

OPERATIONS GUIDE

PLATO LESSON DELIVERY AND AUTHORING 1

RELEASE 34.2 - NOS 2.3 LEVEL 617

RELEASE 34.2 - NOS 2.4.1 LEVEL 630

SMD130934

© Control Data Corporation, 1985  
All Rights Reserved

## Table of Contents

1	PLATO Overview	1
1.1	PLATO programs	1
1.1.1	MASTOR	2
1.1.2	MASTORN	2
1.1.3	PLATO	3
1.1.4	FRAMAT / FFORMAT	4
1.1.5	PNI	4
1.1.6	CONDENSOR	5
1.1.7	Control Point Communication	5
1.1.8	Utility Programs	6
1.1.8.1	Print Utilities List	6
1.1.8.2	Operations Utilities List	6
1.1.8.3	Batch Job Utilities List	7
1.1.8.4	Application Utilities List	7
1.1.8.5	Backups Utilities List	7
1.1.8.6	Usage Data Utilities List	8
1.1.9	Utility Lessons	8
1.2	NDS User Names	9
1.3	PLATO Master Files	10
1.3.1	NDS - Master File Relationship	10
1.3.2	Master File Types	10
1.3.2.1		11
1.3.2.2	Required vs. Temporary	22
1.3.3	Master File Names	13
1.3.4	File Allocation Within Master Files	13
1.4	Special User Groups	14
1.4.1	PLATO Support (group "s")	14
1.4.2	Courseware Services (group "coserv")	14
1.4.3	System Controllers (group "p")	15
1.4.4	Operators (group "o")	15
1.4.5	Communications (group "m")	15
1.4.6	System Consultants (group "pso")	15
1.5	User Features	16
1.6	System disk space requirements	16
2	System Operation	18
2.1	Start-up	18
2.1.1	Deadstart	18
2.1.2	Turn Off Broadcast Unit	19
2.1.3	Bring Up The PLATO Application	18
2.1.4	System Check	18
2.2	Shutdown	18
2.2.1	Start File Dumps	18
2.2.2	Warning Message	19
2.2.3	Backout Users	19
2.2.4	Stop Control Points	19
2.2.5	Turn On Broadcast Unit	20
2.2.6	Do File Dumps If Required	20
2.2.7	Checkpoint Operating System	20
2.2.8	Stop System	20
2.2.9	Finish	20
2.3	System Crashes	21
2.3.1	System Monitoring	21
2.3.2	Check Terminal	21

2.3.3	Check PLATJ Control Points	21
2.3.4	wait for One Minute	22
2.3.5	Reload Missing Control Points	22
2.3.6	60 Second Backout	23
2.3.7	Turn On the Broadcast Unit	23
2.3.8	System is Hung or Major Problems	23
2.3.9	Shut Down	23
2.3.10	Deadstart Dump	24
2.3.11	Try a Level 3 then Level 0 Deadstart	24
2.3.12	Problem is Solid	24
2.3.13	recording the Interruption	24
2.4	Monitoring Error Messages	24
2.5	Broadcast unit	25
2.6	Prints	25
2.7	Preparing a Pack for Use	25
2.7.1	Initialize the Pack	26
2.7.2	MST	26
2.7.3	FORMAT if Necessary	27
2.7.4	Reinitialize the Pack	27
2.7.5	Summary of Disk Preparation	27
2.8	Disk Error Recovery Procedures	27
2.8.1	Single File Error Recovery	28
2.8.2	Master File Recovery	29
2.8.2.1	Binary Master Files	29
2.8.2.2	Other Master Files	29
2.8.3	Disk Pack Recovery	29
2.8.3.1	System Pack Recovery	29
2.8.3.2	Pack Recovery	30
2.8.4	Recovery Integrity Tests	30
2.9	Copying PLATO Files to Tape	30
3	Network Management	32
3.1	NAH	32
3.1.1	Rebuilding the 2550 Load File	32
3.2	PLATJ-NAH Interface (PNI)	32
3.2.1	Rebuilding the Resident Load Files	32
3.2.2	PNI with AIP Trace	33
3.3	The Network Database	33
3.3.1	Verifying the Database	33
3.3.2	Scanning the Database	34
4	PLATJ Account Management	35
4.1	Account Creation	35
4.2	Editing an Account	35
4.3	Destroying an Account	36
4.4	Renaming an Account	36
4.6	The File Management Log	36
4.7	Account Restoration	37
4.7.2	Partial Restoration	37
4.7.3	Complete Restoration	37
4.8	emergency Actions	38
4.9	TRANSMIT Utility Usage	38
4.10	Lesson "operator"	39
5	"Runner" Lesson Management	40
5.1	Runner Lessons	40

6	File Archive Management	42
6.1	General Description	42
6.2	Disk Pack Healing Scheme	43
6.3	Setting Archive Rights	43
6.4	Archive Processing	44
6.5	Retrieval Processing	44
7	File Dumps/Backups	45
7.1	Dump Sets	45
7.2	Dump Phase Overview	45
7.3	Initializing the Slot Table	46
7.4	Setting Up Dump Directory Datasets	46
7.5	Changing Master Files to be Dumped	47
7.6	Writing a Message in the VSN Table	47
7.7	File Dumping	48
7.8	Recovering a Single File	49
7.9	Recovering a Master File	50
7.10	Backups Utilities	50
7.10.1	BACKCOPY	51
7.10.2	BACKDMP	51
7.10.3	BACKLIB	51
7.10.4	BACKLST	52
7.10.5	BACKMDD	52
7.10.6	BACKONE	53
7.10.7	BACKTWO	54
7.10.8	BACKSTART	55
7.10.9	COPYMF	55
7.10.10	COPYPF	56
7.10.11	MFDX	57
7.10.12	RECOVAL	58
7.10.13	RECOVMF	58
8	File Conversions	59
8.1	General Procedures	59
8.1.1	Load PLATO with "PLAINS."	60
8.1.2	Clear Leslist	60
8.1.3	Choosing the Conversion	61
8.1.4	Setting Parameters	61
8.1.5	Conversion Methods	61
8.1.6	Fix Problems	62
8.2	Execution Errors	62
8.3	Error Messages	63
8.4	Conversion Programs	65
8.4.1	Conversion Programs Continued	65
8.5	Conversions by release	67
8.5.13	Release 13 -- March, 1976	67
8.5.15	Release 15 -- August, 1978	67
8.5.18	Release 18 -- May, 1979	67
8.5.19	Release 19 -- November, 1979	67
8.5.20	Release 20 -- March, 1980	67
8.5.21	Release 21 -- June, 1980	67
8.5.22	Release 22 -- October, 1980	67
8.5.23	Release 23 -- February, 1981	67
8.5.25	Release 25 -- February, 1982	67
8.5.27	Release 27 -- June, 1982	68
8.5.31	Release 31 -- October, 1983	68

9	PLATO Inter-System Link	69
9.1	Inter-System Link Operation	69
10	Usage Tracking	71
10.1	How the Tracking is Done	71
10.1.1	NDS Account File	71
10.1.1.1	PS entry	71
10.1.1.2	PA entry	71
10.1.1.3	PR entry	72
10.1.1.4	PC entry	72
10.1.1.5	PI entry	72
10.1.1.6	PX entry	73
10.1.1.7	PL entry	73
10.1.1.8	PM entry	74
10.1.1.9	PD entry	74
10.1.1.10	PJ entry	75
10.1.1.11	PO entry	75
10.1.2	DFTERM	75
10.1.3	The Billing Cycle	76
10.1.4	Daily Processing	76
10.1.5	End of the Billing Cycle	76
10.2	Account Summary Processing	77
10.2.1	Account Summary Data Format	77
10.2.2	NDS Account Data Summary Format	79
10.3	Reports and Programs Available	80
10.3.1	Availability Report - RAFFDD	80
10.3.1.1	Sample Program	81
10.3.1.2	Error Exits	81
10.3.2	Port Usage Report - PJRTX	82
10.3.2.1	Sample Program	83
10.3.3	Generating a sorted billing cycle file	83
10.3.3.1	Reducing the file - RAFFBC	83
10.3.3.2	Sorting the file - ASM1	84
10.3.3.3	Sample Program	84
10.3.3.4	Error Exits	84
10.3.4	User Usage Report - JURBC	84
10.3.4.1	Sample Program	86
10.3.5	Lesson Usage Report - LURBC	85
10.3.5.1	Sample Program	87
10.3.6	Combined Reports & Example	87
10.3.7	DATESCN	88
11	Courseware Installation	89
11.1	Courseware Update Installation	89
11.2	Destroy Obsolete Courseware	90
12	The PLATO bulletin board	92
13	Consultant Features	93
13.1	"e"	93
13.2	"pnotes"	93
13.3	"aids"	93
14	Logical Site Director Features	94
14.1	Recommended Use	94
14.2	Setting Up Additional Sites	94
14.3	Site Director Options	95

20	Application Utilities	97
20.1	CMUMP	97
20.2	CONDX	97
20.3	CONFIGX	98
20.4	EFAD4P	98
20.5	EMDMQ	98
20.6	EMDTAPE	99
20.7	EXEC	99
20.8	FORMCMD	99
20.9	FRANX	100
20.10	MASJOB	101
20.11	MFNX	101
20.12	PLATX	102
20.13	PNIX	103
20.14	PRDUTE	104
20.15	SUBMITM	104
20.16	VERSX	104
21	Operations Utilities	106
21.1	CJNSOLE	106
21.2	CJPPYD	107
21.3	DDPT	107
21.4	DJMPPRT	108
21.5	ECSTST	109
21.6	EMPRT	109
21.7	ESH	110
21.8	MENPRT	110
21.9	MFPACK	110
21.10	MFTCOPY	111
21.11	MFTLOAD	111
21.12	MFUTIL	112
21.12.1	MFADD	112
21.12.2	MFALTER	112
21.12.3	MFCREAT	113
21.12.4	MFLIST	114
21.12.5	MFPRINT	114
21.12.6	PFDEST	114
21.12.7	PFIN	115
21.12.8	PFOUT	115
21.12.9	PFUSE	116
21.13	NETPRT	117
21.14	PDCAT	117
21.15	PDPRT	117
21.16	REQPACK	117
21.17	WAIT	118
22	Print Utilities	119
22.1	ACCPRT	119
22.2	JJCPRT	119
22.3	JPRINT	120
22.4	MDCPRT	120
22.5	NPRINT	120
22.6	PLMPRT	120
22.7	PPRINT	120
22.8	TPRINT	120

30	Problem Reporting	121
30.1	Purpose	121
30.2	Definitions	121
30.3	User Responsibilities	122
30.3.1	Monitoring Your System	122
30.3.2	Reporting Problems	123
30.3.2.1	Problem Reporting Procedures	123
30.3.2.2	Information Required	123
30.3.2.3	Setting a Priority Level	123
30.4	ODC Responsibilities	127
30.4.1	User Interface Organization	127
30.4.2	PSR Status Reporting	127
30.4.2.1	Resolution Schedule	127
30.4.2.2	PSR Responses	127
40	Add-on Products	130
40.1	PCD2	131
40.1.1	PCD2 Operation	132
60	Error Messages	133
60.1	Messages A	134
60.2	Messages B	135
60.2.1	Messages BA) S	135
60.2.2	Messages BAT	137
60.3	Messages C	139
60.3.1	Messages CH	140
60.4	Messages D	142
60.4.1	Messages DDP	142
60.4.2	Messages DE	144
60.5	Messages E	145
60.5.1	Messages EM	145
60.5.2	Messages EN	147
60.5.3	Messages ERRJR L	149
60.5.4	Messages ERKOR (number)	150
60.5.5	Messages EX	151
60.6	Messages F	152
60.7	Messages G	153
60.9	Messages I	154
60.9.1	Messages IM	155
60.10	Messages J	157
60.11	Messages K	157
60.12	Messages L	158
60.13	Messages M	160
60.13.1	Messages MAX	161
60.13.2	Messages MR	163
60.14	Messages N	164
60.14.1	Messages NJ L	165
60.14.2	Messages NDA	167
60.15	Messages O	168
60.16	Messages P	169
60.16.1	Messages PL	170
60.16.2	PNA	172
60.16.3	Messages PNI E	172
60.16.4	Messages PNI F	174
60.16.5	Messages PD	175
60.17	Messages Q	176
60.18	Messages R	176



60.18.1	Messages RED	173
60.19	Messages S	179
60.19.1	Messages ST	180
60.20	Messages T	182
60.21	Messages U	183
60.23	Messages W	184
60.23.1	Messages W3	186
60.26	Messages Z	185
70	Release Changes	187
70.1	Release Changes	187
80	Alphabetical Index	188
80.1	Index: A	189
80.2	Index: B	190
80.3	Index: C	191
80.3.1	Index: CO	192
80.4	Index: D	193
80.5	Index: E	194
80.6	Index: F	195
80.7	Index: G	195
80.8	Index: H	196
80.9	Index: I	196
80.10	Index: J	197
80.11	Index: K	197
80.12	Index: L	197
80.13	Index: M	197
80.13.1	Index: MASTOR	199
80.13.2	Index: MF	199
80.14	Index: N	200
80.15	Index: O	201
80.16	Index: P	201
80.16.1	Index: PL	202
80.16.1.1		202
80.16.2	Index: PA	203
80.16.3	Index: PS	203
80.17	Index: Q	205
80.18	Index: R	205
80.19	Index: S	205
80.19.1	Index: SY	207
80.20	Index: T	207
80.21	Index: U	209
80.22	Index: V	209
80.23	Index: W	209
80.26	Index: Z	209
80.27	Index: Numbers	209

## 1 PLATO Overview

Documentation for the PLATO application will be found in the following publications.

PLATO Installation Guide	SMD 130933
PLATO Operations Guide	SMD 130934
PLATO Configuration Handbook	SMD 130935
PLATO User's Guide	LDS 97405900

### 1.1 PLATO programs

The PLATO application consists of several jobs each running at its own control point. The job names are shown below. Refer to the following sections for information on each program listed below.

MASx	MASTOR / MASTORN control point
PLAx	PLATO executor control point(s)
FORx	FRAMAT / FORMAT control point(s)
PNIx	PLATO / NAM Interface (PNI) control point
CDIx	CONDENSOR control point(s)

where,

x = program ordinal. If x = 1, then this is the primary copy of a program. For example, MAS1, PLA1, etc.  
If x > 1, then this is a secondary copy of a program.

l = alphabetic representation of the "aford" configuration file entry. This is used to identify which mainframe a CONDENSOR is running on. In single mainframe systems, this will always be the character "A".

On a typical, single mainframe system the control points which are always active are:

MAS1	MASTOR
PLA1	PLATO
FOR1	FRAMAT
PN11	PNI
CD1A1	CONDENSOR

Multi-mainframe features are not available for CYBER 170-800 series mainframes. Multiple copies of PLATO executors and formatters are also not available on CYBER 170-800 series mainframes, but may be used on CYBER 170-700 series mainframes, in either single or multiple mainframe configurations. Multiple copies of PLATO condensers may be run on any mainframe. Only a single copy of PNI may be run on a single mainframe system, while multiple mainframe systems may have a copy of PNI on each mainframe.

MAS1 always identifies the MASTOR job while MAS2, MAS3,

etc. always identify a MASTORN job. Only one job with the MASx name may appear on each mainframe of a multi-mainframe system. The MASTOR (MAS1) job runs only on the primary mainframe (the one with "aford=0." in its configuration file), while the MASTORN job runs on each of the other mainframes.

FOR1 always identifies the FRAMAT (master formatter) job while FOR2 always identifies the FORMAT job.

### 1.1.1 MASTOR

MASTOR is the interface between the PLATO application and the Network Operating System (NOS). MASTOR obtains Extended Memory (EM) from the operating system and allocates it to parts of the PLATO application. It also manages PLATO disk files, submits the procedures which bring up the rest of the application, handles requests for console-like display information and passes job action requests to the operating system. Each PLATO system may have only one MASTOR program.

MASTOR requires at least two dedicated PPUs. MRQ is used as an interface to the operating system to perform functions which are more easily done in a PPU rather than a CPU program. PMS is the PLATO application disk file processor. All disk accesses made by the PLATO application on PLATO master files are handled by PMS. Multiple copies of PMS are allowed by setting parameters in the PLATO configuration file. MASTOR also uses two transient PPU programs. MAX is called only during the initialization of MASTOR to reserve EM for the PLATO application. EPE is called by MASTOR and all other PLATO application programs when an EM error occurs.

The MASTOR job is normally started through the "PLATO." DSD command and stopped through entering "K.STOP" on the K-display when assigned to the MASTOR job.

If MASTOR fails, the entire PLATO application must be reloaded by individually dropping each of the control points and using normal procedures to restart the MASTOR job.

### 1.1.2 MASTORN

MASTORN performs a subset of the functions performed by MASTOR. It is only used on a secondary mainframe in a multi-mainframe system.

In a multi-mainframe system, each of the secondary mainframes will have a MASTORN program. MASTOR handles requests from these copies of MASTORN for PLATO file management and will send requests to the copies of MASTORN to load and/or drop PLATO application control points on secondary mainframes, to return information for the console-like displays, and to perform other job action requests, such as submitting and dropping background batch jobs.

Each MASTORN requires one dedicated PPU for a copy of MRQ.

MASTORN is normally started through the "PLATO." DSD command. The job will automatically determine that the MASTORN program should be loaded instead of the MASTOR program when this command is entered on a secondary mainframe. If MASTORN is the only PLATO application control point on a mainframe, then it may be stopped via a "KILL,jsn." console entry, where "jsn" is the job name. If other PLATO application control points are active on a mainframe, they must be stopped using options in lesson "sysopts" before MASTORN may be stopped.

