update Display Command, 13-31
 Verification of Proper Installation, 13-32
MHD Control Panel, 1-12
MIRA,
 Canceled and Denied Requests, 6-6
 Concurrent Maintenance Requests, 6-4
 Diagnostic Commands, 6-4
 MIRA Initialization, 6-3
 Priority Scheme, 6-6
 Scheduling Requests, 6-5
MIRA Initialization, 6-3
MMGR Recent Change Forms, 14-174
 auderr form for MMGR 1, category 1, 14-178
 auderr form for MMGR 1, category 2, 14-179
 auderr form for MMGR 1, category 3, 14-179
 auderr form for MMGR 1, category 4, 14-180
 auderr form for MMGR 1, category 5, 14-180
 auderr form for MMGR 2, category 1, 14-181
 auderr form for MMGR 2, category 2, 14-181
 auderr form for MMGR 2, category 3, 14-182
 auderr form for MMGR 2, category 4, 14-182
 auderr form for MMGR 3, category 1, 14-183
 auderr form for MMGR 3, category 2, 14-183
 auderr form for MMGR 3, category 3, 14-184
 auderr form for MMGR 3, category 4, 14-184
 auderr form for MMGR 4, category 3, 14-185
 auderr form for MMGR 5, category 3, 14-185
 auderr form for MMGR 9, category 0, 14-186
 audinst form for MMGR 1, 14-186
 audinst form for MMGR 2, 14-187
 audinst form for MMGR 3, 14-187
 audinst form for MMGR 4, 14-188
 audinst form for MMGR 5, 14-188
 audinst form for MMGR 9, 14-189
 audrec form for MMGR 1, 14-174
 audrec form for MMGR 2, 14-175
 audrec form for MMGR 3, 14-175
 audrec form for MMGR 4, 14-176
 audrec form for MMGR 5, 14-177

MMGR Recent Change Forms (Continued)
 audrec form for MMGR 9, 14-177
Mount Tape for Disk 770, 11-32
Mount Tape for Disk 772, 11-34
Mount Tape for Disk 775, 11-36
Moving Head Disk Frame, 1-12
 300-MB Disk, 1-12
 Disk Inverter, 1-12
 MHD Control Panel, 1-12
Moving Head Disk Units, 1-33
MSGBUF Recent Change Forms, 14-198
 auderr form for MSGBUF 1, category 0, 14-200
 auderr form for MSGBUF 1, category 1, 14-200
 auderr form for MSGBUF 1, category 2, 14-201
 auderr form for MSGBUF 1, category 3, 14-201
 auderr form for MSGBUF 1, category 5, 14-202
 auderr form for MSGBUF 2, category 0, 14-202
 auderr form for MSGBUF 2, category 1, 14-203
 auderr form for MSGBUF 2, category 2, 14-203
 auderr form for MSGBUF 2, category 3, 14-204
 audinst form for MSGBUF 1, 14-204
 audinst form for MSGBUF 2, 14-205
 audrec form for MSGBUF 1, 14-198
 audrec form for MSGBUF 2, 14-199
MSG File, 13-12
MTTY Suspend, 3-37
 KS-22921 Video Terminal, 3-39
 KS-23554, L5 and L6 Video Terminal, 3-37
 KS-23996, L1 and L5 Video Terminal, 3-38
 Teletype Model 4424 Terminal, 3-40

N

NCR006-3300608/Comcode 40745243 DAT Drives, 19-19
NCR006-3300608/Comcode 40754524 DAT Drives, 19-37
NCR006-3300608/Comcode 407545243 DAT Drive Status Indicators, 19-40
NCR006-3503341 DAT Drives, 19-19, 19-37

NCR006-3503341 SCSI DAT Drive Status Indicators, 19-40
Non-MTTY Suspend, 3-36
Nondeferrable Maintenance Function, 16-24
 Operational Functions, 16-26
 Remove Nondeferrable Maintenance Function, 16-25
 Restore SCSD PC, 16-24
Nonkillable Processes, 13-3

O

Obtain Status of Requests, 6-20
Obtain SU Status Report, 21-35
Off-Line Boot,
 Checking OFLBOOT Results, 7-17
 End Operation, 7-2
 Examples of Performing OFLBOOT, 7-20
 Performing OFLBOOT, 7-4
 Side Switch Operation, 7-2
 System Configuration, 7-1
 Using OFLBOOT, 7-3
Off-Line Disk Method, 11-4
Off-Line Hardware Faults, 4-52
OFF-LINE REGISTERS, 4-15
Office Alarms, 16-6
On-Line Hardware Faults, 4-49
One or More Terminals Faulty, 3-33
OP AUD Output Message, 14-23
OP AUDERR Output Message, 14-23
OP FNAME Output Message, 14-24
Operating, Maintaining, and Diagnosing the Magnetic Tape System, 15-4
 Guidelines for MTC Diagnostics, 15-5
 Guidelines for Preventive Maintenance, 15-5
 Guidelines for Tape Transport Operation, 15-4
Operating-System Audits, 14-50
 Control Unit Hardware Status Audit, 14-58
 Control Unit Memory Comparison Audit, 14-51
 ECD Audits, 14-72
 ECD Owner Audit, 14-95
 File-Manager Audits, 14-100
 File-System Block Audit, 14-133
 File-System Compaction Audit, 14-155
 File-System Link Audit, 14-144
 Memory-Manager Audits, 14-167
 Message-Buffer Audits, 14-190

Operating-System Audits (Continued)
 Plant Measurements System Audits, 14-206
 Process-Administration Audit, 14-218
Operational Functions, 16-26
Operations for Comparing and Correcting Disk Data, 9-39
 Compare Operation, 9-40
 Correction Operation, 9-41
 Read-Only Operation, 9-39
 Stop Operation, 9-42
Other Update Package States, 13-20
Output Message Format, 6-26

P

Page 1940 Commands, 21-46
Page 1940 Description, 21-44
Page 1940 Example Procedure, 21-48
Page 1940 Fields, 21-45
Page 1940 Easy SU Installation, 21-44
 Page 1940 Commands, 21-46
 Page 1940 Description, 21-44
 Page 1940 Example Procedure, 21-48
 Page 1940 Fields, 21-45
Page 1941 Commands, 21-51
Page 1941 Description, 21-49
Page 1941 Fields, 21-50
Page 1941 SU Automation Scheduling, 21-49
 Page 1941 Commands, 21-51
 Page 1941 Description, 21-49
 Page 1941 Fields, 21-50
Page 1942 Commands, 21-54
Page 1942 Description, 21-53
Page 1942 SU Automation Office Profile, 21-53
 Page 1942 Commands, 21-54
 Page 1942 Description, 21-53
Page 1943 Commands, 21-57
Page 1943 Description, 21-56
Page 1943 SU Automation Healthcheck, 21-56
 Page 1943 Commands, 21-57
 Page 1943 Description, 21-56
Page 1950 Commands, 21-59
Page 1950 Description, 21-58
Page 1950 SU Maintenance, 21-58
 Page 1950 Commands, 21-59
 Page 1950 Description, 21-58
Page 1960 Commands, 21-63
Page 1960 Description, 21-61
Page 1960 SU Installation, 21-61