The effects of a MASTORN failure depend on what other control points are also running on that mainframe. If MASTORN is the only PLATO application control point on the mainframe, there will be no effect on the rest of the PLATO application other than no background batch jobs may be submitted to that mainframe and no console or job information will be available for that mainframe. If other PLATO application control points are active on the mainframe with the failed MASTORN, the effect on the entire application is dependent on which jobs are running at those control points. In general, the effect is a degradation which may be recovered from by reloading the MASTORN job and any other failed jobs on that mainframe.

### 1.1.3 PLATO

PLATO is the PLATO Author Language executor program. When executing a lesson, parts of the lesson are swapped from Extended Memory (EM) into the PLATO Central Memory (CM) field length. The PLATO application then interprets the commands of the lesson. Output generated by users executing lessons is sent to FRAMAT for further processing (formatting) and then on to the various network processors such as PNI. The CONDENSOR converts PLATO Author Language into a more easily interpreted form under the control of the PLATO executor. PLATO makes various requests to MASTOR for all PLATO file management and operating system functions.

NOTE: PLATO system applications written in the PLATO Author Language (TUTOR) are often also called "lessons" in this Guide.

The primary PLATO executor (PLA1) requires one or two dedicated PPU's if the Computer Interface Unit (CIU) Network is available. This network is not available for CYBER 170-800 series machines. PIJ is the driver program for the CIU. If two CIUs are used, there is one copy of PIJ for each CIU. Secondary executors (PLA2, PLA3, etc.) do not require dedicated PPU's.

The PLATO executor is automatically started and stopped by MASTOR with no operator intervention. Secondary executors may be started and stopped through options in lesson "sysopts".

The failure of any executor for any reason will cause the entire PLATO application to abort. MASTOR will automatically restart the PLATO and all other control points if possible.

#### 1.1.4 FRAMAT / FORMAT

FRAMAT and FORMAT receive terminal output in a standardized format from the PLATO executor, format it into output for specific terminal types, and place the result into "parcels". Parcels are then received by a secondary process called the FRAMER running only in the FRAMAT control point. If the terminal is connected to the CIU network, the parcels are reformatted into "frames" to be sent to the CIU through the PPU program PII. Otherwise, the output is sent to the network processor, PNI.

FRAMAT is automatically started and stopped by MASTOR. If more than one formatter is started, the first one to complete initialization is considered to be the primary formatter, FRAMAT, which cannot be stopped without interrupting service. The secondary formatter may be started and stopped through options in lesson "sysopts".

Failure of FORMAT will cause degradation of the PLATO application by overloading the formatting process in FRAMAT if the system is heavily loaded. Failure of FRAMAT will stop all terminal output because of the loss of the FRAMER process. FRAMAT and FORMAT are not automatically reloaded by MASTOR as the PLATO control point is, but may be restarted through operator intervention by executing the procedure FRAMX at the computer console.

#### 1.1.5 PNI

The PLATO-NAM Interface (PNI) is used by the PLATO system as a communications medium to the NDS communications network, referred to as Network Access Method (NAM), and the American Standard Code for Information Interchange (ASCII) terminals connected to it.

The PLATO/NAM Interface program (PNI) directs the traffic between NAM and PLATO. NAM provides a simple interface to the network, but does not provide the type of interface required by PLATO. PNI will perform the functions required to interface NAM to PLATO.

PLATO ASCII terminals such as the IST2, IST3 and CDC-721 have no ability to perform any function until a resident is loaded. This resident must be loaded before the terminal can log into NAM. To perform this function, a downline-load is initiated by PNI when a terminal requires its resident to be loaded.

PNI reads the terminal resident binary data stored in a NDS permanent file and transmits the data via NAM to a loadable terminal in transparent mode.

Load protocol error processing is done via the downline load module. Upon detection of an error, the terminal sends a "no acknowledge" message (NAK) followed by the block number

in which the error occurred to the downline load module. The downline load module will then back up to that block number and resend the data.

Once an ASCII terminal is logged into the PLATO application, PNI monitors the flow of data between the terminal and a lesson. The keys that the user types at the terminal are forwarded to PNI by NAM. PNI will translate these keys from the terminal ASCII character set to the PLATO terminal key codes and place the keys in the application key buffer. If the key buffer is full, PNI will hold these overflow keys in a buffer. As the key buffer empties, PNI will place the overflow keys into the application key buffer.

Output for the terminal will be received from FRAMAT and given to NAM. PNI will control the flow of output by insuring that the number of outstanding blocks is always less than the maximum number allowed for this connection. As NAM delivers blocks to the terminal, PNI will request more blocks from FRAMAT.

PNI will also process all NAM supervisory messages. If the application needs to be informed of a particular condition, PNI will pass the appropriate status information along.

PNI does not require dedicated PPU's. PNI uses one transient PPU program, PNA, for operating system functions.

PNI is automatically started and stopped by MASTOR.

If PNI fails, all ASCII terminals will be unable to continue communication with the PLATO application. PNI may be restarted through operator intervention by executing the procedure PNIX at the computer console. All users of ASCII terminals are signed out of the PLATO application when PNI is restarted.

#### 1.1.6 CONDENSOR

The CONDENSOR is something like a compiler. When an author edits a lesson, it must then be condensed. The CONDENSOR then takes the source code and prepares a binary copy of the lesson. It is this binary copy which is executed when a student takes a lesson.

CONDENSOR does not require dedicated PPU's.

After the PLATO application has been loaded, it will start and stop condensers as necessary, or the operator may do this manually via lesson "sysupts".

CONDENSOR failure normally should not affect the rest of the PLATO application. The CONDENSOR may be restarted by executing the procedure CONDX at the computer console.

#### 1.1.7 Control Point Communication

Communication between the various PLATO control points and MASTOR is handled by a transient PPU program or through EM shared by all control points. MAS is called by the various PLATO control points during initialization to request EM and set up shared EM request areas to be used later. MAS is also called by several utility programs to make requests to MASTOR to read and write PLATO files.

During the initialization of MASTOR, all EM needed for the PLATO application is reserved and assigned to the MASTOR control point. Later, at the request of the PLATO executor, most of this EM is assigned to the executor. When the FRAMAT, PNI and CONDENSOR control points initialize, they are allowed to use the same EM which was assigned to the executor. All communication among these control points is handled through buffers in this shared EM.

In multi-mainframe configurations, all copies of MASTORK, each running on a different mainframe, communicate with MASTOR on the primary mainframe through this shared EM. The other PLATO control points also share the same EM, no matter which mainframe they are running on.

### 1.1.8 Utility Programs

There are many utility programs supplied as part of the PLATO application. These are briefly described below, with references to more complete descriptions.

#### 1.1.8.1 Print Utilities List

##### File print utilities

These utilities are needed for central system prints requested through lesson "prints". They are documented in the section "Print Utilities".

ACCPRT - print account file management logs  
(Submitted by lesson "account1".)  
DJCPRT - documentor files  
DPRINT - student data files  
HJCPRT - PLM modules  
NPRINT - general and student notes files  
PLMPRT - PLM groups  
PPRINT - used to format upper/lower case prints  
TPRINT - tutor and code files, nameset and datasets

#### 1.1.8.2 Operations Utilities List

##### Operations utilities

These utilities are documented in the "Operations Utilities" section.

CONSOLE - PLATO terminal simulator  
CUPYPD - copy PLATO application dump files to tape  
DOPT - ESM low-speed port or ODP test utility

DUMPPRT - print PLATO application dump tapes  
 ECSTST - EM test utility  
 EMPRT - print PLATO application EM dump file  
 ESM - ESM management utility  
 MEMPRT - print PLATO application memory dump files  
 MFPAK - procedure to recover fragmented disk space  
           on PLATO master files  
 MFTCOPY - procedure used to copy PLATO master files to  
           a tape for shipment to another system  
 MFTLOAD - procedure used to copy PLATO master files from  
           a tape to disk  
 MFUTIL - master file utilities  
 NETPRT - print network (lesson "pnet") database  
 PDCAT - catalog PLATO application dump tape  
 POPRT - print contents of master file directories  
 REPACK - request removable pack  
 WAIT - pause batch job

### 1.1.8.3 Batch Job Utilities List

#### Utilities used in background batch jobs

These are user utilities. See lesson "noselds" for documentation.

PCODE - set codewords for access to PLATO files  
 PF - read and write PLATO files  
 PPACK - set master file for PLATO file access  
 TFORM - format files to be written into PLATO files

### 1.1.6.4 Application Utilities List

#### Utilities used in PLATO application jobs

These utilities are documented in the "Application Utilities" section.

CMDMP - dump central memory to a file  
 CONDX - CONJENSJR load procedure  
 CONFIGX - procedure to obtain the PLATO configuration file  
 EFRDMP - print contents of ESM extended flag registers  
 EMDMP - dump extended memory to a file  
 EMDTAPE - format PLATO application dump tape  
 EXEC - secondary PLATO executor load procedure  
 FORMCMD - format PLATO application CM dump file  
 FRAMX - FRAMAT / FORMAT load procedure  
 MASJOB - controls security of PLATO jobs  
 MFNX - procedure to attach required PLATO master files  
 PLATX - primary PLATO executor load procedure  
 PNI - PNI load procedure  
 PRDUTS - schedule PLATO jobs to a control point  
 SUBMITM - submit job to a specific mainframe  
 VERSX - procedure used only on the PLATO development  
           system

### 1.1.8.5 Backups Utilities List



### Utilities used by lesson "backups"

These utilities are documented in the "File Dumps/backups" section.

BACKCOPY - copy backups information from NDS to PLATO  
 BACKDMP - procedure to attach and dump master files  
 BACKLIB - print audit trail and backups parameters  
 BACKLST - print dump directory information  
 BACKMOD - change backups parameters  
 BACKONE - phase one of backups database merge  
 BACKTWO - phase two of backups database merge  
 BKSTART - reinitialize communications buffer  
 COPYMF - dump master file to tape/disk  
 COPYPF - recover PLATO file or entire master file  
 MFDX - procedure to dump master files  
 RECOVAL - procedure to recover all master files  
 RECOVMF - procedure to recover a single master file

### 1.1.8.6 Usage Data Utilities List

#### PLATO application usage data utilities

These utilities are documented in the "Usage Tracking" section.

ASK1 - sort billing cycle file  
 DATESCN - generate list of dates from billing cycle file  
 LURBC - lesson usage report  
 PORTX - PLATO port usage report  
 RAFFBC - generate billing cycle file  
 RAFFPD - application availability report  
 UURBC - user usage report

### 1.1.9 Utility Lessons

The following utilities are PLATO system applications written in the PLATO Author Language. These utilities are used to manage various resources within the PLATO application or for common operational functions. The use of these lessons is documented in the PLATO Operations Guide and the PLATO Configuration Handbook.

accounts - user options for managing PLATO accounts  
 accountu - PLATO account verify/clean-up options  
 aids - user PLATO documentation  
 allocate - logical site system options  
 archiver - archive/retrieval processing  
 backups - PLATO file recovery processing  
 binary - parameters for managing binary master file disk space  
 c - system consultant options  
 enforcer - restricted lesson options  
 installu - system installation options  
 lpedit - system operations parameters  
 ljr - master file attach/unload options  
 nosalds - user batch job documentation

operator	- PLATO file management for system controllers
opjobs	- usage tracking processing lesson
pnet	- network database editor
prints	- central print request processing
runnersys	- runner program management
site	- logical site management
stats	- system crash log
sysaids	- system features documentation
sysmtr	- system operations monitoring
sysopts	- system operations options
s0calutil	- editor to control access to system options
s0pnlif	- rebuild terminal resident load files
transfer	- PLATO file installation options
utility	- file verify/clean-up options

## 1.2 NJS User Names

Certain NJS user names (and associated files) are required for the PLATO application to run.

1. The PLATO application user name is set by the "subun" configuration file entry. The usual value is "SYS", but it may be any legal NJS user name.

The permanent file catalog for this user name will contain the file submitted by MASTER to start the rest of the PLATO application control points. This file is usually named "PLATOD".

NOTE: This user name must be validated to submit system service class jobs via the "VM" MODVAL parameter.

2. A user name to be used for submitting print jobs is set by the "prtun" configuration file entry. The usual value is "PRINTS", but it may be any legal NJS user name, including the value of "subun".

There are no required files for this user name.

NOTE: This user name must be validated to submit system service class jobs via the "VM" MODVAL parameter.

3. If you are using the PLATO Inter-system Link optional feature, there are two user names used by the link software, PLASEND and PLARECV. These names are required, the link will not work correctly if you do not have these exact user names. The permanent file catalogs of these user names contain files of data to be transmitted to remote systems or data which has been received by the local system.
4. The "PLATDIF" user name is specifically reserved in NJS for the PLATO application and is created when the operating system is installed. The user index for this user name is always 3777730.

The permanent file catalog for this user name should contain all PLATO master files and the PNI load files.

5. The "SYSTEMX" user name is created when the operating system is installed. The user index for this user name is always 377777b.

The permanent file catalog for this user name contains the PLATO load procedures PLATO, PLAINS, and PLAUPD.

### 1.3 PLATO Master Files

The PLATO application uses mass storage devices through special NOS files called "master files." These files are created and manipulated through various utility programs which are described in the "Operations Utilities" section of this Guide.

#### 1.3.1 NOS - Master File Relationship

A master file is a NOS direct access permanent file. Usually these files are very large. A full size master file takes up almost half of a single-density 844 disk pack. Once created, a master file is never shortened or lengthened.

The master files which must be available to the active PLATO application are attached during the initialization of MASTOR through a procedure called MFNX, which is on the deadstart tape. Thus, whenever there is a change in the required master files, procedure MFNX must also be changed. This will involve either making a new deadstart tape or SYSEDITing in the new MFNX. Temporary master files may be attached to or detached from the PLATO application through options in lesson "lor".

The PLATO application handles PLATO file management by assigning certain areas within the NOS master files. Since the PLATO software has this management capability built in, as far as NOS is concerned, the amount of disk space used by the PLATO application only changes when a master file is added or removed. On the PLATO application, however, the required space is changing constantly as parts of the master files are allocated to the various users. See the "File Allocation Within Master Files" section for more information.

#### 1.3.2 Master File Types

The PLATO application classifies master files into different "types" and uses them differently depending on this "type". The types of master files are:

- a. GENERAL - This is a master file which is part of the active system. All user files will automatically be placed on this type of master file. There is no way for the normal user to create a file on any other type of master file.

WARNING

A general master file should never be placed on the system, for even a second, unless it is intended to remain on the system permanently. A user might have enough time to create a file on that master file and it would be lost when the master file was unloaded.

- b. MASTER - This master file may contain the same kinds of PLATO files as a general master file, such as the source for PLATO lessons, but it is treated as a private pack. While any user can execute a lesson on a master master file, only systems people can place a file on one. Thus, this type of master file can be unloaded without affecting any of the user owned files (user owned files are not normally on this type of master file).

#### 1.3.2.1

- c. BINARY - A binary master file is used by the system to store disk copies of PLATO lesson binaries. Whenever a lesson is in use, the binary is loaded into EM. If no binary master file exists, a lesson would have to be recondensed every time a user tried to execute the lesson (unless it was already in EM). It is possible to run PLATO without a binary master file on the system, but this should not be done in a normal production mode, as system performance will be degraded due to re-condensing all lessons requested by users.

Binary files are created whenever a lesson is condensed. When a binary file is created, the directory of the master file containing the file is not automatically saved on disk. This is done periodically by a "runner program" named "checkpt". If "checkpt" is not running, no disk binaries will be saved across system reloads. See the section on "Runner Lesson Management for more information on these programs.

The system automatically destroys disk binaries of lessons in the following ways.

1. When a lesson is changed in any way, the disk binary is assumed to be obsolete and is destroyed.
2. When a user is loading a disk binary and the system discovers that the -user- file for the lesson has been changed since the binary was created, the disk binary file is destroyed and the lesson is re-condensed.
3. A runner program called "binary" automatically destroys old binary files based on parameters which are controlled by the system operators. This is done to make sure there is always a minimum available disk

space on the binary master files for new disk binaries to be saved.

4. If the PLATO configuration file is changed, all disk binaries are considered obsolete and are destroyed and recondensed when a user attempts to load the disk binary into E4.

When a serious mainframe hardware problem has occurred and been fixed, all binary master files should be destroyed and recreated to prevent further system failures due to attempting to execute incorrect binary files. If a parity error in a disk channel is found, it is very important that this be done.

- d. BACKUP - Backup master files are used to store backups of active PLATO files. If a backup master file is on the system, no file on it can be executed. However, the file may be transferred to a general or master type master file by systems personnel. Once this is done, it would be possible to execute the file in a normal manner.
- e. ARCHIVE - An archive master file is used exclusively as part of the PLATO archive feature. Archive files are transferred from the production master files to archive master files whenever the system operators process files waiting to be archived. PLATO files on these master files are treated the same as if they were on a backup master file.

#### 1.3.2.2 Required vs. Temporary

"Required master files" are those which are a permanent part of the production system. They should be attached when the PLATO application is brought up. This is done through the procedure called MFNX on the oadstart tape. If not all required master files are available, users will be unable to create files. Required master files are of types general and master.

Temporary master files may be created for the following reasons.