- Page 1960 SU Installation (Continued)
 - Page 1960 Commands, 21-63
 - Page 1960 Description, 21-61
- Page Description, 12-44, 12-81
- Pathname File, 12-124
- Performing Diagnostics, 6-8
 - How to Abort or Cancel a Request, 6-21
 - How to Inform Diagnostic Program of Hardware Changes, 6-23
 - How to Inhibit/Allow Automatic Maintenance Requests, 6-19
 - How to Interpret Diagnostic Output and Clear Trouble, 6-25
 - How to Obtain Status of Requests, 6-20
 - How to Remove and Restore Units, 6-22
 - How to Request Diagnostics, 6-8
 - How to Respond to a Denied Request, 6-16
 - Main Store Failure Data, 6-30
- Performing OFLBOOT, 7-4
 - How to Side Switch During OFLBOOT, 7-13
 - How to Start OFLBOOT, 7-5
 - How to Stop OFLBOOT, 7-12
- Performing OFLBOOT With One IOP Selected, 7-23
- Performing OFLBOOT Without IOP Selected, 7-20
- Peripheral Control Frame, 1-10
 - Input/Output Processor Unit, 1-11
 - Storage Module Drive Disk File Controller Unit, 1-11
- Peripheral Growth Cabinet, 1-56
- Peripheral Microcode File, 13-5
- Pin Location, 6-57
- Pin Set, Backplane, 6-58
- Plant Measurements System Audits, 14-206
 - Interpreting AUD PMS Output Messages and Error Reports, 14-207
 - PMS Recent Change Forms, 14-211
 - Requesting PMS Audits, 14-206
 - Responding to AUD PMS Error Reports, 14-210
- PMS Recent Change Forms, 14-211
 - auderr* form for PMS 1, category 1, 14-213
 - auderr* form for PMS 1, category 2, 14-213
 - auderr* form for PMS 1, category 3, 14-214
 - auderr* form for PMS 1, category 4, 14-214
 - auderr* form for PMS 1, category 5, 14-215
 - auderr* form for PMS 2, category 0, 14-215
 - auderr* form for PMS 2, category 3, 14-216
 - auderr* form for PMS 2, category 5, 14-216
 - audinst* form for PMS 1, 14-217
- PMS Recent Change Forms (Continued)
 - audinst* form for PMS 2, 14-217
 - audrec* form for PMS 1, 14-211
 - audrec* form for PMS 2, 14-212
- Port Switch, 3-42
- Post Initialization Analysis, 3-4
- Post Recovery Procedure, 3-32
- Power Cabinet (Optional), 1-51
- Power Distribution Frame, 1-16, 1-33
- Power Distribution Unit, 1-16, 1-31, 1-49
- Power Down Options, 3-13
- Power Interrupt (Fiber Optic Feature Only), 3-34
- Power Switch Monitor, 16-39, 16-57
 - PSM Initialization, 16-46
 - PSM Interfaces, 16-46
 - PSM Message Structures, 16-48
 - PSM Messages, 16-47
- Power Units, 1-9
- Preparation and Requirements for Growth and Degrowth Procedures, 16-64
- Prepare DFCs and MHDs for Recovery Procedure, 3-63
- Preventive Maintenance, 2-4
 - Routine Hardware Maintenance, 2-4
 - Software Exercises, 2-5
- Print the PERM Section, 13-66
- Priority Scheme, 6-6
- PROAD Recent Change Forms, 14-222
 - auderr* form for PROAD 1, category 1, 14-223
 - auderr* form for PROAD 1, category 2, 14-223
 - audinst* form for PROAD 1, 14-224
 - audrec* form for PROAD 1, 14-222
- Procedure 1 Memory History File, 4-26
- Procedure 2 On-Line Hardware DMA Detected, 4-40
- Procedure 3 On-Line Hardware DMA Detected, 4-41
- Procedure 4 On-Line Hardware DMA Detected, 4-42
- Procedure 5 On-Line Hardware Fault, 4-43
- Procedure 6 Off-Line Hardware Failures, 4-45
- Procedure 7 Software Fault, 4-46
- Procedure for Adding Logical Groups, 16-70
- Procedure for Deleting Logical Groups, 16-72
- Procedure for Handling Disk Errors, 9-44
- Procedure to Respond to Alarms, 2-15
- Procedures for Modifying Audit Records, 14-38
 - How to Change Audit Forms, 14-43
 - How to Insert Audit Forms, 14-47

Procedures for Modifying Audit
Records (Continued)
 How to Invoke RC/V, 14-39
 How to Modify Audit Records, 14-38
 How to Review Audit Forms, 14-42
PROCEED Process Terminates, 11-40
PROCEED Request, 11-6
Process Form, 13-78
Process-Administration Audit, 14-218
 Interpreting AUD PROAD Output Messages
 and Error Reports, 14-218
 PROAD Recent Change Forms, 14-222
 Requesting the PROAD Audit, 14-218
 Responding to AUD PROAD Error
 Reports, 14-221
Processor Cabinet, 1-36
 Central Processor Unit, 1-42
 Growth Unit, 1-46
 IOP Basic Unit, 1-45
 MAS-IO-DFC Unit, 1-43
Processor Control Frame, 1-18
 Central Processor Unit, 1-23
 Growth Unit, 1-27
 IOP Basic Unit, 1-26
 MAS-IO-DFC Unit, 1-24
Processor Recovery Messages, 4-16, 6-59,
 14-24, A-29, B-29
PSM Initialization, 16-46
PSM Interfaces, 16-46
PSM Message Structures, 16-48
PSM Messages, 16-47
Purpose of REX, 5-1

Q

Quasi-Killable Processes, 13-4

R

Read-Only Operation, 9-39
READLOG Utility, 11-7
REAL-TIME CLOCK, 4-16
Reason for Issue, xxxii
Receive-Only Printer Suspend, 3-42
Reclaim Patch Space, 21-12
Reclaim Space in SU Storage Directory, 21-12