1. special work needed during an installation
2. creating a temporary master file in order to copy some of the active files to it
3. used as an in-between step when loading files from a tape onto the active PLATO application
4. loading a backup master file to provide a backup copy of an existing file

Temporary master files may be loaded through lesson "ldr".

They should be unloaded as soon as you are finished with them. A temporary master file should never be of type general, or regular users would be allowed to create files on it. No file should be created through "accounts", or "operator" on a temporary master file because this would create an entry in the account table. The normal way to move lessons onto or off of a temporary master file is through the use of lesson "transfer".

### 1.3.3 Master File Names

A master file name may contain any 7 alphanumeric characters which are legal NDS permanent file names.

#### NOTE

It is important that all active master files of types master and general have names which are unique within the first four characters. PLATO uses the first four characters of the master file name in forming the name of a binary file. If this has not been followed, the result will be "binary direct bad" messages appearing in the dayfile. Master files of any other type may be given any name.

The following are suggestions for master file names which have been commonly used.

- a. These names have been used to name the binary master file: "binary", "binary1", "binary2", etc...
- b. The name of the system master file is "system". Basically this master file will only contain those lessons shipped with each new release. This master file is always type master to prevent user files from being created on it.
- c. The names given to the type general master files used during normal PLATO operation to hold the user owned files are "amast", "bmast", "cmast", etc...

### 1.3.4 File Allocation within Master Files

Master files are very large NDS direct access permanent files. PLATO application files are created by allocating space within these NDS files. All PLATO files are made up of "parts" or "spaces." Each part contains 7 blocks of 320 60-bit computer words, or a total of 2240 words. A PLATO master file may contain from 1 to 5159 of these parts. A required master file should normally be the maximum size as a certain amount of space is reserved for the master file directory in EM regardless of actual size. The maximum size of a master file depends on the type of mass storage device it is created on. See the section on MFCREAT for information

on maximum sizes of master files and how many master files will fit on a specific device type. Every master file starts with a file named "4;;;;;;;;;" which is used to store certain information about the master file, and to hold the disk copy of the directory used by PLATO to locate specific PLATO files within the master file.

When a user creates (copies, extends, etc.) a file, the system tries to create the file on a type general master file. The master file is chosen based on the free space remaining on that master file. The first attempt is made to fit the file on the type general master file with the second most amount of space remaining. If enough space is not found, all other general master files will be checked in order of increasing space remaining. This algorithm is used in an attempt to save the one with the most free space until it is really needed.

The space used for a new PLATO file must be contiguous space. Thus, if a user gets the message "no room for this file", it means that there is not enough CONTIGUOUS space on any master file. (Unlike regular users, system people are able to specify which master file the file will go onto. This is useful when a file needs to go onto a type master master file).

When a user tries to access a file, a search is made through the directories of each master file. The order of the search is based on the order in which the master files are attached in the procedure MFNX. Only type master and general master files are searched. It is possible, although not desirable, that, due to some system problem, a PLATO file may exist on more than one required master file. If this happens, the first occurrence is the one used.

#### 1.4 Special User Groups

The PLATO application allows special abilities to certain groups, depending on the role performed by that group. These special groups exist and have the same abilities on all systems. Access to system features can be changed through options in lesson "socalutil".

##### 1.4.1 PLATO Support (group "s")

Software Maintenance tracks and fixes problems in the PLATO software. Group "s" needs access to virtually every feature of the PLATO application.

This function is normally performed remotely and people in this group are not considered to be part of a local services organization.

##### 1.4.2 Courseware Services (group "coserv")

Courseware Services manages the published courseware library. Group "coserv" needs access to those utilities

used in the management of that library.

This function is normally performed remotely and people in this group are not considered to be part of a local services organization.

#### 1.4.3 System Controllers (group "p")

System controllers are the people who allocate resources and control accessibility and performance. Group "p" needs access to all utilities required for this function as well as all utilities available to any other member of the services organization. They do not need access to those utilities which are only required as part of software support or courseware management. Where groups "o", "m" and "ps" are a subset of group "p", group "p" is in turn a subset of group "s", in terms of access to all system functions.

The PLATO Configuration Handbook contains information needed by this group to configure the PLATO application and to optimize its performance.

The functions performed by group "p" are not normally performed on a daily basis. The daily functions are handled by the other special groups.

#### 1.4.4 Operators (group "o")

Operators perform those operation functions which are done on a regular (often daily) basis. These include starting/dropping the PLATO application, running usage accounting programs, doing prints and backups, etc.

The PLATO Operations Guide contains the information needed by this group for the day-to-day operation of the PLATO application.

#### 1.4.5 Communications (group "m")

This group includes both communications and customer engineers, that is all people responsible for the maintenance of the hardware. Group "m" needs access to all utilities which aid in the resolution of hardware (usually terminal or communication line) problems.

#### 1.4.6 System Consultants (group "ps")

System consultants are responsible for customer support via TERM-consult or phone. Group "ps" needs access to utilities which:

- \* help the customer solve a problem
- \* help determine if a problem is a hardware or system software problem, and assign it to the appropriate people



The consultant does not normally fix the problem, but determines the nature of the problem and instructs the appropriate person (operator, customer engineer, software maintenance, etc.).

Besides the ability to use TERM-consult, consultants should have "Inspect-only" access to virtually all information on the system. They should not be able to change anything which might affect system performance.

Utilities which are intended for this group are described in the "Consultant Features" section of the PLATO Operations Guide. Consultants will also need to know about many of the other operations functions in this Guide to help with some user problems.

### 1.5 User Features

Users in groups other than the special ones in the previous section are divided into three types: authors, instructors, and students. Refer to the PLATO User's Guide for more information about these user types.

There are also users who are assigned the responsibility of managing the activities or resources available to other users. These users are called:

- account owners
- account directors
- group directors
- notesfile directors

All of these classes of responsible users and the options available to them are described in the PLATO User's Guide.

The options available to all other users are also described in the PLATO User's Guide.

### 1.6 System disk space requirements

The following shows the disk space requirements for the PLATO Lesson Delivery and Authoring system 1. One "part" as used here occupies 35 physical sectors on the disk.

Master file	# parts
sys1	3680
newins	2970
binary	depends on device type used (3680 - 5000 parts)

The following table shows the disk space requirements for the various courseware categories which may be ordered through Control Data Courseware Delivery. One "part" as used in this table occupies 35 physical sectors on the disk.

Cat #	#files	#parts
-----	-----	-----
I	2206	6613
II	939	2578
III	3750	13409
IV	2966	9524

## 2 System Operation

The functions of the computer operator may vary depending upon the type of system being used. This section is intended as an example of what a computer operator's guide might look like. It may be necessary to modify this section at the local site if procedures differ.

### 2.1 Start-up

#### 2.1.1 Deadstart

Deadstart according to standard procedures. These are the normal procedures described in the NJS Operator's Handbook. Sites may have special needs which modify these standard procedures.

#### 2.1.2 Turn Off Broadcast Unit

Before the PLATO application is brought up, ensure that the broadcast unit is not operating by checking that the out-of-service switch is in the "off" position. This only applies to systems which have a CIU. Refer to Broadcast Unit later in this section for more information.

#### 2.1.3 Bring Up The PLATO Application

When the operating system is loaded and MAM is running, the operator types in "PLATO." at the computer console to bring up the PLATO application.

This will bring MASTOR to a control point and it will submit the other jobs which are required to run the PLATO application.

If your system is running on a multi-mainframe system, repeat the process for the other mainframes:

- a. Deadstart the additional mainframes.
- b. Bring up MASTORN by entering "PLATO." on the computer console.

#### 2.1.4 System Check

After the application is brought up, log onto a terminal and run lesson "sysatr" to ensure that the application is functioning correctly.

### 2.2 Shutdown

#### 2.2.1 Start File Dumps

If file dumps are done with the PLATO application active, you should start them prior to any other phase of the shut-

down. Refer to the section on "File Dumps/Backups" for more information.

### 2.2.2 Warning Message

About 15 or 20 minutes before shutdown, send a message to all users of the application telling them that the application will be going down. Include the time at which it will be going down. To send a message to all users, do the following at any PLATO terminal:

- a. Sign on and go to lesson "sysopts".
- b. Select the number for the option titled "Message to all terminals". You will then be asked to type in your message.
- c. When finished typing the message, press NEXT and the message will be displayed at the bottom of all users' screens. Pressing LAB will repeat the message.

In addition, use the option in "sysopts" to set the first line message to display the same message. This message will appear on the first line of the Author Mode display and the signon display to warn users who have missed the message previously sent.

Repeat the above procedure 5 minutes before shutdown.

### 2.2.3 Backout Users

At the scheduled time, all users should be backed out. Backout means that all users will be taken off the application and all of the files they had been working on at the time of the backout will be returned to disk and saved for the next time they sign-on.

To backout all users, do the following:

- a. Sign on, and go to lesson "sysopts".
- b. Select the option titled "Backout Options".
- c. You will see an option to press SHIFT-LAB for a 60-second backout. Do this.
- d. Messages will appear on the display that are sent to all users. After everyone is backed out, press SHIFT-STOP twice to log yourself off.

### 2.2.4 Stop Control Points

To stop the PLATO control points, do the following:

- a. At the computer console type

K,MASL. (cr)

b. Next, at the console enter:

K,STOP (cr)

This drops all PLATO application jobs. It will take a few seconds for these jobs to go away. When they do, go to the next step in the shutdown.

#### 2.2.5 Turn On Broadcast Unit

Turn on the broadcast unit to reflect the time that the PLATO application will next be available to users. This only applies to systems which have a CIU.

#### 2.2.6 Do File Dumps if Required

Do file dumps if they are scheduled at this time. Refer to the section on "File Dumps/Backups" for more information.

#### 2.2.7 Checkpoint Operating System

If it is desired, use the following steps to shut down the operating system. Whether this is done or not should be determined by local site procedures.

a. At the console, enter:

CHECKPOINT SYSTEM. (cr)

b. Check the "E,M." display, by typing:

E,M. (cr)

Wait until the letter "C" does not appear in any column of the status display before proceeding.

#### 2.2.8 Step System

When there is no longer a checkpoint pending for any NDS packs (on the "E,M." display), type the following at the console.

UNLOCK. (cr)  
STEP. (cr)

Next press the deadstart button so that the display is either blank or has the deadstart options displayed.

#### 2.2.9 Finish

The following steps are all determined by local site procedures.

a. Turn off all disk units.

b. Store any tapes or disk packs.

c. Make a log entry.

## 2.2 System Crashes

Whenever there is a problem, no matter how small, the operator should try to determine the source of the problem. Also, an entry should be made into the PLATO logbook, if one is kept, describing the problem and the action taken.

### 2.3.1 System Monitoring

It is the responsibility of the operations staff of the site to detect when the application crashes, and, in the event of a crash, to bring the system back up.

To monitor or test the system, log onto the terminal and execute lesson "sysmtr". Lesson "sysmtr" is a special lesson that tests various critical operations of the computer. As long as the lesson keeps flashing letters, the system is considered to be up and running. Run this lesson often.

When a crash is detected, follow the crash procedures step by step until the application is back up. Once it is up, fill out the log book and go back to lesson "sysmtr".

### 2.3.2 Check Terminal

It is possible that, instead of the PLATO application having crashed, the terminal you are using has broken. If possible, try to verify that the problem is not in the terminal by using another terminal, or by contacting someone else who is at a terminal.

If another terminal is not available, the operator may test his terminal by pressing the clear button at the base of the terminal. A properly functioning terminal should display words similar to "TERMINAL READY" at the bottom of the screen. If this message does appear, the operator should then press SHIFT-STOP several times. If the terminal is working, and the application is running, normal output should appear on the display. You may then sign into "sysmtr" once again.

If the problem appears to be with the application, and not just with the terminal, proceed to the next step.

### 2.3.3 Check PLATO Control Points

A properly working application should have all of the following jobs running. If one of them is missing, something is wrong.

MAS1  
PLA1

FOR1  
 PNI1  
 COA1

Check the system dayfile and/or the running PLATO jobs for B-display or dayfile messages which may indicate an error. Error messages for the PLATO application are documented in the PLATO Operations Guide.

#### 2.3.4 Wait for One Minute

If you just noticed the crash, wait at least a minute before entering anything at the console. During this minute, things may clear up. This is true if a job has dropped, and MASTOR is able to detect it. If this happens, lesson "sysatr" (or whatever else you are using at your terminal) will stop, and the message "PLATO OFF" may be displayed if your system has a CIU. MASTOR may then try to recover the application.

If you see the message "dumping" at any of the control points, a crash has occurred. Wait until the dump has completed before taking any further action.

If MASTOR is able to recover, the "Press NEXT" will appear. Make a log entry, and go back into lesson "sysatr". Note that there may be jobs in the output queue at this time. One or more of these jobs may relate to the reason for the crash. Print them out if possible.

There may be jobs requesting tapes. These jobs are started by the PLATO application when a crash occurs. They will copy information about the crash onto the tape so it can be sent to PLATO Support for analysis. Mount and assign tapes. Copy dumps to tape and send them to PLATO Support. Refer to the section on "Problem Reporting Procedures".

#### 2.3.5 Reload Missing Control Points

If MASTOR is not at its control point, you must kill each of the remaining jobs, and then restart with the standard "PLATO." type-in.

If PLATO is not at its control point, MASTOR should drop and reload all PLATO related control points after a short time. If after several minutes this has not happened, continue with the next step.

If FRAMAT is not located at a control point, it will be necessary to bring it up. To do so, type "X.FRAX(CP=cp)" at the console where cp is the FRAMAT control point. If FRAMAT will not stay up, continue with the next step.

If PNI is not located at a control point, it will be necessary to bring it up. To do so, type "X.PNIX(CP=cp)" at the console where cp is the PNI control point. If PNI will not stay up, continue with the next step.

If CONDENSOR is not located at a control point, it will be necessary to bring it up. To do so, type "X.CONDX(CP=cp)" at the console where cp is the CONDENSOR control point. If CONDENSOR will not stay up, continue with the next step.

If the CONDENSOR is at its control point, but will not condense lessons, try to drop it by typing in "DROP,jsn." (where jsn is the job sequence identifier) and press CR. If it does drop, type in "X.CONDX(CP=cp)" where cp is the CONDENSOR control point. If it will not stay up or still does not condense lessons, continue with the next step.

### 2.3.6 60 Second Backout

If you have reached this step and the application is still not back up, it will be necessary to do a deadstart dump. It is possible, depending upon conditions, to be able to do a backout, even though some other part of the application is not working properly. For instance, if only the CONDENSOR is hung or missing, a backout is normally still possible.

Go to a terminal and try to sign out of "sysmtr" (or any other lesson) by pressing SHIFT-STOP. If this is successful, try to do a 60-second backout. Whether successful or not, proceed to the next step.

### 2.3.7 Turn On the Broadcast Unit

By this step, it is obvious there is a major problem, and that the application will be down for an extended time. Turn on the broadcast unit. Initially set it to the 5 to 10 minute message. If it later appears that the down time will be even longer, change to one of the other messages. This only applies to systems which have a CIU.

### 2.3.8 System Is Hung or Major Problems

If the application will not come up or will not stay up, there are probably major hardware or software problems. You should try to analyze the problem.

If there appears to be a hardware problem, you should contact a Customer Engineer (CE).

### 2.3.9 Shut Down

Go to the "E,M" display, if possible, and check for a "C" in the sixth column of the status display. If there is a "C", wait a minute to see if it will go away. If the console will not allow any input, and/or the "C" will not go away, continue with the procedures.

At the computer console, enter:

```
UNLOCK.      (cr)
STEP.        (cr)
```



If possible.

#### NOTE

By all means, DO NOT enter "CHECKPOINT SYSTEM". This will clear all of the control points, and any dump will become meaningless.

#### 2.3.10 Deadstart Dump

If the problem is not known to be a hardware problem, you should do a full express deadstart dump according to the procedures in the NDS Operators Guide.

#### 2.3.11 Try a Level 3 then Level 0 Deadstart

After taking a deadstart dump, you should try to do a Level 3 deadstart, to recover the contents of the CM buffers containing the error log, system dayfile, and NDS account file. Once that comes up, then do a Level 0 deadstart and bring up the application in the usual way.

If you are unable to deadstart the system successfully, it is likely there is a hardware problem. You should contact a Customer Engineer.

#### 2.3.12 Problem Is Solid

If, after a level 0 deadstart, the application will still not come up or stay up, contact a CE or the PLATJ Services group as needed. Be sure to change the time on the broadcast unit to reflect the fact that the application will be down for an extended period of time if your system has a CIU.

#### 2.3.13 Recording the Interruption

Recording of any interruption of PLATJ service should be done if site procedures require it.

Statistics about application performance are very important, and must be as accurate as possible.

Lesson "stats" contains some statistics which are automatically maintained by the application. One of these is a crash log. Every time the application is brought up or shut down, an entry is recorded in the crash log. If a successful backout was done by the operators, then the shutdown is recorded as a backout. If there was a crash or if the operators did not complete the backout before dropping the application, "stats" records the shutdown as a crash.

#### 2.4 Monitoring Error Messages

Notes file "s0sysmsg" holds messages to the PLATJ service operators. Those things which are not user comments or

messages about errors in a lesson but concern only the correct operation of the application will be automatically sent here instead of going to "sysin." These items are not usually reported as PSRs unless they are software errors.

This file should be monitored on a daily basis.

See the section on "Problem Reporting" for more information.

## 2.5 Broadcast unit

This section applies only to systems which have a CIU.