Recognizing an Initialization, 3-3
Recommended Frequency of Selected
 Supplementary Phases, A-11, B-13
Recovering from DIOP, 3-59
 Boot Procedure, 3-74
 Load Disk from Tape Procedure, 3-67
 Prepare DFCs and MHDs for Recovery
 Procedure, 3-63
 Recovery Decision Procedure, 3-70
 SCSI DFC Repair Guidelines, 3-81
 SCSI MHD Repair Guidelines, 3-84
 SMD DFC Repair Guidelines, 3-77
 SMD MHD Repair Guidelines, 3-79
 Verify DIOP Mode in Effect Procedure, 3-60
Recovering the System Manually, 3-8
 Recovery Options, 3-9
 Recovery Procedure, 3-19
Recovery,
 Isolating Emergency Action Page
 Problems, 3-45
 Loading Disk from Tape, 3-85
 Recovering from DIOP, 3-59
 Recovering the System Manually, 3-8
 Recovery Using Software Backup Disk and
 Alternate Boot Disk Options, 3-51
 Responding to a Terminal Suspend, 3-33
 Responding to Automatic Recovery
 Actions, 3-1
 Using Processor Recovery Messages, 3-33
Recovery Decision Procedure, 3-70
Recovery from Backups Procedure, 3-25
Recovery Options, 3-9
 Bootstrap Options, 3-13
 DIOP Options, 3-18
 Force Disk Selection Options, 3-16
 Forced Configuration Options, 3-9
 Inhibit Options, 3-11
 Initialization Options, 3-12
 Power Down Options, 3-13
 Severity of Recovery Options, 3-8
 Software Backup Disk Options and
 Alternate Boot Disk, 3-15
Recovery Procedure, 3-19
 Application Recovery Procedure, 3-31
 Post Recovery Procedure, 3-32
 Recovery from Backups Procedure, 3-25
Recovery Using Alternate Boot Disk, 3-56
Recovery Using Software Backup Disk and
 Alternate Boot Disk Options, 3-51
 Recovery Using Alternate Boot Disk, 3-56
 SBD Switching Identities Procedure, 3-55

- Redisplay the APPLY Section, 13-53, 13-64
- Reexecute the APPLY Section, 13-49, 13-54
- Regular Software Update, 21-2
 - Activation, 21-9
 - Backout, 21-10
 - SCCS and MCC Interface, 21-6
 - SU Format, 21-4
 - SU Reception from the CSCANS, 21-8
 - SU Reception from the SCANS, 21-7
 - Verification, 21-9
- Related Information Products, xxxix
- Remove and Restore Units, 6-22
- Remove Nondeferrable Maintenance Function, 16-25
- Remove UCL Argument from Specified Command Line Argument, 11-51
- Removing DCI from Service, 22-10
- Removing from or Inserting into the System, 19-11
- Removing from or Restoring to Service, 19-10
- Removing Tape During a Power Failure, 19-33
- Removing Tape with Power On, 19-32
- Removing the Installed Package from the Workstation, 22-9
- Reporting Status of Audits, 14-19
- REPT AUDSTAT Output Message, 14-26
- REPT SIMCHK Output Message, 14-26
- Request Summary/Long Summary of SUs, 21-36
- Request, Abort or Cancel, 6-21
- REQUESTED INITIALIZATION PARAMETERS, 4-9
- Requesting Audits, 14-17
- Requesting Diagnostics, 6-8
- Requesting ECD Audits, 14-72
- Requesting FMGR Audits, 14-102
- Requesting Interactive Diagnostics, 6-12
- Requesting MSGBUF Audits, 14-190
- Requesting PMS Audits, 14-206
- Requesting Supplementary Phases, 6-10
- Requesting the CUMEM Audit, 14-51
- Requesting the CUSTAT Audit, 14-58
- Requesting the ECDOWN Audit, 14-95
- Requesting the FSBLK Audit, 14-134
- Requesting the FSCMPT Audit, 14-159
- Requesting the FSLINK Audit, 14-145
- Requesting the MMGR Audits, 14-168
- Requesting the PROAD Audit, 14-218
- Respond to Diagnostic Audit Message, 6-41
- Responding to a Denied Request, 6-16
- Responding to a Terminal Suspend, 3-33

- Responding to a Terminal Suspend (Continued)
 - All Terminals Do Not Function, 3-34
 - Craft Initialization, 3-42
 - Creating a Dump Tape, 3-43
 - MTTY Suspend, 3-37
 - Non-MTTY Suspend, 3-36
 - One or More Terminals Faulty, 3-33
 - Port Switch, 3-42
 - Power Interrupt (Fiber Optic Feature Only), 3-34
 - Receive-Only Printer Suspend, 3-42
- Responding to Alarms, 2-6
- Responding to AUD CUMEM Error Reports, 14-54
- Responding to AUD CUSTAT Error Reports, 14-66
- Responding to AUD ECD Error Reports, 14-84
- Responding to AUD ECDOWN Error Reports, 14-97
- Responding to AUD FMGR Error Reports, 14-106
- Responding to AUD FSBLK Error Reports, 14-137
- Responding to AUD FSCMPT Error Reports, 14-162
- Responding to AUD FSLINK Error Reports, 14-148
- Responding to AUD MSGBUF Error Reports, 14-195
- Responding to AUD PMS Error Reports, 14-210
- Responding to AUD PROAD Error Reports, 14-221
- Responding to Automatic Recovery Actions, 3-1
 - Initialization and Recovery Analysis Guidelines, 3-6
 - Initialization Causes, 3-3
 - Post Initialization Analysis, 3-4
 - Recognizing an Initialization, 3-3
- Responding to MMGR Audit Reports, 14-173
- RESTORE** Request, 11-6
- Restore SCSD PC, 16-24
- Restore Units, Remove and, 6-22
- Restoring DCI to Service, 22-10
- RETRO Argument, 11-53
- Return to System Update Menu Page, 11-30
- REX Inhibit Implementations, 5-9
 - REX Inhibits, 5-9, 5-10
- REX Inhibit Recommendations, 5-8
- REX Inhibits, 5-7, 5-9, 5-10
- REX Manual Commands, 5-3