There is a little black box in the communications area attached to the PLATO CIU. It is called a "broadcast unit". The purpose of this unit is to inform users of the availability of the application. Any time that the application is not up, the broadcast unit should be on. There are four different displays that this unit can send out. They are:

0. Short delay (5 to 10 minutes).
1. Somewhat longer delay, will be back up at tttt (where tttt is the hour on a 24 hour clock).
2. PLATO not scheduled at this time, will be up at tttt (where tttt is the hour on a 24 hour clock).
3. PLATO is not scheduled at this time, will be up at the next scheduled time.

Assuming the application is down, the out-of-service switch should be on. The dial should be set to the position (0 through 3) which best describes the situation.

The time switch should be set as follows:

Message	Time Switch
0	off
1	on
2	on
3	off

If the time switch is off, nothing needs to be done with the time dial. However, if the time switch is on, you must set the dial to time you expect the application to again be available (0000 through 2400).

## 2.6 Prints

PLATO file prints are done as follows:

- a. Go to lesson "prints".
- b. Go to the option entitled "PRINT the Requests".
- c. Cycle through the requests and press SHIFT-DATA to print each file.

- d. Collect the prints from the printer.
- e. Return to the option entitled "PRINT the Requests". For each print you have collected, press SHIFT-LA3 to mark the print as being "DONE". For any print not found (lost or some other reason), press SHIFT-DATA to re-submit.

## 2.7 Preparing a Pack for Use

Before using a disk for any purposes, the disk should be initialized and formatted to remove any possible flaws. If possible, the CE at the site should do this. However, if this is not possible it may be done by a system controller using the procedures described here.

Regardless of who formats the pack to remove flaws, it will always be necessary to initialize the pack using these procedures.

### 2.7.1 Initialize the Pack

When initializing the pack, use the normal NOS procedures to initialize the pack at the console. When initializing, use the following entries:

```
K.FM= a 1-7 character name
K.TY=X.
K.DM=377.
K.SM=377.
K.NC=2.
K.DN=0.
```

The name to choose depends on the use of the pack. Names often used include:

```
scratch - use when initializing pack for the first
           time. A scratch pack should never have
           valuable information on it.

platoe, platob, etc. - these are the active packs
                       which must be available when running PLATO
                       in a production mode.
```

When all of the other parameters on the K-display have been entered, complete initialization by typing:

```
K.GG.      (cr)
```

Refer to the NOS Operators Guide or the NOS Installation Manual for more information on pack initialization.

### 2.7.2 AST

After using the NOS FDKIAT command to set the utility table for the pack so that it has all the same flaws as the factory

table, run MST to see if there are any additional parts of the pack which must be flawed. To run MST, enter the following job under DIS:

```
SUI(277773)
PACKNAM(scratch)    $$ assuming scratch is the pack name
DEFINE(DISK1/R=dt)  $$ where dt is the device type
```

Next go to the "E,M" display to see how many tracks are remaining on pack "scratch". Assuming there are "t" tracks left, multiply "t" by the number of sectors per track for the device type you are using. Use the resulting octal number obtained in the following control card:

```
MST(N=octal number)
```

#### NOTE

If using the MST control statement, there is no need to enter a "D" to designate an octal number, since octal assumed.

This job should test the remaining tracks in the pack. If the job hangs while doing the initial disk write of MST, too large an octal number must have been entered. This frequently happens if "t" was obtained before defining the file "disk1".

#### 2.7.3 FORMAT If Necessary

If MST produces any errors, either format the pack using the NDS FORMAT command or have a CE format the pack.

#### 2.7.4 Reinitialize the Pack

After running MST and formatting the pack, reinitialize the pack with its operational name. The pack is now ready for use with the PLATO application.

#### 2.7.5 Summary of Disk Preparation

The following steps summarize the process used in preparing a pack to be used with the PLATO application.

1. Initialize the pack as a "scratch" pack.
2. Run MST to see if there are any areas that need to be flawed.
3. Format the pack if errors are found.
4. Initialize the pack using its operational pack name.

#### 2.8 Disk Error Recovery Procedures

The procedure for recovering from disk errors varies depending on whether the errors are affecting a single file, an entire master file, or the entire disk.

Throughout the procedures, the following assumptions are made:

- a. That a disk pack may be bad, and should be replaced. If uncertain, it should be assumed to be bad. If the pack is good, alter the procedures slightly so that the same disk pack is reused, rather than initializing a new one.
- b. The procedure is done with the application unavailable to users. If users are allowed on, turn off file operations and warn the users to minimize any work that might be lost.

### 2.8.1 Single File Error Recovery

If a disk error is affecting only one or a few files, it is desirable to recover the disk in a manner such that only the bad file(s) have to be backed up, and that work done by users on other files on the corresponding disk pack will not be lost. There are also certain error conditions which may have been unnoticed for some time, and the dumps may also be bad. This is the only procedure which will successfully restore the disk.

- a. Load the application and backout all users.
- b. Create a temporary master file using MFCREAT and load it using lesson "ldr".
- c. Use lesson "transfer" to copy all files on the master file containing the bad file(s) to the temporary master file.
- d. Lesson "transfer" should halt on the disk error. Just skip the file and it will be logged for later use.
- e. Drop the application.
- f. Purge the master file which contained the bad file(s).
- g. Use PFDUMP to dump the rest of the disk pack which previously contained the bad files.
- h. Initialize a new disk pack for use.
- i. Use PFLDAD to load the master files just dumped.
- j. Copy the temporary master file to the new disk pack.
- k. Use CHANGE and MFALTER commands to change this master file to the name and type previously used by the master file containing the bad files.
- l. Load the application again for normal use.
- m. Use the "transfer" log from the earlier step to

determine which files need to be backed up individually.

## 2.8.2 Master File Recovery

### CAUTION

Execute these procedures ONLY when told to do so by PLATO Services personnel.

If a problem is suspected with permanent files, immediately contact PLATO Services personnel. If there is a need for reloading files, these are the procedures to be followed.

### 2.8.2.1 Binary Master Files

Recover binary master files by purging the bad master files and recreating them using MFCREAT.

### 2.8.2.2 Other Master Files

If an entire master file is bad, and the single file recovery procedures cannot be used, recover the master file as follows:

- a. Drop the application.
- b. Use NDS COPY commands to dump the good master files to tape or a scratch pack.
- c. Initialize a new disk pack and use NDS COPY commands to copy the files dumped in the previous step to the new pack.
- d. Use standard backup procedures, for this site, to recover a bad master file.
- e. Load the system for use.
- f. Run the "recovery integrity tests", described in a later subsection, to insure that recovery is complete.

## 2.8.3 Disk Pack Recovery

### 2.8.3.1 System Pack Recovery

If the system pack is bad, reload according to following procedures:

- a. On the EQPDECK display during deadstart, enter an INITIALIZE command for the system pack. The correct entry is:

```
INITIALIZE,AL,eee.      (cr)
```

where "eee" is the cST ordinal for the system pack.

- b. Reload the system pack using standard procedures.

which usually include a PFLDAD from previous dumps.

- c. After reloading the pack, use "X.ISF." to update the validation file.

#### 2.8.3.2 Pack Recovery

If an entire pack is destroyed, recover it as follows:

- a. Drop the application, if it is still up.
- b. Initialize a new pack for use.
- c. Use standard load procedures for this site to recover all master files for this disk pack.
- d. Load the application.
- e. Run the "recovery integrity tests" described in a later subsection to ensure that recovery is complete.

#### 2.8.4 Recovery Integrity Tests

Since a file may move from one master file to another (and even to another disk pack) during the course of a day, reloading files can cause the following problems:

- a. Duplicate files: (e.g., suppose file "abc" was on amast at the last dump, but is on bmast when a reload of amast is done).
- b. Missing files: (e.g., Suppose file "abc" was on bmast at the last dump, but was on amast when amast had to be reloaded).

The following procedures must be done to correct these possible problems:

- a. Run "installu" option to "Search for duplicate files". Scan all active master files. Duplicate files should be deleted or renamed.
- b. Run "accountu" option "Account directory and file checks". Run this option for all accounts. Errors are logged in "accerrlog", but most errors will be corrected automatically, or by the next step in this procedure.
- c. Run "accountu" option "Search for files not in any account". Errors are logged in "accerrlog". Take appropriate corrective action for each error shown.

#### 2.9 Copying PLATO Files to Tape

Individual files cannot be copied to tape, but rather an entire master file must be copied. The procedure for copying individual files to tape is as follows:

1. Create a temporary master file of the appropriate size using MFCREAT. The master file should be type "master".
2. Load the master file using lesson "ldr".
3. Use lesson "transfer" to copy the files from the required master files to the temporary master file.
4. Unload the master file using "ldr".
5. Use standard NOS commands to copy the master file to tape or use PRDC/MFTCOPY, whose parameters are described in another section of this document.
6. Purge the disk copy of the temporary master file.



### 3 Network Management

#### 3.1 NAM

Refer to the NJS Operator's Guide for instructions regarding NAM. This manual contains instructions about:

- a. loading NAM.
- b. commands that may be issued to NAM to control lines, terminals, and applications.
- c. dropping NAM.
- d. error messages issued by NAM.

##### 3.1.1 Rebuilding the 2550 Load File

Refer to the NJS Version 2 Installation Handbook. Follow the Initial Setup procedures and then refer to the section on CCP Installation Procedures.

Corrective code for CCP should be placed in file UCCP, as described in the section on CCP Installation Procedures. Place any corrective code from the PLATO binary tape (VSN=PLAT1A) into file UCCP, along with any other recommended or local modsets. See the PLATO Installation Guide for the format of this tape.

#### 3.2 PLATO-NAM Interface (PNI)

The PLATO/NAM Interface program (PNI) directs the traffic between NAM and PLATO. NAM provides a simple interface to the network, but does not provide the type of interface required by PLATO. PNI will perform the functions required to interface NAM to PLATO.

##### 3.2.1 Rebuilding the Resident Load Files

PLATO ASCII terminals such as the IST2, IST3 and CDC-721 have no ability to perform any function until a resident is loaded. This resident must be loaded before the terminal can log into NAM. To perform this function, a downline-load is initiated by PNI when a terminal requires its resident to be loaded.

PNI reads the terminal resident binary data stored in a NJS permanent file and transmits the data via NAM to a loadable terminal in transparent mode.

The terminal resident load files are built as follows:

- a. Execute lesson "sCpnllf".
- b. Load the resident file for each terminal resident to

be supported.

These files are also built during initial installation of the PLATO application.

### 3.2.2 PNI with AIP Trace

A version of PNI, loaded with the NAM AIP trace routines (NETIJD library) exists for the purpose of tracking communications problems. This version is not included with the standard PLATO Release Materials, but may be requested if needed.

PNI recognizes CFD commands which allow initiation and suspension of logging of both data and supervisory messages as well as routing of the AIP trace file for processing.

PNI recognizes the following CFD type-ins:

DBG,ON.	initiates logging of data and supervisory messages
DBG,OFF.	suspends logging of data and supervisory messages
DBG,OUT.	causes the debug log file "ZZZZZDN" to be routed to the input queue

When PNI is initialized, message logging is initially OFF.

## 3.3 The Network Database

A network database describing all aspects of your communications network may be built using lesson "pnet". This database may be quickly scanned to aid in the isolation of network problems.

A HELP-section in lesson "pnet" provides additional information.

### 3.3.1 Verifying the Database

Due to system errors, it is possible that the network database may become damaged. This is especially true if the system aborts while you are editing. In addition, if the number of sites is increased, the database must be adjusted to compensate for the additional ports. The procedure for verifying the database is as follows:

- a. Execute lesson "pnet".
- b. Press LAB for more editing options.
- c. Run the option to "verify the network database".
- d. If errors occur, obtain a backup of the database files and commands and redo any editing done since

the backup copy was made. The affected comsons are block "pnet" of file "sysfile". The affected files are "sGnetwk" and "sGnetwk1".

### 3.3.2 Scanning the Database

The entire network database may be scanned, to find a particular equipment or station, via the "network arrow" option in "pnet". You may also cycle through all equipment in a particular location.

## 4 PLATO Account Management

All PLATO files belong to a PLATO account. Files are managed through lesson "accounts". Most of this file management is done by the account owner or directors with no need for system controller help. The only thing the system controller normally needs to do is to control the overall creation or deletion of the accounts themselves. The files within an account are generally of no concern to the system controller. The account owner can designate other users as account directors through options in the account access list.

### 4.1 Account Creation

To create an account, go to the system options in lesson "accounts". The request for an account should include the following parameters:

1. Account name
2. Number of file spaces
3. Number of subscriptions
4. Name and group of the account owner

After creating the account, choose the system option to edit the account. The lesson access bits must be set to enable the account to access courseware in the published and special libraries.

### 4.2 Editing an Account

Normal account editing directives (file creation, etc.) are done by an account director or designates through the normal account options. There is also a system option to edit an account which allows certain things to be edited. They include:

- a. the account owner's sign-on
- b. file parts allotted/used/NL  
(Note: The term "NL" refers to a feature which is only available on Control Data Services systems.)
- c. parts not charged
- d. number of subscriptions
- e. archive rights
- f. whether or not the account is allowed to create groups with authors
- g. whether or not the account is a publishing account (only people in groups "s" and "coserv" can set/change this)
- n. transmit files to NDS (TRANSMIT feature)  
For an account which is enabled to use TRANSMIT, access for individual users must be granted in the account

access list. The user must have a valid NOS user name and password on the destination system. (NOTE: It is not necessary for the user to have batch access enabled in his user record.)

- i. networking options (options relating to the ability to send files and/or notes to another system if the PLATO Inter-System Link is available)
- j. NL options (Control Data Services systems only)
- k. the lesson access bits that control which published files this account may access
- l. the account codewords

#### 4.3 Destroying an Account

To destroy an account, follow this procedure.

- a. Any files not to be destroyed should be moved to another account or saved on tape by creating a master file of all files in the account and copying the account to a tape.
- b. Destroy the account using the system option to destroy an account.
- c. Use the system option to edit print access, and remove this account from the list of accounts which have print access.

#### 4.4 Renaming an Account

The steps to rename an account are as follows:

- a. Choose a time when there are very few users on the application. This is necessary to avoid attach conflicts.
- b. Use the system option in "accounts" to rename the account.
- c. Use the system option in "accounts" to edit print access, and rename the account there as well.

#### 4.6 The File Management Log

The "file management log" is used to record all operations for all accounts on the system. Both operations by normal account directors (e.g., create a file, lengthen a file, etc.) and operations by system controllers (e.g., create an account, change the number of subscriptions, etc.) are recorded.

The size of each log file is fixed. The log is always circular, so as new operations are logged, the oldest operations in the log are overwritten. If more history of account actions

is desired, the number of files in the Log may be increased by changing the "nalog" parameter in the PLATO configuration file (see the PLATO Configuration Handbook for more information).

There is an account system option to print the file management log. With this option, a set of questions are asked to determine exactly what portion of the log is to be printed. Once all questions have been answered, a batch job is submitted to generate the print.

#### 4.7 Account Restoration

During normal operations, accounts should remain intact as long as normal procedures are followed. However, when new files are added to the system from a tape, or after certain system crashes, some of the accounts may become damaged. If this happens, it will be necessary to restore the damaged accounts. Either a complete or partial restoration may be done depending upon the situation.

##### 4.7.2 Partial Restoration

A partial restoration assumes that all files currently in an account belong there, and that there exist files with account pointers for this account that are not in the account already. The procedures for a partial restoration are as follows:

Choose the option in lesson "accountu" to "Update an account file name table". Use this option to fix the desired account.

##### 4.7.3 Complete Restoration

During complete restoration, the account is destroyed and rebuilt file by file.

Go to the system option in "accounts" and edit the account to be restored. Write down the following information.

1. name and group of the account owner
2. number of subscriptions
3. number of file spaces allocated to the account
4. lesson access bits which were set
5. whether or not there were codewords set

Go to lesson "accounts" and destroy the account. When asked if the pointers are to be zeroed in the files themselves, select "n". By leaving the account pointer in each file, the file may be restored to the account later.

Re-create the account in the system options in lesson "accounts". Use the information written down before destroying the account in order to properly assign the account owner, file spaces, etc.

Rebuild the directory in the following manner:

1. Go to lesson "accountu" and choose the option called "Update an account file name table".
2. From there, choose to form a new list of files.
3. Finally, specify which account is to be updated.
4. Reset codewords or ask the account owner to do so.

#### 4.8 Emergency Actions

When attempting to recover from some types of problems with the accounts system, it may be necessary to do file operations while the accounts common is missing or partially or totally overwritten by bad data.

Lesson "operator" has an option, "disconnect accounts common", which will allow file operations to be done without updating the accounts common.

#### NOTE

Since file operations done with this option enabled will result in files which are not listed in any account, this option should only be used in extreme emergency situations, and all accounts verifications in lesson "accountu" should be run when the system is back to normal.

#### 4.9 TRANSMIT Utility Usage

The TRANSMIT feature allows users who may edit an account to copy the contents of various file types into NDS files for processing by batch jobs.

Before using TRANSMIT for an account, the following steps must be performed.

1. Enable TRANSMIT for the account:
  - a. Use the system options in "accounts" to edit the account which is to be allowed to use the TRANSMIT feature.
  - b. Set the "Transmit files to NDS" option to "yes".
2. Edit the account access list to permit use of TRANSMIT.
  - a. When editing the account access list options for an individual, choose "Special Options".
  - b. Set the "Transmit PLATO files to NDS" option to "yes".

More information is available on this feature, when it is selected, while editing the account with the normal user options.