REX Manual Commands (Continued)

- Allow REX Command, 5-5
- Display REX Inhibits, 5-5
- Execute **REX** Command, 5-4
- Inhibit REX Command, 5-4
- Stop REX Command, 5-4

REX Operation, 5-1

REX Scheduling, 5-2

REX Scheduling Recommendations, 5-2

REX Summary, 5-12

Routine Exercise Procedures,

- Purpose of REX, 5-1
- REX Inhibit Implementations, 5-9
- REX Inhibit Recommendations, 5-8
- REX Inhibits, 5-7
- REX Manual Commands, 5-3
- REX Operation, 5-1
- REX Scheduling, 5-2
- REX Scheduling Recommendations, 5-2
- REX Summary, 5-12

Routine Hardware Maintenance, 2-4

Routine Operations and Maintenance

Procedures, 21-24

- Back Out Last Official SUs, 21-33
- Back Out SUs, 21-32
- Check/Make Official Temporary SUs, 21-26
- Install SUs, 21-25
- Load SUs from SCANS or Tape, 21-40
- Make SUs Official, 21-29
- Manual Stop and Start of SU
 - Expansion, 21-34
- Obtain SU Status Report, 21-35
- Request Summary/Long Summary of SUs, 21-36
- Schedule Making SUs Official, 21-24

S

Safety Labels, xxxviii

SBD Switching Identities Procedure, 3-55

SC Status and SD Point Operation,

- Capabilities and User Requirements, 16-18
- ECD Requirements, 16-7
- General Information, 16-1
- Growth and Degrowth Procedures, 16-64
- Interfaces, 16-8
- SCSD Commands, 16-62
- SCSD ECD Records, 16-10
- SCSDA Requirements, 16-4

SC Status and SD Point

Operation (Continued)