#### 4.10 Lesson "operator"

Lesson "operator" is used by system controllers for many functions related to maintaining PLATO accounts.

- a. log of all file operations  
This allows the system controller to inspect the file management log for any account.
- b. lesson information  
This allows the system controller to inspect the author information for a file.
- c. master files on and disk space left
- d. detach file/signon or clear security count  
This allows the system controller to detach a file or user record which has been left attached by a system error or to clear the security count for a user.
- e. master file to master file PLATO file transfer
- f. file options  
This allows the system controller to perform file management actions for any account.
- g. copyover file contents  
This option allows the system controller to copy the contents of a file to another file when the two are not in the same account.
- h. recover a file lost during a system crash  
If a user is lengthening or shortening a file when the system crashes, the original file may have been renamed to a temporary name. This option allows the system controller to rename the temporary file back to the original name.
- i. disconnect accounts common  
See the "Emergency Actions" section.
- j. edit/inspect system options access list



## 5 "Runner" Lesson Management

There are a number of functions which must be periodically performed on the PLATO system, such as checkpointing of user records, checking for inter-system link requests, checking for full binary master files and deleting old binaries to recover space, etc. These functions are automatically performed by lessons under the control of the PLATO system which are known as "runners". These lessons are automatically signed in when the PLATO system is loaded and sign out when they have finished. Periodically, needed lessons are again signed in automatically. The list of lessons which are run in this manner and their attributes are managed through lesson "runnersys".

The runners which normally execute on all systems are listed below along with their "attributes". The attributes are described in lesson "runnersys". Other lessons may be added at the local site's option.

In addition to those lessons listed below, two other programs always run at the start of the runner program stations, "stats1" and "runrexec". These programs are automatically started by the system without being listed in "runnersys" and are always present. "stats1" produces the data which may be viewed in lesson "stats". "runrexec" is the driver which schedules the other runner programs.

### 5.1 Runner Lessons

alarm: process pending alarm messages.

```
cycle = 1
restart = 1
priority = 63
```

binary: scan binary master files for old binaries to delete.

```
cycle = 30
restart = 30
priority = 30
```

checkpt: checkpoint comons and student records to disk.

```
cycle = 1
restart = 1
priority = 63
```

enforcer: monitor site activity and back out users in restricted lessons (optional lesson).

```
cycle = 1
restart = 1
priority = 03
```

notesys: automatically clean up full group notes files.

cycle = 10  
restart = 10  
priority = 30  
schedule = 2 hours during non-peak usage

pnotes: distribute personal notes from remote systems  
(only for systems using the PLATO Inter-system Link  
feature).

cycle = 3  
restart = 2  
priority = 30

s0cpustat: collect CPU usage statistics over time (optional  
lesson).

cycle = 30  
restart = 5  
priority = 30

s0notrun: distribute group notes from remote systems  
(only for systems using the PLATO Inter-system Link  
feature).

cycle = 3  
restart = 3  
priority = 30

s0pnetrn: send message to a set of users as defined in  
lesson "pnet" (optional lesson).

cycle = 5  
restart = 5  
priority = 63

s0rhp: PLATO Inter-system Link driver (only for systems  
using the PLATO Inter-system Link feature).

cycle = 5  
restart = 1  
priority = 30

utility: scan all files on system and log files which are  
found to be invalid (optional lesson).

cycle = 60  
restart = 5  
priority = 30  
schedule = non-prime time only

## 6 File Archive Management

Archiving is a process of off-line, long-term file storage. It serves two main purposes.

- a. It provides storage for files that are not used frequently enough to justify the cost of permanent on-line storage.
- b. It saves the state of a file for future reference.

### \*\*\*\*\* NOTE \*\*\*\*\*

When a user archives a file, it is expected to be safely stored. Consequently, archiving must work all the time, under all conditions.

This feature has not been fully tested and is not officially supported by Control Data Corporation (CDC).

Extensive tests are being conducted on this feature prior to its official release. Because of the importance of its proper operation, CDC will continue to conduct tests prior to announcing/enabling this feature in the domestic service systems environment.

Each site is advised to also test this feature in their own environment prior to enabling it, addressing the following questions.

- Do you wish to offer the service?
- Do you have the proper hardware and personnel resources to offer the service?
- If there is to be a charge, do you have an appropriate billing mechanism in place?
- Are you willing to accept the risk of providing an unsupported feature to your users?

### \*\*\*\*\* NOTE \*\*\*\*\*

#### 6.1 General Description

Archive rights are the tracking mechanism used to control and record use of off-line disk space for archived files. Operations people assign archive rights to an account based on the contract for that account. A charge of 1 archive right per part is made for archivals and 3 archive rights per file for retrieval requests.

Once an account has archive rights, the account owner and account members may request that certain files be archived and, if desired, retrieved at a later time. When a user requests that a file be archived, an exact copy of the file

is made on line. The file is set to type "archive" and is named using the -day- command in order to give the file a unique name. The file is moved into account "s0arch" where it remains until it is copied off line by operations. When the copy is made, the user then has the option of destroying the original on-line copy.

When a user requests that an archive file be retrieved, an empty file is created to reserve the desired name and number of parts in the account. The file is set to type "retrieval". It resides in the user's account, but it cannot be destroyed until retrieved by operations.

On a regular basis, operations personnel will complete these archival and retrieval requests by loading archive disk packs, moving the archives off line, and bringing copies of the requested retrievals on line. Once archived on a disk pack, the archive remains on the disk pack for a period of time determined by the site. After this time, the pack is recycled, and the archive lost, unless the owner has retrieved the file to be archived again.

## 6.2 Disk Pack Naming Scheme

Archive disk packs are initialized according to procedures documented in the "System Operation" section. The archiving utilities will tell the operator the name to assign the archive pack whenever a new pack is needed. The operator must use the name requested. Once a pack has been assigned as an archive pack, no other use should be made of the pack until the archives on that pack are no longer needed.

The names which the archiving utilities will request to be assigned will use the format "rrrss" where "rrr" is the system routing ID and "ss" is a sequence number from "aa" to "zz".

Master files are created automatically by the archiving utilities. The names are all 6 characters long, with the first 5 being identical to the disk pack name and the last character being a letter from "a" to "z". A single archive disk pack could contain up to 26 archive master files, but current disk size limits this to a smaller number.

## 6.3 Setting Archive Rights

Before an account may use archiving, it must have "archive rights". These are set as follows.

- a. Go to the system options in "accounts".
- b. Edit the desired account.
- c. Set the desired number of archive rights or set "deferred billing" if you want the account to have unlimited use.

#### 6.4 Archive Processing

Archive files are copied off-line as follows.

- a. Execute lesson "archiver".
- b. Press SHIFT-HELP to begin processing archive files.
- c. As archive packs are needed, requests will be made to mount them. If there are no existing archive packs or if all current packs are full, then the operators will have to initialize a new pack having the name specified by "archiver".
- d. After the desired pack is mounted, "archiver" will automatically jump to "ljr" to load archive master files and the entire archive process should be automatic.
- e. Upon completion of archiving, it is usually convenient to do retrieval processing. (Refer to the section on "Retrieval Processing".)
- f. It is crucial that a dump be taken of the archive pack as soon as possible after archive processing is complete. Such a dump may be done using PFDUMP and is usually done to tape.

#### 6.5 Retrieval Processing

Archive retrieval processing is done as follows.

- a. Either initiate after archive processing is complete, or process as follows.
  1. Execute lesson "archiver".
  2. Press SHIFT-LAB to begin retrieval processing.
- b. As packs are needed, lesson "archiver" will automatically request them. Operators must then mount the requested packs.
- c. Beyond the mounting of packs, the rest is automatic. The lesson will write a message when processing is complete.

## 7 File Dumps/Backups

### 7.1 Dump Sets

The backup system can keep track of 30 sets of dumps. These dump sets are referred to as "slots". Each site may set up its own dump cycle.

One example of a dump cycle would be to use a different dump set for each of 30 consecutive days, and at the end of those 30 days cycle back to the first dump set. The slots for this type of cycle would go in the following order: 1, 2, 3, 4, ..... 29, 30, 1, 2, 3, 4, .....etc.

Most sites would probably require a backup capability of a longer duration. An example would be a system of six daily dumps and a weekly dump. Since there are a maximum of 30 dump sets, this system could continue for 24 weekly dumps (30 total - 6 reserved for daily dumps). The six daily dump sets would be reused each week while the 24 weekly dumps would not be reused for 24 weeks. The slots for this type of dump would be as follows: 1,2,3,4,5,6,7; 1,2,3,4,5,6,8; 1,2,3,4,5,6,9; ..... 1,2,3,4,5,6,30. After using dump set 30, the cycle would start from the beginning.

#### NOTE

In the example, slots 1-6 are the daily sets which are reused each week, and slots 7-30 are the weekly sets which are only reused after 24 weeks.

### 7.2 Dump Phase Overview

The "dump phase" of the PLATO backups system begins when an operator enters "X,BACKDMP." at the computer console. The communications buffer (file name = CMBUF) contains a record for each master file dumped. Initially, it is an empty file, and is empty at the beginning of each cycle.

Procedure MFNX is called to attach all required master files. Procedure MFDX is then called to dump all required master files to tape or disk.

The dump master file program (CJPFMF) is assigned one or more master files to dump to a medium (tape or disk). Before dumping the master file(s), the communications buffer is searched to make sure that the requested master file has not been dumped before. If it has been dumped, this dump request will be skipped. This allows restart of the dump cycle from the beginning (if there are problems in dumping) as all previously dumped master files will be skipped.

After each master file is dumped to a designated tape or disk, an additional record is written to the communications buffer. Each record will contain information about the dump, including the master file name, master file pack name, date and time

of the dump, VSN or pack name of the dump medium and tape EST ordinal, if tape was used for the dump medium. Each PLATO file on the master file will have information showing the account name, file name, size and type of file.

After all master files have been dumped, the data in the communications buffer, called the "dump directory" and the "audit trail", are processed by programs BACKONE and BACKTWO to produce a sorted index to where all PLATO files are to be found on the dump tapes/disks used. This index is saved in PLATO files with the BACKCOPY program for use in recovering individual PLATO files.

### 7.3 Initializing the Slot Table

The dump cycle slot table determines the type of dump cycle to be used at the site. It is initialized from the computer console as follows:

- a. X.BACKMOD.
- b. K,jsn. (Assign K-display to the job called "jsn".)
- c. K.ST (Select to see the slot table.)
- d. K.DA=n  
where n = the number of dumps in the daily cycle.
- e. K.WK=n  
where n = the number of dumps in the weekly cycle.
- f. K.SS (Set up slot table using DA and WK.)
- g. K.wR (Write slot table to disk.)
- h. K.END (Exit.)
- i. X.BACKCOPY. (Write changes to PLATO files.)

### 7.4 Setting Jp Dump Directory Datasets

The number of dump directory datasets may be changed by the following procedure:

- a. Add the correct number of datasets in account "s0files". The names are "jCa", "dCa", "dCb", etc. See the "PLATO Bill of Materials" section of the PLATO Installation Guide for specifications for these files.
- b. Run the following job at the computer console:
  1. X.BACKMOD.
  2. K,jsn. (Assign K-display to the job named "jsn".)

3. K.END=n  
where "n" is the desired number of datasets.
4. K.WR (write changes to disk.)
5. X.BACKCOPY (Copy changes to PLATO files.)

### 7.5 Changing Master Files to be Dumped

The following procedure describes how to set up or change the list of master files to be dumped:

- a. Update the list of master files to be dumped with the following job run at the computer console:
  1. X.BACKMOD.
  2. K,jsn. (Assign K-display to the job called "jsn".)
  3. K.RP (Display master files to be dumped.)
  4. K.n=#####  
where n = slot number to change  
##### = master file name for slot "n" or blank to clear slot "n".
  5. Repeat the "K.n=#####" entry for each slot you want to add or change.
  6. K.WR (write changes to disk.)
  7. K.END (Exit.)
  8. X.BACKCOPY. (Copy changes to PLATO files.)
- b. Update PROC/MFDX so that each master file to be dumped is included in a COPYMF statement. The format used in the statement will determine if the master file is to be dumped to tape or disk, and the number of master files dumped to each device. Refer to the "MFDX" section in the "Backups Utilities" section for more information.

### 7.6 Writing a Message In the VSN Table

It is possible to assign a message corresponding to any one dump device (a single pack or tape) or to a group of devices (e.g., an entire dump cycle). This is normally done if a particular device is bad (e.g., tape parity errors), or to indicate that a particular dump cycle has been moved off site. A user requesting a backup will see this message, and will know in advance if the backup is available.

To write such a message, run the following job at the computer console:



- a. X.BACKMOD.
- b. K,jsn. (Assign K-display to the job called "jsn".)
- c. K.VS (Select to see the VSN table.)
- d. K.SL=n or K.n  
Use these entries, if desired, to control which entries are displayed.
- e. K.M=text  
where "text" is the message to be entered.
- f. K.m= or K.m=n  
where "m" and "n" are entries in the VSN table.  
This option stores the message at these table entries.
- g. K.WR (write changes to disk.)
- h. K.END (Exit.)
- i. X.BACKCOPY. (write changes to files.)

## 7.7 File Dumping

To dump all files, do the following:

- a. Turn on a tape unit from the computer console.
- b. Drop the PLATO system, if desired. Otherwise, go to the "accounts" system options and turn off all file operations.
- c. At the console, type "X.BACKDMP." or "X.BACKDMP(SV=5)". The job should flash "REQUEST RFDUMP" at its control point.
- d. Mount a LABELED tape. The label may contain anything, but it should be a different name from any of the tapes previously used, unless it is the time to reuse tapes from a previous dump. The backup program will read the VSN and save it in its database.
- e. Assign the tape by typing:  

```
ASSIGN,jsn,nnn.
```

at the computer console, where:  

```
jsn = job sequence number
```

```
nnn = the EST ordinal of the tape unit
```
- f. Continue assigning tapes until the job finishes. If

a tape fails due to read/write errors, and the program aborts, do the following:

1. Get a new labeled tape.
  2. Re-enter the proper "X.BACKDMP." at the console. It will skip to where the problem occurred and start dumping again.
- g. A dayfile will be printed. It is crucial that this dayfile be scanned for errors as it is the only place that certain error conditions will be shown. If an error is found, take appropriate action.
- h. Use the systems options or "accounts" to turn file operations back on, if they were turned off.
- i. After dumping all packs, the system pack should be dumped. Do this as follows:
1. Drop the application, if it is running.
  2. Do all DFTERMs to terminate account file, dayfile, etc. (The application must be down when the account file is terminated.)
  3. Dump the system pack using site procedures. Usually, this is done via PFDDMP.
- j. Reload the PLATO application.
- k. Copy the dump information from NJS files to PLATO application files by entering "X.BACKCOPY." at the computer console.

### 7.3 Recovering a Single File

To recover a single file, do the following:

- a. Enter lesson "backups" and go to the "system options".
- b. Go to the "operator options".
- c. Choose option to "process backup requests".
- d. Choose the option to "inspect/submit requests from all accounts".
- e. Submit a request by typing "nn" followed by DATA, where "nn" = the number corresponding to the request. The message "SUBMITTED" will be shown and the status of the request will change to "S". If the submit fails, a message with "zreturn" will be shown. Refer to the -submit- command in lesson "ajds" for the meaning for the value of "zreturn" shown.
- f. The submitted job will rollout waiting for the proper

tape. Consult the "E,P." display to see the VSN of the tape being requested.

- g. Mount the requested tape with the write ring out.
- h. The job will find the tape, restore the requested file, and then unload the tape. Upon completion, the message "all done" will be displayed in the job dayfile.
- i. Return to lesson "backups" and look at the status of the request. If the file has been correctly returned to the PLATO system, an R will be displayed. The user does not yet have access to the file.
- j. Enter "nn" followed by DATA (where "nn" is the corresponding request number) to allow the user to access the file. The request status will change to "\*".
- k. If it is decided to remove this entry from the log, enter "nn" and press SHIFT-HELP to remove the request.

### 7.9 Recovering a Master File

An entire master file may be recovered as follows:

- a. Either drop the application or unload the master file to be recovered.
- b. At the computer console, enter the following control statement with the proper parameter substitutions.  

```
X.RECDVMF(MF=master file,PN=disk pack,R=device type)
```
- c. If the master file does not already exist on this disk pack, the job will halt with the following message:  

```
WAIT. *GD* TJ DEFINE FILE
```

This is a precaution in case the reason the old master file cannot be found is because the wrong pack name was entered. To proceed, enter "GD,jsn." at the computer console (where jsn is the job sequence number). To stop, drop the job by entering "DROP,jsn."
- d. The job will now request the K-display. Assign it by typing "K,jsn." at the computer console (where jsn is the job sequence number).
- e. Enter the pack name desired and the slot table number for the data desired or the data desired.
- f. Enter "GD." to begin processing.

### 7.10 Backups Utilities

The following sections describe the utilities used by the

file backups procedures.

### 7.10.1 BACKCPY

This program attaches the dump directory, audit trail and parameters files and copies the information from NOS permanent files to the equivalent PLATO files. The PLATO application must be running when this command is used.

The control statement format is:

BACKCPY.

### 7.10.2 BACKDMP

This procedure is called to dump the PLATO master files. It attaches all master files via a call to procedure MFNX, dumps them via a call to procedure MFDX, and merges the communications buffer with the dump database. This procedure requires that SORT/MERGE be available on the system. The default SORT version used is SORTMRG V4, but SORT5 may be specified with the SV parameter.