- Verifying SCSD Operation, 16-76

Scan Report Dispatching, 16-5

SCANS File, 13-9

SCANS, Load SUs from, 21-40

SCCS and MCC Interface, 21-6, 21-11

Schedule Making SUs Official, 21-24

Scheduling Requests, 6-5

Scheduling Routine Audits, 14-12

Scope Indicators, 13-24

Scrolling Area Only Commands, 13-29

Scrolling Area/Poke Messages, 13-30

SCSD Commands, 16-62

- Access to Scan and Distribute Points, 16-62

SCSD ECD Records, 16-10

- Definition of Logical Groups, 16-11

- Listing Logical Groups Defined in ECD, 16-17

- Logical Addressing, 16-14

- Storing Data in the SCSD Option Block, 16-15

SCSD Error Handler, 16-29

SCSD Features and Limitations, 16-23

- Functional Messages, 16-26

- Nondeferrable Maintenance Function, 16-24

- SCSDH Interfaces, 16-24

SCSD Handler, 16-23

SCSD Point Assignments, 16-52

SCSD Responses, 16-19

SCSD System and Alarm Requirements and Assignments, 16-51

- General Requirements, 16-51

- SCSD Point Assignments, 16-52

SCSDA ECD Requirements, 16-35

SCSDA Functions, 16-18

SCSDA Requirements, 16-4

- Craft TTY Interface, 16-6

- Device Translation, 16-5

- Distribute Operation Requests, 16-6

- Office Alarms, 16-6

- Scan Report Dispatching, 16-5

SCSDH Interfaces, 16-24

SCSDH Message Interface, 16-8, 16-30

SCSI DFC Repair Guidelines, 3-81

- 3B20D Computer, 3-81

- 3B21D Computer, 3-83

SCSI DFC Units, 1-16, 1-31, 1-49

SCSI Disk Cabinet (Optional), 1-14, 1-49

- Cooling Unit, 1-16, 1-31, 1-49

- Power Distribution Unit, 1-16, 1-31, 1-49

- SCSI Disk Cabinet (Optional) (Continued)
 - SCSI DFC Units, 1-16, 1-31, 1-49
 - SCSI DUP Units, 1-16, 1-31, 1-50
 - Tape Drive, 1-49
- SCSI Disk Cabinet (Optional, Found in Release 6 Only), 1-31
- SCSI Disk Drives, 19-7
- SCSI DUP Units, 1-16, 1-31, 1-50
- SCSI *logdev* Records, 20-5
- SCSI MHD Repair Guidelines, 3-84
- SCSI Peripheral System,
 - Firmware vs. Pumpcode, 19-2
 - Installing New Pumpcode, 19-5
 - SCSI Disk Drives, 19-7
 - SCSI Tape Drives, 19-9
 - Verify the Integrity of Pumpcode, 19-4
- SCSI Tape Drives, 19-9
 - Cleaning SCSI Tape Drives, 19-13
 - DAT Drive Cartridge Tapes and Status Indicators, 19-39
 - Diagnosing SCSI Tape Drives, 19-10
 - Drive Identification, 19-9
 - Loading/Unloading, 19-30
 - Removing from or Inserting into the System, 19-11
 - Removing from or Restoring to Service, 19-10
 - Setting Device ID, 19-21
 - Tape Capacity/Tape Density, 19-28
 - Write Protection, 19-27
- SCSI/IOP Backward Compatible *logdev* Records, 20-4
- Setting Device ID, 19-21
- Setting Device Identification,
 - DAT Drives, 19-25
 - KS-23909, L10 9-Track Tape, 19-21
 - KS-23909, L21 9-Track Tape, 19-24
- Severity of Recovery Options, 3-8
- SG Forms, 22-12
- SG/ECD Setup, 22-12
 - ECD Forms, 22-12
 - Examples of ECD Forms, 22-18
 - SG Forms, 22-12
- Shell Configuration, 12-123
- Side Switch Operation, 7-2
- Signal Designations, xxxviii
- Site Maintenance Plan, 2-2
- SMD DFC Repair Guidelines, 3-77
- SMD MHD Repair Guidelines, 3-79
- Software Backup Disk Options and Alternate Boot Disk, 3-15

- Software Debugging Tools, 18-4
- Software Errors, 16-30
- Software Exercises, 2-5
- Software Faults, 4-53
- Software Installation Procedure, 22-5
 - 3B20D/3B21D Computer Product Installation, 22-5
 - Removing the Installed Package from the Workstation, 22-9
 - Sun* System Product Installation, 22-5
- Software Integrity Records, 14-6
- Software Update,
 - Enhanced SU Capabilities, 21-14
 - Error Conditions, 21-12
 - Page 1940 Easy SU Installation, 21-44
 - Page 1941 SU Automation Scheduling, 21-49
 - Page 1942 SU Automation Office Profile, 21-53
 - Page 1943 SU Automation Healthcheck, 21-56
 - Page 1950 SU Maintenance, 21-58
 - Page 1960 SU Installation, 21-61
 - Routine Operations and Maintenance Procedures, 21-24
 - Software Update Types, 21-2
 - Space Reclamation, 21-12
 - SU Installation, 21-16
 - SU Maintenance, 21-20
- Software Update Types, 21-2
 - Emergency Fix, 21-10
 - Regular Software Update, 21-2
- Space Reclamation, 21-12
 - Reclaim Patch Space, 21-12
 - Reclaim Space in SU Storage Directory, 21-12
- Spare Circuit Packs,
 - General Guidelines, 17-1
- Specify Conditional/Unconditional Mode, 11-21
- Specify DFC Preference, 11-19
- Specify **ENTER** Command, 11-17
- Specify First Disk Sequence Option, 11-25
- Specify First Disk to Be Updated, 11-24
- Specify Second Disk Sequence Option, 11-27
- Specify Second Disk to Be Updated, 11-26
- Specify Tape Device, 11-22
- Specify Third Disk Sequence Option, 11-29
- Specify Third Disk to Be Updated, 11-28
- Specify UCL Argument with Command Line Argument, 11-50
- Specify Update Backup Partitions Option, 11-20