The control statement format is:

BACKDMP(VSN=vsN,SV=sv)

where:

vsN = parameter to be passed on to PRDC/MFDX  
The default value is null. Since MFDX is site-specific, personnel at each site may determine if they want to use it. A common use would be to use VSN as the prefix to the tape VSNs used in the COPYMF control cards.

sv = sort version (4 or 5, default is 4)

### 7.10.3 BACKLIB

This program gives a listing of the audit trail on file DJTPUT. The audit trail and parameters files are attached, the current slot value is displayed, and all audit information is formatted for the printer.

When an entry has a slot value that is the same as the current slot, the string "\*\*\*next\*\*" is shown in the listing to indicate that this tape will be obsolete with the next dump, and this could be reused.

The control statement format is:

BACKLIB(SK)

where:

SK is included if you want to skip listing for

\*\*\*next\*\* tapes.

#### 7.10.4 BACKLST

This program extracts data from the dump directory. It is intended to be used when a file needs to be recovered, but its account of residence is unknown. Other information can also be extracted.

The control statement formats are:

```
BACKLST(FN=pfX,AN=acc,TY=typ,SL)
```

```
BACKLST(Z)/FN=pfX/AN=acc/TY=typ/SL
```

where:

- pfX = first letters of any file name to be listed  
The default is to list all file names.
- acc = account name of files to be listed  
The default is to list files in any account.
- typ = PLATO file type of files to be listed  
The default is to list all file types.
- SL = short listing flag - 60 character lines
- Z = allows parameters after the closing parentheses  
This is required for parameters longer than seven characters.

The information formatted into the output file includes the account and file name, and file size and type for selected entries.

#### 7.10.5 BACKMOD

This program is used to make changes to the backup parameters, slot table, master files to be dumped table, or to place messages about tapes in the VSN table.

The control statement format is:

```
BACKMOD.
```

The job will allow parameter values to be changed via the K-display. An entry of "KK." will give an on-line listing of all parameters.

K-display parameters available at all times:

GI	Display general information page.
INITIALIZE	Initialize the entire database.
RP	Display master files to be dumped table.
ST	Display slot table.
VS	Display VSN table.

RR            Re-read parameters from disk.  
 WR            Write parameters to disk.  
 STOP         Exit from program (no write to disk).  
 END          Same effect as STOP.  
 +            Page through entries.

On General Information display:

ND =        Change number of PLATO dump directory datasets.

On Master Files to Be Dumped display:

n =        Store master file to dump at entry "n".

On Slot Table display:

m =        Store cycle number at entry "m".  
 DA =        Number of days in daily cycle.  
           ("DA" + "WK" must be less than or equal to 30.)  
 WK =        Number of days in weekly cycle.  
           ("DA" + "WK" must be less than or equal to 30.)  
 SS =        Set up slot table according to "DA" and "WK"  
           parameters.

On VSN Table display:

M =        Set message to be stored later.  
 K =        Start display at entry "K".  
 SL =        Display entries with selected slot value. This  
           also limits which slots to store messages at.  
           Use "SL=0" to select all slots.  
 n =        Store message at entry "n".  
 m-n =      Store message at entries "m" through "n".

#### 7.10.6 BACKONE

This program performs the first phase of the backups database merge process. This involves the following:

- Attaches the communications buffer, audit trail, and parameters file
- Extracts the current slot value from the parameters file
- Clears the VSN table of all entries using this slot
- Clears the audit trail of all entries using this slot
- Reads the communications buffer

Each record in the communications buffer contains a record for each master file that has been dumped. As each record is read:

- the audit trail is updated to reflect this master file

- the VSN table is updated to reflect this VSN
- an entry for each PLATO file found, with account name, file name, file type, file size and VSN, is written to a temporary file for later use
- the record is checked against the master files to be dumped table in the parameters file to insure that all master files were dumped
- After all records on the communications buffer have been processed without errors, the audit trail and parameters files are rewritten with the new information.
- Information written to the output file includes:
  - old audit information (those entries removed from the audit trail)
  - new audit information (those entries added)

The control statement format is:

BACKONE.

#### 7.10.7 BACKTWO

This program performs the second phase of the backups database merge process. This involves the following:

- Uses the parameters file and the PLATO file data from BACKONE (after it has been sorted by SORT/MERGE)
- Attaches the dump directory
- Extracts the current slot value from the parameters file
- Compares the dump directory and the sorted PLATO file information entry by entry (account name, file name, file type and size)

If an old entry in the dump directory is found without an equivalent new entry, the current slot information is cleared. If all slots are empty, this entry is discarded. Otherwise, it is written to the new dump directory file.

If a new entry is found without an equivalent old entry, a new entry is created with the current slot information.

If an old and new entry are found to be equivalent, a merged entry with updated slot information is written to the new directory file.

As information is written to the dump directory, a first-level index is created in a portion of the parameters

file called the "look-aside" buffer.

- After all entries have been written to the new dump directory file:
  - the new dump directory file is rewound and is copied over the original file
  - the slot pointer is advanced to the next slot
  - the parameters file is rewritten

The control statement format is:

BACKTWD.

#### 7.10.8 BKSTART

This procedure destroys the old backups communications buffer and starts a new one. It is used only when part of the backups dump cycle has been done and you wish to re-start the entire cycle from the beginning.

The control statement format is:

BKSTART.

#### 7.10.9 COPYMF

This program dumps master files onto a tape or disk. Several master files may be designated to be dumped at one time. All master files that are to be dumped must first be attached in read mode.

This program does the following:

- Attaches the communications buffer to determine if any of the requested master files have been dumped before. If any have been dumped, an error message is issued to the dayfile and the job aborts unless the NA parameter has been specified. If an undumped master file is found in the parameter list, the tape or disk is requested.
- After the dump medium is assigned, the master file is dumped. As the file is being copied to the dump medium, information is being extracted as the data passes through central memory. Information regarding file names, account names, file types and file sizes is written to a temporary file.
- After all master files have been dumped, the temporary file is copied onto the communications buffer to prevent dumping this file again, and to give necessary information to the backup merge programs, BACKJNE and BACKTWD.



The control statement formats are:

CDPYMF(MF=mf,V=vsn,D=den,NA)

CJPYMF(MF=mf,PN=pn,NA)

where:

- mf = name of master file to be dumped  
This parameter may be repeated to dump several master files. You normally don't want to include more master files than will fit on a single tape or disk pack.
- vsn = VSN of tape to dump onto  
If "V" is included without a value, the operator may assign any labeled tape rather than one with a specific VSN. If neither "V" or "PN" is selected, a labeled scratch tape is needed.
- den = tape density  
If omitted, or "D" is included without a value, the system default density is used. Valid values are HD,PE,GE.
- pn = name of disk pack to dump to
- NA = specified if you do not want the job to abort if a master file has already been dumped

#### 7.10.10 CJPYMF

This program has two modes of operation: "single PLATO file recovery mode" and "master file recovery mode". The mode of operation depends on the "FF" parameter on the control statement. If "FF" is present, then single file mode is selected. If not present, then master file mode is selected.

In single file mode:

- all parameters must be specified on the control statement
- PLATO must be up
- a PLATO file must have been created to hold the recovered file ("TF" parameter) of type "backup" residing in the same account as the original file and of the same size

The CJPYMF program will request the tape/disk via the "L,P" display. The tape/disk will be scanned for the requested file, and the file will be restored if all conditions for account, file type and file size are met.

## In master file mode:

- parameters may be passed on the control statement or via the K display (the operator may enter "KK." for a complete listing of valid parameters)
- all master files which are to be recovered must be attached to the control point prior to recovery

When the proper master files have been selected, a "K.GD." will start the process. The program will request tapes/disks via the "E,P." display. The operator continues to mount tapes/packs as needed until all requested master files have been recovered.

The control statement formats are:

```
COPYPF(FF=ff,TF=tf,V=vs)
```

```
COPYPF(MF=mf,PN=pn,DA=ddd,SL=s)
```

where:

- ff = name of the dumped file to be recovered
- tf = name of existing PLATO file to be copied to upon recovery. The default is the name specified in the "FF" parameter.
- vs = VSN of the tape containing the file
- mf = name of master file to be recovered
- pn = NOS pack name containing master file to be recovered
- ddd = month and day to be recovered from
- s = slot number to be recovered from

## 7.10.11 MFDX

This is a system-unique procedure used to dump master files.

The procedure is a series of COPYMF statements used to dump each master file. A typical procedure would look similar to the one below:

```
.proc,mfdx.
.*
.*      dump plato master files
.*
copymf,v,n,mf=system,mf=ast.
copymf,v,n,mf=baast,mf=caast.
copymf,v,n,mf=daast,mf=eaast.
.
.
```

revert.

The control statement format is:

MFDX.

#### 7.10.12 RECOVAL

This procedure is used to recover all master files on a specific pack or all master files on the entire system.

The control statement format is:

RECOVAL.

#### 7.10.13 RECDVMF

This procedure is used to recover a single master file from the backups database. It also permits more flexibility for sites which do not keep their PLATO master files on removable drives, and permits the renaming of a master file during recovery. In addition, the density of the dump tape may be specified. There is a limitation that the master file can only be recovered onto the NDS pack on which it originally existed.

See the description of COPYPF for an explanation of the K-display parameters which may be used with this procedure.

The control statement format is:

RECDVMF(MF=mf,PN=pn,R=r,D=d,N=n)

where:

- mf = name of master file to be recovered
- pn = name of NDS pack on which the master file resides and to which it will be recovered  
The default is the system pack.
- r = NDS device type of "pn"  
The default is "0" (not the system default device type).
- d = density of the dump tape  
The default is PE - 1500 dpi.
- n = name of the newly recovered master file (optional)  
The default is the original master file name.

revert.

The control statement format is:

MFDX.

#### 7.10.12 RECOVAL

This procedure is used to recover all master files on a specific pack or all master files on the entire system.

The control statement format is:

RECOVAL.

#### 7.10.13 RECOVMF

This procedure is used to recover a single master file from the backups database. It also permits more flexibility for sites which do not keep their PLATO master files on removable drives, and permits the renaming of a master file during recovery. In addition, the density of the dump tape may be specified. There is a limitation that the master file can only be recovered onto the NDS pack on which it originally existed.

See the description of COPYPF for an explanation of the K-display parameters which may be used with this procedure.

The control statement format is:

RECOVMF(MF=mf,PN=pn,R=r,D=d,N=n)

where:

- mf = name of master file to be recovered
- pn = name of NDS pack on which the master file resides and to which it will be recovered  
The default is the system pack.
- r = NDS device type of "pn"  
The default is "C" (not the system default device type).
- d = density of the dump tape  
The default is PE - 1600 dpi.
- n = name of the newly recovered master file (optional)  
The default is the original master file name.

## 8 File Conversions

As software changes, new commands are added to the PLATO Author Language, and old commands may have their format altered or may be deleted from the language. Also, new features may require that the internal structure of files be changed.

To aid users in converting their lessons to run with new versions of the software, conversion programs are usually written which will convert command format and file format from that required on the old software to that required on the new software. These conversions are normally run by Services personnel as part of a software installation. Regular users should not have to worry about the conversions but will have to learn to use the new command formats.

Each file has a word in the directory reserved for its update level. This is an indicator of which conversions have been run on the file. Conversion programs normally will only run on files of the correct type and update level. Once converted, the conversion program will increase the update level by one. This marks the file as converted and prevents the conversion program from being able to convert the same file a second time.

For example, assume that a particular conversion converts all -jumpout- commands to a new format. Also, assume that the old format files are at update level 5. When this conversion is run, it will scan all files looking for files at update level 5. Files with an update level less than 5 or greater than 5 will be skipped.

Files with an update level exactly at 5 will be converted to use a new form of -jumpout-. As a final step for each converted file, the update level is increased from 5 to 6.

Unconverted files may continue to work even after a release is installed. However, eventually the software necessary to support the old format will disappear and these old format files will stop working. Therefore, it is crucial that all files be converted as soon as possible.

### 8.1 General Procedures

Each conversion program operates on different commands or possibly different file types. Since there are often special procedures which must be done for some conversions, there is no fixed procedure to follow for all conversions. There are, however, some general procedures applicable to most conversion programs.

- a. Load the PLATO system using the "PLAINS." command.
- b. Clear testlist "list" of file "convert11".

- c. Choose the conversion program.
- d. Set parameters for the conversion.
- e. Select the proper conversion method.
- f. Check log file for errors, correct problems and convert skipped files.

These procedures are described in more detail in the following sections.

#### 6.1.1 Load PLATO with "PLAINS."

Conversions often take a long time to run. Also, while they are running, the files being converted may not work correctly until the conversion is complete. If you are converting all files on the system, it is wise to keep all non-system users off the system until the conversion is complete. If you are converting a few user files (e.g., files from backups which missed the conversion), this is not necessary. Non-system users can be kept from signing into the PLATO system by loading it with the "PLAINS." DSD-command instead of the normal "PLATO." DSD-command. Only users in groups "p" and "convertc" will still be allowed to sign in. Group "convertc" may be used to sign in "multiple" records which are routed directly to the desired conversion program.

#### 6.1.2 Clear Leslist

If errors are detected during the conversion, the errors will be logged in a student data file. Also, files which could not be converted, but may be convertible if some changes are made, will be added to a leslist. After the conversion, you may make necessary changes and rerun the conversion only on the files listed in the leslist. A typical problem of this type is when the file must be lengthened before the conversion can be completed. Any file which is skipped will remain unchanged.

You may be asked for the name of the file containing the leslist and of the log file. Enter "convertll" as the name of the leslist file. File "convertll" contains a single leslist named "llst". When asked for the log file name, press NEXT and "convertlog" will be used. These files were included as part of the original release of the PLATO system.

If the leslist already contains file names from previous conversion runs, clear the leslist by hand before beginning the conversion. Do this by editing the leslist and pressing SHIFT-HELP to clear. There will be an option in the convert lesson to clear the log, so you do not have to edit the log itself before beginning the conversion.

#### CAUTION

If you are using several terminals

to convert at the same time, only clear the list or error log once when starting the first terminal.

### 8.1.3 Choosing the Conversion

Conversion programs may be executed by groups "s", "p", "convertc".

You may use author records in group "p" and choose the proper conversion program from the Author Mode display or you may create "multiple" records in group "convertc" and set the lesson for the record to the proper conversion program in the "Routing Options" for the record.

With each release, you will be told which conversions are necessary to update files to be acceptable with the new release software.

When converting files from backups or files which missed previous conversions, it is possible that the file is quite old and has missed several conversions. Refer to a later subsection to determine which conversions are necessary based on current update level, age and type of the file.

### 8.1.4 Setting Parameters

Upon entering the conversion, the program normally has a general description of what the conversion does. From that point you may either start running the conversion or choose special options, if any exist.

Before starting the conversion program, you may be given a list of controlling parameters. These parameters control the resources used by the conversion program and determine whether or not this is just a test run or the real thing. Unless directed otherwise, you should choose conversion parameters as follows:

1. High ECS, High Speed
2. Foreground Mode
3. Do Disk Writes
4. Debug Displays OFF

You may also be asked for the name of the conversion log file. At this arrow, press NEXT and "convertlog" will be used. You may also be given the option to clear the log file before beginning the conversion. This should be done only when starting the first conversion program, if you are using more than one user record to run the program.

### 8.1.5 Conversion Methods

The conversion method you choose will depend on the conditions of the conversion.

#### 1. CONVERTING ALL FILES

If the conversion is a fast conversion, use a single terminal and run the conversion on "required packs". This option will convert all master files in the required packs table seen in lesson "ipedit".

If the conversion takes a long time to run (the more frequent case), run the conversion using many terminals with a "multiple" signon. Each terminal should be assigned a separate master file from the required list. Bring up as many terminals as resources permit, but try to avoid disk conflicts by assigning each terminal a master file which is on a different disk pack.

## 2. PARTIAL CONVERSIONS

When converting a small set of files because they missed previous conversions, use the most appropriate option. If they are all on a single master file, use that option. If they are all in a list, use the list option. You may also use the single file option and type in the name of each file to be converted.

If the system crashes during a conversion or the conversion must stop, there should be no problem. Restart the conversion at a later time. If using the "whole pack" conversion method, use the "partial pack" method to restart where you left off (assuming that you know where the terminal was when it stopped). Restarting the conversion at the beginning of the master file will not take much longer because the program will skip over any converted files.

### 8.1.6 Fix Problems

Once all files are converted, either print the log or scan it with your terminal. Make any changes required. Rerun the conversion on files in the list or files which were changed. Hand convert any files if so instructed.

## 8.2 Execution Errors

By the time a remote system uses a conversion, the conversion program has normally been well tested on thousands of files. Thus, if an execution error occurs in a conversion program, it usually means there is an error in the file being converted, not the conversion program. If you get an execution error, do the following:

- a. Choose the "partial pack" conversion method for the master file which was being converted at the time of the error.
- b. When asked for the starting file, enter the name of the file being converted when you got the execution error.



- c. Choose to continue the conversion at the following file. This will skip the file which causes the execution error.
- d. Obtain a backup of the bad file and convert it using the single file convert option.
- e. If the problem recurs, notify PLATO Support so that they may determine the cause of the execution error. See the "Problem Reporting" section for information on how to report this problem.

### 8.3 Error Messages

Error messages which may be seen while running a conversion program are listed below. Actions to be taken to correct these problems are also given.

#### ATTACH ERROR

The file to be converted was attached by another user. The file name is added to a leslist so the conversion may be rerun on the file.