- Specify Update Method, 11-18
- splinfo* Record in the ECD, 12-123
- Status Indications,
 - KS-24367, L1/Comcode 407771260 DAT Drive Status Indicators, 19-40
 - NCR006-3300608/Comcode 407545243 DAT Drive Status Indicators, 19-40
 - NCR006-3503341 SCSI DAT Drive Status Indicators, 19-40
- Status of Requests, Obtain, 6-20
- Stepping Through the **EXC** Command, 7-10
- Stepping Through the **STOP** Command, 7-13
- Stepping Through the **SW** Command, 7-15
- Stop Operation, 9-42
- Stop REX Command, 5-4
- STOP** Utility, 11-7
- Stopping Audits, 14-21
- Storage Module Drive Disk File Controller Unit, 1-11
- Storing Data in the SCSD Option Block, 16-15
- SU Format, 21-4
- SU Installation, 21-16
- SU Maintenance, 21-20
- SU Reception from the CSCANS, 21-8
- SU Reception from the SCANS, 21-7
- Successful System Update Session
 - Example, 11-15
 - Boot to New Generic, 11-40
 - COMMIT** Process Terminates, 11-42
 - Continue Update for Disk 770, 11-33
 - Continue Update for Disk 772, 11-35
 - Continue Update for Disk 775, 11-37
 - Display Enter Arguments Page 2 Screen, 11-23
 - Display System Update Menu Page, 11-16
 - ENTER** Process Terminates, 11-38
 - Execute **COMMIT** Command, 11-41
 - Execute **ENTER** Command, 11-31
 - Execute **PROCEED** Command, 11-39
 - Mount Tape for Disk 770, 11-32
 - Mount Tape for Disk 772, 11-34
 - Mount Tape for Disk 775, 11-36
 - PROCEED** Process Terminates, 11-40
 - Return to System Update Menu Page, 11-30
 - Specify Conditional/Unconditional Mode, 11-21
 - Specify DFC Preference, 11-19
 - Specify **ENTER** Command, 11-17
 - Specify First Disk Sequence Option, 11-25
 - Specify First Disk to Be Updated, 11-24
 - Specify Second Disk Sequence Option, 11-27

- Successful System Update Session
 - Example (Continued)
 - Specify Second Disk to Be Updated, 11-26
 - Specify Tape Device, 11-22
 - Specify Third Disk Sequence Option, 11-29
 - Specify Third Disk to Be Updated, 11-28
 - Specify Update Backup Partitions Option, 11-20
 - Specify Update Method, 11-18
 - Terminate System Update Menu Page, 11-43
- Suggested Routine Test Procedures, 6-11
- Sun* System Product Installation, 22-5
- Supplementary Phases, A-11, B-13
- Supplementary Phases for CU, A-11, B-13
- Supplementary Phases for DFC 2, B-13
- Supplementary Phases for DFC 3, B-13
- Supplementary Phases for DFCs 2 and 3 SCSI Magnetic Tape, B-13
- Supplementary Phases for IOP, A-11
- Supplementary Phases for IOP 2, B-13
- Supplementary Phases for SCSI DFC, A-11
- Supplementary Phases for SMD DFC, A-11
- Supplementary Phases, Requesting, 6-10
- SUPR Description, 11-5
 - Conditional and Unconditional Execution Mode, 11-8
 - SUPR Processes, 11-5
 - SUPR Utilities, 11-6
- SUPR Processes, 11-5
 - BACKOUT** Request, 11-6
 - COMMIT** Request, 11-6
 - ENTER** Request, 11-5
 - PROCEED** Request, 11-6
 - RESTORE** Request, 11-6
- SUPR Utilities, 11-6
 - CONTINUE** Utility, 11-7
 - READLOG** Utility, 11-7
 - STOP** Utility, 11-7
- Switching Control Center Interface, 2-18
- Synchronous Data Link Restore Tool, 6-54
- System Audits,
 - Operating-System Audits, 14-50
 - Procedures for Modifying Audit Records, 14-38
 - System Integrity Subsystem, 14-3
- System Backup, 9-11
 - 3B20D/3B21D Computers TOP Tape, 9-11
 - Common TOP Tape, 9-12
- System Bootstrap, 10-9
- System Configuration, 7-1

- System Generation Database, 13-74
 - File-System Body Form, 13-75
 - Process Form, 13-78
 - System Generation Form 2, 13-77
 - ULARP Form, 13-79
- System Generation Form 2, 13-77
- System Integrity Subsystem, 14-3
 - Audit Control Subsystem, 14-5
 - Audit Output Messages and Error Reports, 14-22
 - Audit User Input Commands, 14-16
- System Restoration, 3-18
- System Update,
 - Input Messages, 11-59
 - SUPR Description, 11-5
 - System Update Menu Page
 - Description, 11-9
 - System Update Procedures, 11-1
 - System Update Session Examples, 11-15
 - Troubleshooting System Update Procedures, 11-54
- System Update **BACKOUT** Example, 11-44
 - Boot to Old Generic, 11-45
 - Execute **BACKOUT** Command, 11-44
 - Execute the **RESTORE** Command, 11-45
- System Update Menu Page Description, 11-9
 - Enter Arguments Page 1 Screen, 11-12
 - Enter Arguments Page 2 Screen, 11-13
 - System Update Menu Page Screen, 11-10
- System Update Menu Page Screen, 11-10
- System Update Methods, 11-3
 - Backup Partition Method, 11-3
 - Off-Line Disk Method, 11-4
- System Update Procedures, 11-1
- System Update Session Examples, 11-15
 - Application Dependent Processing Example, 11-52
 - RETRO Argument, 11-53
 - Successful System Update Session Example, 11-15
 - System Update **BACKOUT** Example, 11-44
 - System Update Utilities Examples, 11-47
 - Unconditional Execution Example, 11-49
- System Update Steps, 11-2
- System Update Utilities Examples, 11-47
 - Execute **READLOG** Command, 11-48
 - Execute **STOP** Command, 11-47

T

- Tape Capacity/Tape Density, 19-28
 - 9-Track Drive Tape Capacity, 19-28
 - 9-Track Drive Tape Density, 19-28
 - DAT Drive Tape Capacity, 19-29
 - DAT Drive Tape Density, 19-29
- Tape Device *logdev* Records, 20-1
- Tape Drive, 1-29, 1-46, 1-49
- Tape Unit Frame, 1-14, 1-29
 - 300 V A Inverter Synthesizer, 1-29
 - Format Control Unit, 1-14
 - Magnetic Tape Unit, 1-14, 1-29
- Tape, Load SUs from, 21-40
- Tape/Disk Cabinet, 1-29, 1-46
 - 340-MB Disk, 1-29, 1-47
 - Disk Power Supply, 1-30, 1-47
 - Disk Power Switch, 1-30, 1-47
 - Tape Drive, 1-29, 1-46
- Teletype Model 4424 Terminal, 3-40
- Terminate System Update Menu Page, 11-43
- Terminate the Field Update Session, 13-72
- Terminating the Running DCI Driver, 22-10
- Test Procedures, A-6, B-6
 - CU Suggested Routine, B-6
 - CU Suggested Routine Test Procedure, A-6
 - DCI Suggested Routine, B-12
 - DFC 2 and DFC 3 Suggested Routine, B-8
 - DFC Suggested Routine Test Procedure, A-8
 - IOP 2 Suggested Routine, B-10
 - IOP Suggested Routine Test Procedure, A-9
- Timed Audits, 14-14
- TIMER, 4-13
- TLP Circuit Pack List, Clear Trouble Using the, 6-36
- TLP Option, Interpret Output from, 6-30
- Training, 2-18
- Troubleshooting Hints, 13-31
- Troubleshooting System Update Procedures, 11-54
- Troubleshooting the SCSD, 16-79
- Typical Update Package States, 13-19

U

- ucb* Form, 22-13
- ULARP Form, 13-79
- ULARP Records in the ECD, 10-7
- UN33/UN933 Circuit Pack (SCSDC Board), 16-19
- Unconditional Execution Example, 11-49
 - Add UCL Argument to Specified Command Line Argument, 11-51
 - Remove UCL Argument from Specified Command Line Argument, 11-51
 - Specify UCL Argument with Command Line Argument, 11-50
- Unconditional Execution Mode, 11-8
- Unloading, 19-36
- Update Display Command, 13-31
- Update Files, 13-15
- Update Methods, 13-1
 - File Replacement, 13-2
 - Function and Byte Replacement, 13-2
- Update Methods and Processes, 13-1
 - Field Updatable Processes, 13-3
 - Update Methods, 13-1
- Update Package Rules, 13-15
- Update Package States, 13-18
 - Other Update Package States, 13-20
 - Typical Update Package States, 13-19
- Use of the **OP:ULARP** Command, 10-15
- Using a Warm Spare Disk Drive, 9-31
- Using Automatically Generated Data for Troubleshooting, 2-17
- Using CRON, 10-18
- Using File-System Audits, 9-5
 - File-System Block and Link Audits, 9-5
 - File-System Compaction Audit, 9-8
- Using Log Files, 8-3
- Using OFLBOOT, 7-3
- Using Operating System Audits, 10-4
- Using Processor Recovery Messages, 3-33
- Using the **DGN** Command, 6-8
- Using the **EXC** Command, 7-5
- Using the Magnetic Tape System, 15-3
- Using the **STOP** Command, 7-12
- Using the **SW** Command, 7-15
- Utility Circuit, 1-55
- UUCP Feature, 23-1
 - UUCP Manual Pages, 23-65
- UUCP Manual Pages, 23-65

V

- Verification, 21-9
- Verification of Proper Installation, 13-32
- Verify DIOP Mode in Effect Procedure, 3-60
- Verify the Integrity of Pumpcode, 19-4
- Verifying Scan Point Status, 16-76
- Verifying SCSD Operation, 16-76
 - Troubleshooting the SCSD, 16-79
 - Verifying Scan Point Status, 16-76
 - Verifying SD Point Operation, 16-78
- Verifying SD Point Operation, 16-78

W

- When to Use the Compaction Audit, 14-156
- Write Protection, 19-27
 - 9-Track Drives, 19-27
 - DAT Drives, 19-27
- Writing a File from a Disk, 15-7
- Writing SMD Disk Drive Defect Tables, 10-19