**ACTION:** After the initial run of the conversion program is complete, rerun the conversion on all files listed in the leslist. If the file remains attached even though no one is using it, you may use an option in lesson "operator" to detach it, then rerun the conversion on the file.

#### BLOCK LENGTH ERROR

There is an error in the block directory. The space used in the block plus space left is greater than the block size.

**ACTION:** Obtain a backup of the file and then convert.

#### -CHANGE- COMMAND #fname/filename#

This means a -change- command was in use in file "filename" on master file "fname". The file was converted anyway, but it is possible a changed command was not found and not converted.

**ACTION:** Notify the author of such a file.

#### DISK ERROR (error #), BLOCK (block #)

A disk error has occurred.

**ACTION:** Check NDS error log, contact a DE if necessary, and restart the conversion when ready.

#### ECS PACK DIRECTORY MOVED

This should not happen unless account operations are being done during the conversions.

**ACTION:** This should recover itself, but to be sure

rerun the conversion on the master file involved.

**LESSON MUST BE EXTENDED**

There was not enough room in the file to add all of the necessary code resulting from the conversion. Usually this only happens to files which are using every available block in the file. The name of the file is added to the leslist of files with errors.

**ACTION:** Lengthen the file using "operator" or "accounts" and rerun the conversion on the file. If the file cannot be lengthened, the author must be contacted to make more space available in the file.

**LESSON NOT FOUND**

This should only occur when using the "leslist method" of conversion. A lesson in the leslist does not exist, and was not converted.

**ACTION:** Correct the name of the file or ignore.

**LINE OVERFLOW**

A converted source line is longer than the legal limit.

**ACTION:** Convert the line by hand and then convert this file using the "single file" method.

**NESTED USE**

The conversion program has detected a -use- command which will not work. The lesson will be converted anyway.

**ACTION:** If this is a published lesson, report via PSR.

**OBSOLETE LESSON**

This file cannot be converted because it has not been run through previous required conversions.

**ACTION:** Run the necessary conversions from previous releases and rerun this conversion on the file.

**OLD FORMAT FILE**

This file probably has a bad file directory.

**ACTION:** Load the file from backups or destroy it.

**OVERSIZED FILE**

The file length is too large to convert.

**ACTION:** Shorten the file, if possible, and then convert. If the file cannot be shortened, you may have to contact the lesson author to make changes to allow shortening the file.

**-USE-D BLOCK MISSING**

The conversion program has detected a -use- command which will not work. The lesson will be converted anyway.

**ACTION:** If this is a published lesson, notify Published Courseware Delivery.

#### 8.4 Conversion Programs

- convert29** rename commands: -dataset- with tag to -attach-, blank-tag -dataset- to -detach-, -touch- to -ntouch-, -innibit dropset- to -innibit dropfile-, -reserve dataset- to -reserve file-, -reserve dataset,x,y- to -reserve records,x,y-, -group- to -keylist-, -course- to -group-.  
 rename reserved words: "zdsrecs" to "zrecs", "zdsapr" to "zapr", "zdsname" to "zfile", "zdsusers" to "zfusers".  
 eliminate -branch q-, replace semi-colons in -keytype- with commas, -enable-/-disable- to -enable touch-/-disable touch-. changes update level of TUTOR files from 0 to 1.
- convert31** clears old catalog info words in TUTOR file directories, move common codeword into attach codeword (TUTOR files) and move common/inspect codes into records write/read codes (dataset/nameset files). changes update level of TUTOR files to 2, dataset and nameset files to 1.
- convert33** corrects error in lineset blocks. no changes to update levels.
- convert42** -ntouch(w)- to -touch(w)-, -color- to -mode-. changes update level for TUTOR files to 3.
- convert46** initialize unique serial ID numbers. changes update level for group files to 2.
- convert47** CMI to PLM conversion. no changes to update levels.
- convert48** convert general note files to use access blocks. changes update level for gnotes files to 1.
- convert49** clear NDS family name word in records. no changes in update levels.
- convert50** account file directory given "TUTOR" file format (allows variable length accounts for archiving lists). changes update level for account files to 1.
- convert51** PLM (GED version) conversion. no changes to update levels.
- convert52** fix -jumpout- commands with bad format which is no longer allowed since release

of ~~-jumpout-~~ commands with arguments,  
no change in update levels.

convert54 fix problems in notes files left by a  
previous conversion. changes update level  
for gnotes files to 2.

convert55: zero data fields in group directories  
and user records which are to be used  
with TERM-ask feature. no changes to  
update levels.

convert57 put commas on conditional ~~-do-s~~ which were  
using end-of-line as separator between the  
condition and the first unit, removes semi-  
colons in unit arguments (semi-colons now  
signal start of return arguments). changes  
update level for TUTOR files to 4.

convert58 clears TERM-ask user flag for all records.  
changes update level for group files to 4.

convert59 converts all PLM curriculum files and  
modules to specific PLM file types. sets  
update levels for PLM curriculum files to  
2 and for PLM module files to 1.

#### 8.4.1 Conversion Programs Continued

convert60 ~~-return-~~ to ~~-break-~~, ~~-nrat-~~ to ~~-return-~~.  
changes update level of TUTOR files to 5.

convert61 PLM V5 conversion - changes format of PLM  
module files. changes update level of PLM  
module files to 2.

convert63 "adaptable records" conversion - zeroes  
the "application lesson" field in group  
file directories, adds author/instructor  
options. changes update level of group  
files to 5.

convert64 clears fields in all file types to be  
reserved for a future development project.  
no changes in update levels.

convert65 reformats account files to remove the  
obsolete lesson usage data to allow accounts  
to have up to 5000 files. changes update  
level of account files to 2.

convert66 copies processor lesson for group files  
into editor lesson slot in directory,  
sets fields in each record for use in  
new security features, sets default fore-  
ground and background color fields in all

records. changes update level of group files to 6.

## 8.5 Conversions by release

This section lists the names of each conversion run as part of each release. For details about each conversion, refer to the section entitled "Conversion Programs."

Releases for which there were no conversion programs are not listed.

### 8.5.13 Release 13 -- March, 1978

convert28 TUTJR files (file type "a")

### 8.5.15 Release 15 -- August, 1978

convert31 TUTJR files (file type "a")

convert33 TUTJR files (file type "a")

### 8.5.18 Release 18 -- May, 1979

convert42 TUTJR files (file type "a")

convert46 group files (file type "f")

convert47 CM1 to PLM conversion

### 8.5.19 Release 19 -- November, 1979

convert48 general notes files (file type "i")

convert49 group files (file type "f")

convert52 TUTJR files (file type "a")

### 8.5.20 Release 20 -- March, 1980

convert50 account files (file type "i")

convert51 PLM files

convert54 general notes files (file type "i")

convert55 group files (file type "f")

### 8.5.21 Release 21 -- June, 1980

convert57 TUTJR files (file type "a")

convert58 group files (file type "f")

### 8.5.22 Release 22 -- October, 1980

convert59 PLM curriculum (file type "n")

PLM module (file type "o")

### 8.5.23 Release 23 -- February, 1981

convert60 TUTJR files (file type "a")

### 8.5.26 Release 26 -- February, 1982

convert64 all file types

8.5.27 Release 27 -- June, 1982

convert63 group files (file type "f")  
convert65 account files (file type "l")

8.5.31 Release 31 -- October, 1983

convert66 group files (file type "f")

## 9 PLATO Inter-System Link

### INTRODUCTION

The PLATO Inter-System Link is based on Control Data's Remote Host Facility Access Method (RHFAM) product and the Permanent File Transfer (PTF)/Queue File Transfer (QTF) File Transfer Facilities product. The RHFAM and PTF/QTF products must be installed in order for the PLATO Inter-system Link to work.

The PLATO Inter-System Link provides three basic features:

- Inter-system Personal Notes

PLATO Personal Notes can be exchanged by users on separate linked PLATO systems.

- Inter-system Group Notes

PLATO Group Notes files may be linked to one or more other PLATO Group Notes files. These files may be set to send and/or receive from each other, allowing either one-way or two-way communication.

- Inter-system File Transfers

Users may direct a PLATO file to be sent to or from another system. Users must be validated by both the sending and receiving PLATO accounts. The accounts must be authorized for networking features by the respective site's system staff.

The link software also supports the following features:

- Store and Forward Mode

Each system stores and forwards notes and files destined for other systems in their network. The link obtains the request records of these queued files, determines the true destination, and forwards them to the next system in the network. This process continues until the files reach their respective destination systems.

- Network Protocols

There are a number of different network protocols which will be supported by the PLATO Inter-system Link. This includes direct data lines between 2550's, remote trunks between 2550's, host-to-host connections between couplers in a shared 2550, and the X.25 protocol, using any suitable Public Data Network (PDN), such as UNINET, TYMNET, TELENET, etc.

### 9.1 Inter-System Link Operation

Outside of the normal operation of your PLATO application, the only operational responsibility for the PLATO Inter-system Link is that of noting and resolving network problems with the NAM and RHP applications, which could cause the link to fall. With this in mind, a brief overview of the basic flow of the link software is in order.

A request is received by the link from: 1) a user wishing to send a file or 2) a runner program distributing personal and group notes to systems connected to your network. A batch job is submitted which will use the NOS MFLINK command to send the data through NAM. After having done this, the link checks for errors. If any are found, it will issue an informative message to its error log. At this point, the operator or system controller must resolve the problem through the following procedure.

1. Use the A-display of the computer console to scan the system dayfile for the job submitted by the link. If the job cannot be found, try the request again and watch for the job to be submitted.
2. Check for any error messages following the MFLINK command. Refer to Appendix B of the NOS Version 2 Reference Set Volume 3 (System Commands) for MFLINK error messages and their meanings.
3. If a MFLINK error message can not be found, you should status the data line through NAM to see if it is at fault. Refer to the NOS Version 2 Operations Handbook for more instructions regarding NAM.



## 10 Usage Tracking

Application usage statistics are gathered to:

- a. track the growth of the application
- b. charge customers based on usage
- c. determine payment of royalties to authors

### 10.1 How the Tracking is Done

#### 10.1.1 NDS Account File

Lesson usage tracking is done through the NDS account file.

The following entries are written in the active NDS account file by the PLATO application:

##### 10.1.1.1 PS entry

PS - PLATO Start entry:

Part of the initialization process for the PLATO application is to write into the NDS account file a message indicating the date and time that the application was brought up. The format of this entry is:

```
1111111111222222222233333333334444444444
1234567890123456789012345678901234567890123456789
```

```
hh.mm.ss. PLAT5. PS yy/mm/dd.xxxxxxx
```

where:

```
hh.mm.ss = time of start-up
yy/mm/dd = date of start-up
xxxxxxx = system name ("sid" configuration parameter)
```

##### 10.1.1.2 PA entry

PA - PLATO Account entry:

During initialization, the PLATO application writes the names and ordinals of all active PLATO accounts into the NDS account file. The format of this entry is:

```
11111111112222222222333333
12345678901234567890123456789012345
```

```
hh.mm.ss. PLAT5. PA zzzzzzzz000
```

where:

```
hh.mm.ss = time of initialization
```





hh.mm.ss = time of exit from lesson  
 pppp = PLATO port number (32 \* site + station)  
 aaaa = lesson account ordinal  
 llllllllll = lesson name  
 tttt = number of minutes in lesson  
 ccccc = absolute (unscaled) TIPS \* 4  
 bbbbbbbbbb = lesson access class bits (Press LAB while editing an account for the matching between bits and published courseware libraries.)

The relation between the \*cccc\* field and TIPS (Thousands of Instructions Per Second), as the user sees them, is as follows:

$$\text{user TIPS} = (\text{cccc} / 4) * (\text{cpspd} / 1000)$$

where:

ccccc = same as above, unscaled TIPS \* 4  
 cpspd = value of the "cpspd" configuration file entry

#### 10.1.1.8 PM entry

PM - PLATO Module usage entry:

Each time a user exits from a PLATO Learning Management (PLM) module, in which the user spent more than 30 seconds, the time, port number, module account ordinal, module name, and contact time are written to the NJS account file in the following format:

```

11111111112222222222333333333344444444
12345678901234567890123456789012345678901234567

```

hh.mm.ss. PLALS. PMppppaaaaaaaaaaaaaaattttt

where:

hh.mm.ss = time of exit from module  
 pppp = PLATO port number (32 \* site + station)  
 aaaa = module account ordinal  
 aaaaaaaaaa = module name  
 tttt = number of minutes in module

#### 10.1.1.9 PD entry

PLATO user log-off entry:

Each time a user signs (logs) off the PLATO application, the time and port number are written in the NJS account file in the following format:

```

11111111112222222222
1234567890123456789012345678

```



aaaaaaaaa = current allotment for this user's logical site  
 ("tmallot")  
 ccccccccc = total EM in use at user's logical site  
 ("tause")  
 ttttttttt = total EM available in the system  
 rrrrrrrrr = address of the calling subroutine

#### 10.1.2 DFTERM

The current NDS account log is terminated using the DFTERM NDS command. Normally, this is done once a day as part of the dumps. It MUST be done while the PLATO application is DOWN.

When the NDS account log is terminated, its contents are placed in a file named "ACAmmdd" (where "mm" = the current month and "dd" = the current day) under user name SYSTEMX (user index 377777b). If the account file is terminated more than once on the same day, the successive files will be named "ACdmmdd", "ACCmmdd", etc.

#### 10.1.3 The Billing Cycle

The billing cycle describes a period of time for which information is to be kept and stored as a single logical entity for billing purposes. File RAFMDN under user name SYSTEMX (user index 377777b) is used to store all NDS account file information for the current billing cycle. This file is also called the "raw account file".

#### 10.1.4 Daily Processing

On a daily basis, process all NDS account files as follows:

- a. Execute lesson "opjobs".
- b. Choose option to "List/process existing files".
- c. Enter your NDS password, if requested.
- d. When the file list appears, press NEXT to process the files.

#### 10.1.5 End of the Billing Cycle

At the end of each billing cycle (usually the end of the month), all complete billing cycle information must be copied to tape. This option will automatically appear during the daily processing in "opjobs", once all of the files for the current billing cycle have been processed.

Running this job will require that an operator take action at the computer console. Two tapes will be requested so that two copies of the billing cycle can be made.

One of the tapes should be sent to the address below for final processing.

Control Data Corporation  
 PLATO Operations - MEVCIB  
 ATTN: Ken Daniel  
 511 11th Ave. So.  
 Minneapolis, Mn. 55415-1579

The other tape should be saved until it is known that the first tape has safely arrived and been processed.

The option to set the date of the end of the billing cycle must be used to set the correct date when the end of the billing cycle processing should be done. Once it is set, no individual account files beyond this date can be processed until the end of the billing cycle processing. Using this option should prevent mistakes.

## 10.2 Account Summary Processing

On some systems (specifically Control Data Service systems), the account summary data generated in accounts is saved for use by order processing people.

This account summary data is generated as follows.

- a. Use the system options in "accounts" to generate the account summary. See section "Account Summary Data Format" for a description of the format of the data produced.
- b. Go to lesson "opJobs" and choose the option to "save the account summary on a NDS file". See section "NDS Account Summary Data Format" for a description of the format of the data produced.
- c. At the end of a billing cycle, the operator will be asked to mount a tape on which to dump the NDS file. A job will be submitted to copy the NDS file to the tape and to clear the file for a new billing cycle.

### 10.2.1 Account Summary Data Format

Dataset "z1acnt" holds the summary information concerning accounts on a given system. It is generated by the "account summary" option found under the system options of lesson "accounts".

The first 320 words of the file contain the standard dataset directory information. The dataset is composed of 320-word physical records, which contain two kinds of logical records, one "header" record containing the system-wide totals for all accounts, followed by a record for each account.

The header record format: (20 words)

WORD POINTS-----

1 number of accounts  
 2 number of subscriptions  
 3 number of files  
 4 number of file spaces allocated  
 5 number of file spaces in use  
 6 date of last summary update  
 7 time of last summary update  
 8 signon name of last person to run the summary update  
 10 group name of last person to run the summary update  
 11 system name  
 12 number of accounts with network features active  
 13 number of archive rights  
 14 number of publishing accounts  
 15 (unused)  
 16 (unused)  
 17 (unused)  
 18 (unused)  
 19 (unused)  
 20 (unused)

The individual account record format: (55 words)

WORD CONTENTS-----  
 1 account name  
 2 account number  
 3 signon name of account owner  
 5 group name of account owner  
 6 number of subscriptions  
 7 number of files in the account  
 8 number of file spaces allocated  
 (contains integer -1 if no limit on file space)  
 9 number of file spaces in use  
 10 publishing account flag (0 = false, 1 = true)  
 11 number of archive rights  
 (if account uses "deferred billing", this word contains  
 the number of file spaces actually archived - see the  
 "billing mode" flag in word 29)  
 12 network features flag (0 = false, 1 = true)  
 13 (unused)  
 14 (unused)  
 15 (unused)  
 16 (unused)  
 17 author creation allowed in groups (0 = false, 1 = true)  
 18 system name of last director to use options  
 19 account name for redundant error check  
 20 date of last account option use  
 21 time of last account option use  
 22 first 10 characters of signon name of last editor of  
 account  
 23 48 bits - last 3 characters of signon name of last  
 editor of account  
 12 bits - station number in use by last editor of  
 account  
 24 48 bits - group of last editor  
 12 bits - last action type performed by last editor  
 25 name of file edited (if appropriate for last action)



- 26 name of secondary file edited (if appropriate)
- 27 date account was created
- 28 lesson access bits for account
- 29 archive billing mode  
(0 = deferred billing - word 11 contains the actual number of archived spaces, 1 = archive rights - word 11 contains a pre-allocated, consumable number of service units.)
- 30 network library control flag (CDC service systems only)  
(0 = not a network library account,  
-1 = remote accounts being set up,  
'cpl' = full-fledged network library account)
- 31 network library type (0 = remote, -1 = master)
- 32 network library -- spaces in use
- 33 number of uncharged spaces allocated to this account
- 34 TRANSMIT flag (1 = account allowed to use TRANSMIT)

The remainder of the record, through word 55, is unused.

### 10.2.2 NDS Account Data Summary Format

At the end of the billing cycle, the operator uses an option to "save the account summary on a NDS file" in lesson "opjobs" which calls lesson "z1report" to format the account summary data into file "z1data". The data is then transferred from "z1data" to a NDS file to be included on the end-of-billing cycle accounting tape. This data may be used as input to CDBJL or FORTRAN programs which might otherwise be unable to interpret the "bits-and-bytes" format of the "z1act" file.

This file should be kept as short as possible to avoid wasting space during collection and processing during the billing procedures. Its size is a function of the number of accounts on a system. Two or three parts will be adequate for most systems.

The format of "z1data" is "lines" of 160 characters, the last two chars being the end-of-line code (zero). The first line contains information pertaining to the summary itself. Its format is:

Summary of xxxxxxx system on mm/dd/yy at nn.mm.ss.

where xxxxxxx will be replaced by the name of the system (blank-filled) and the other two variables are the date and time when the account summary was generated.

The remainder of the file is composed of one line for each account on the system in the following format:

Description	Chars	Notes
account name	7	
account number	4	
creation date	8	(mm/dd/yy)
owner's name	18	
owner's group	8	

no. of subscriptions	6	
no. of files	6	
no. of spaces allotted	6	
no. of spaces used	6	
no. of archives	6	(see archive billing flag)
(unused)	19	
publishing acct flag	1	("y" or "n")
network options flag	1	("y" or "n")
author creation flag	1	("y" or "n")
lesson access bits	30	(each bit: "y" or "n")
archive billing flag	1	if "a", account has paid in advance for a number of service units and the number of remaining units is in the "no. of archives" field. if "c", account is charged for service units as they are used and the "no. of archives" field contains the number of spaces in use by account's archived files.
network library flag	1	(CDC service systems only) if "n", no network library. if "m", account contains "master network library". if "r", account contains "remote network library".
network library spaces	6	
TRANSMIT flag	1	if "c", account is allowed to use TRANSMIT. if "n", account is not allowed to use it.

The data is followed by another line of the same length which reads:

end file

The remaining space in the file is unused, but may contain data from previous runs.

### 10.3 Reports and Programs Available

#### 10.3.1 Availability Report - RAFFUD

This report shows the following information for the period of time covered by the NDS account file used as input:

1. The time during which the application was available.
2. The times of crashes and stations in use at crash time (where a crash is defined as any situation where the application was not brought to a controlled stop by using a system wide backout followed by a "K.STOP").
3. Usage by user type, maximum users, and hours of lesson usage for each day.

4. Summary data over the entire billing cycle, showing total terminal connect time.
5. The names and lessons of people backed out during EM shortages.

Data can be verified to see whether fields which are supposed to be numerical do, in fact, hold numerical data. Turning on sense switch 1 (via the "ONSW(1)" control card) enables this checking. Note that there will be an increase in the processing time when this option is selected.

The control statement format is:

```
RAFPDD.
```

The raw account file must be attached as a local file named RAF to be used as input.

#### 10.3.1.1 Sample Program

The following generates an availability report for the most recently completed billing cycle.

```
su1,377777.
attach,raf=orafmon.
rafpdd.
```

#### 10.3.1.2 Error Exits

The following is a summary of the fatal and non-fatal error exits which may be encountered.

##### Non-fatal:

With all these non-fatal errors, the record being processed is ignored and processing continues. However, this might lead to the generation of further non-fatal errors which might have been dependent on the first bad record. Either delete the bad record, or ignore the non-fatal errors.

##### ACC ALREADY DEFINED

A "PA" entry was found which duplicates the account ordinal of a previous "PA" entry.

##### ACC ORDINAL OVERFLOW

An account ordinal was found to be greater than the maximum number of accounts.

##### ACCOUNT INDEX ERRJK

A non-numeric account ordinal was found.

##### ACCOUNT NO. ERROR

An unknown account ordinal has been encountered (no "PA" entry for the ordinal). The account name is set to "unknown".

**DEAD START**

Informative message used to show when system deadstarts have been done.

**PORT IN USE ERROR**

A user was trying to log into a port which was already in use.

**PORT IN USE WARNING**

This error usually occurs when a system person has reconcondensed lesson "plato".

**PORT .GT. 2048**

A port number greater than 2048 was found.

**PORT NOT LOGGED IN**

Activity has been recorded for a port which has not been logged in (no corresponding "PI" entry for port).

**PURGE ACC NONEXISTENT**

A "PR" entry was found for an account for which there was no corresponding "PA" entry. Either the account ordinal or the account name may be wrong.

**RECORD IGNORED**

A field held unrecognizable data.

**UNKNOWN USER TYPE**

A "PI" entry was found with an illegal character in the "user type" field (not "A", "B", "C" or "D").

**//USER DELETED LSN//**

A user was deleted from a lesson while it was in use.

**Fatal:****PS ENTRY NOT FIRST**

A "PS" entry was not the first PLATO-related account file entry found. This usually happens when the NGS account file is terminated while the PLATO application is still active. (See an earlier discussion about DFTERM in this section.)

To resolve this problem, you must edit the file being processed so that all PLATO-related entries before the first "PS" entry are deleted. Another way to resolve this problem would be to obtain a portion of the account file from the LAST time it was terminated, starting at the first "PS" entry, and insert that data at the beginning of the current file.

**10.3.2 Port Usage Report - PORTX**

This report lists the total connect time for each port (site-station) which was used during the billing cycle, followed by the total connect time for the entire system.



The control statement format is:

RAFPBC.

File RAF is the local file used as input to RAFPBC. It is a NDS account file. Local file RPAD is used to store the written data to be later used as input to ASMI.

#### 10.3.3.2 Sorting the file - ASMI

This program is used to generate a sorted billing cycle file from the reduced account file produced by RAFPBC.

The control statement format is:

ASMI.

File RPAD is the local file used as input to ASMI. Output from ASMI is to file BC, a local file where the sorted billing cycle data is written. This is later used by other report generator programs, such as UURDC.

#### 10.3.3.3 Sample Program

The following generates a sorted billing cycle file for the most recently completed billing cycle, the data from which is assumed to be previously defined in the file called DRAFMON.

```

sul,377777.
attach,raf=orafmon.
define,bc=bc79jul.      name can be any meaningful name
rafpbc.
asmi.

```

The sorted billing cycle file is now on a permanent file named "bc79jul" which may be attached when running reports requiring a sorted billing cycle file.

Data can be verified to see whether fields which are supposed to be numerical do, in fact, hold numerical data. Turning on sense switch 1 (via the "DNSW(1)" control card) enables this checking. Note that there will be an increase in the processing time when this option is selected.

#### 10.3.3.4 Error Exits

The fatal and non-fatal error exits which may be encountered for this job are the same as those which may be seen when generating an availability report.

#### 10.3.4 User Usage Report - UURDC

The User Usage Report is designed to report on what various users were doing during the billing cycle. The level of detail in the report may be any one of the following:

1. contact hours for each account
2. contact hours for each group within an account
3. contact hours for each user within each group within an account
4. contact hours for each lesson executed by each user within each group within an account (all lessons in account "system" are grouped together without a lesson-by-lesson breakdown).

Furthermore, data may be generated for either a list of accounts or for all accounts.

The control statement format is:

UJRBC.

The sorted billing cycle file must be attached as a local file named BC to be used as input.

The user usage report program requires a file of parameters to be on NDS file TAP21 when it begins. The following is the format of these parameters.

- The first three (3) lines are reserved for comments, and are AUTOMATICALLY ignored.
- The fourth line holds a 10-character (alphanumeric) date. Be sure to include a leading blank for display purposes.
- The fifth line is used to specify the level of detail desired in the report. Permitted values are:

```

account
group
user
lesson

```

- The sixth line is used to specify the scope of reporting. Either specify the string "all" to report on all accounts, or specify a list of up to 10 accounts to report on lines 6-15 (one account per line, left justified. For example:

```

* line 4 date (yy/mm/dd)
* line 5 kind (account, course, user, lesson)
* line 6 to line n (account names)
84 June 2+
account
ps
system

```

Note that the example above uses a list of accounts. The

other alternative would be to follow the "account" entry by the string "..all", which would report on all accounts.

#### 10.3.4.1 Sample Program

The following generates a user usage report. In this example, it is assumed that the billing cycle file to be processed is a NOS permanent file named "bc79jul" and that the input parameters are kept in block "userparaas" of a PLATO file named "reports". (See lesson "nosalds" for a description of the PF command.)

Sample parameters:

```
*line 4  date  m/y
*line 5  kind  (account,course,user,lesson)
*line 6  to    line n      account names
  Jun 80
  account
  ..all
```

Sample Job:

```
sub,377777.
attach,bc=bc79jul.
pf(pb,tape1,z,z)/reports,userparaas
urbc.
```

#### 10.3.5 Lesson Usage Report - LURBC

The Lesson Usage Report is designed to show who was using available lessons over the billing cycle. The levels of detail in the report may be any of the following:

1. contact hours for all lessons in a particular account
2. contact hours for each individual lesson in a particular account
3. contact hours for each individual lesson in a particular account with a further breakdown showing how much of that use was by each account on the application
4. contact hours for each individual lesson in a particular account with a further breakdown showing how much of that use was by each user of the application

Furthermore, data may be generated for either lessons in a list of accounts or for lessons in all accounts.

The control statement format is:

```
LURBC.
```



The sorted billing cycle file must be attached as a local file named BC to be used as input.

The lesson usage report program requires a file of parameters to be on NDS file TAPE1 when it begins. The following is the format of these parameters.

- The first three (3) lines are reserved for comments, and are AUTOMATICALLY ignored.
- The fourth line holds a 10-character date. Be sure to include a leading blank for display purposes.
- The fifth line is used to specify the level of detail desired in the report. Permitted values are:

```

account
lesson
uaccount
user

```

- The sixth line is used to specify the scope of reporting. Either specify the string `"..all"` to report on all accounts, or specify a list of up to 10 accounts to report on.

#### 10.3.5.1 Sample Program

The following generates a lesson usage report. In this example, it is assumed that the billing cycle file to be processed is a NDS permanent file named `"bc79jul"`, and that the input parameters are kept in block `"lessparams"` of a PLATO file named `"reports"`. (See lesson `"nosalds"` for a description of the PF command.)

Sample parameter file:

```

*line 4  date  mm/yy
*line 5  kind  (account,lesson,uaccount,user)
*line 6  to    line n      account names
  Jun 80
account
..all

```

Sample job:

```

sul,377777.
attach,bc=bc79jul.
pf(pb,tape1,z,z)/reports,lessparams
lurbc.

```

#### 10.3.6 Combined Reports & Example

Several reports may be generated as part of one long program. An example of what such a program might look like is as follows:

```

sui,377777.
attach,raf=orafmon.
*
* generate port usage report
*
portx.
*
* generate plate availability report
*
rewind,raf.
rafpdu.
*
* generate sorted billing cycle
*
rewind,raf.
define,bc=bc79jul.
rafpbc.
esml.
return,raf,ypad,ppad.
*
* generate user usage report
*
rewind,bc.
pf(pb,tapel,z,z)/reports,userparams
urbc.
*
* generate lesson usage report
*
return,tapel.
rewind,bc.
pf(pb,tapel,z,z)/reports,lessparams
lurbc.

```

### 10.3.7 DATESCN

This program is called during the end of billing cycle processing job submitted by lesson "opjobs". It compiles a list of all dates and times of system deadstarts found in the raw account file. This list is used by other programs when searching for information for a given time period. The format of the information produced is as follows:

```

billing cycle dates
01.27.23. 82/08/17 aesy
01.15.14. 82/08/18 aesy
01.15.14. 82/08/18 aesy
01.08.39. 82/08/19 aesy
01.08.39. 82/08/19 aesy
01.15.37. 82/08/20 aesy

```

The control statement format is:

```
DATESCN.
```

The raw account file must be attached as a local file named TAPE1 to be used as input. The list of dates is written to local file TAPE4.

## 11 Courseware Installation

Files shipped on tape, to be loaded onto the application, may only be loaded if approved by PLATO Courseware Delivery and PLATO Courseware Maintenance. Files shipped from some third systems may not be compatible with your application, and should not be loaded.

### 11.1 Courseware Update Installation

The following procedure is to be used for installing updates to published courseware files and to install the site-specific files delivered with the initial courseware shipment. Refer to the "Install published courseware" section of the PLATO Installation Guide for instructions on how to install the initial courseware shipment for your system.

The steps to install courseware updates are:

- a. Use procedure MFTLOAD to copy the contents of each of the master file tapes to disk. Refer to the "Operations Utilities" section for information about using MFTLOAD.
- b. You may wish to use MFPRINT to get a listing of all the files in each master file, or prints may already be included with the shipment. MFPRINT is also discussed in the "Operations Utilities" section.
- c. Execute lesson "ldr" to connect (or "load") each master file to the system. (You may want to load all the master files at once, or do them one at a time, depending on the number of free master file slots and the amount of free space in your required master files.)

NOTE: If the number of NEW disk parts is more than the amount of free space currently available in your required master files, it may be necessary to add one or more master files to your system before beginning the installation. Refer to the "Adding a Required Master File" section of the PLATO Configuration Handbook for information on how to do this.

- d. Execute lesson "transfer".
- e. Select the "Install new courseware" option. This option will copy the contents of the new master files into your required master files.

NOTE: Occasionally, during installation, the error message "cardfiles attached" may be seen. In this situation, inhibit lesson "catalogs" via lesson "sysopts", then continue with the installation. Remember to uninhibit "catalogs" after the installation is complete.

NOTE: At times, it is necessary to deliver replacement files which are not of the same PLATO file type as the files currently on a given system. For example, a dataset might be replaced by a nameset. In the process of the installation, this will show up as an error in lesson "transfer" with the message "Files not of same size/type". Should this occur, press SHIFT-NEXT to continue with the installation. Do NOT destroy the master file with the file(s) which could not be installed. After the installation of all master files, perform the steps necessary to "Destroy Obsolete Published Courseware" described in the following section. Upon completion of the destroying, use "ldr" to reconnect the master file containing the file which could not be installed if it is not still connected. Then, re-execute the steps to install the courseware in that master file. By this time, the offending file will have been deleted, so that the replacement copy can be installed without error.

- e. Once the "transfer" process is complete, execute lesson "ldr" and UNload the courseware master file(s). They can then be purged.

Now proceed with the procedures to destroy obsolete published courseware.

## 11.2 Destroy Obsolete Courseware

It is sometimes appropriate to delete files in published accounts. This activity is managed by PLATO Courseware Delivery.

At times, files are deleted because they were sent to a system in error or because a course has been revised and the files are no longer needed by the product. At other times, files are deleted because a product must be removed from the market. In this case, a warning message is added to the router file or to several lesson files in PUBLISHED products. The warning states the date on which the product will become unavailable. After this date, user access to the product is disabled. The files will remain on the system until the courseware update following the date of product unavailability.

A system-specific leslist will be sent with each update to allow Operations to delete any obsolete courseware. Once an update has been installed, the steps for deleting obsolete published courseware are as follows:

- a. Execute lesson "transfer".
- b. Select the "Destroy Obsolete System Courseware" option.
- c. Select the "Destroy obsolete courseware files" option.

d. When prompted, press SHIFT-HELP to begin the operation.

Upon completion of this last step, the installation of your published courseware update shipment is completed.

## 12 The PLATO Bulletin Board

Lesson "bullfile" is not a lesson to be executed in the normal way. Instead, it is displayed by the PLATO application when an author presses "3" on the Author Mode display. This is known as the PLATO Bulletin Board and is maintained by the operations staff.

The content of block "bull" in "bullfile" is displayed by a -text- command, so anything entered there will be displayed to authors looking at the bulletin board. The usual things which are kept in the bulletin board are:

1. A list of prime-time hours. Prime-time hours are those hours during which there is NORMALLY a system consultant present. Temporary changes in these hours (like an upcoming holiday) should not be entered into "bullfile".
2. The PLATO Hotline number (if one exists). This is a phone number which users should be able to dial to reach an operations person or system consultant.

You should leave at least 3 free lines at the top of the display in order to prevent overwrites by system messages which use these lines.

### 13 Consultant Features

Application consulting is the function of group "pso". To perform this function, group "pso" is given inspect access to most of the system operations utilities.

The following are features which are directed primarily toward the PLATO consultant.

#### 13.1 "c"

By using lesson "c", systems people can make themselves consultants. People who consult almost every day may also use "c" to set up an automatic option so that each time they sign in they will be asked if they want to be a consultant or not.

An author normally indicates he needs consulting help by using TER4-consult at his terminal. A request for help will automatically be sent to the terminal of any person currently signed in as a consultant.

Once a consultant receives a request for help, the person requesting help may be monitored through lesson "c".

#### 13.2 "psonotes"

File "psonotes" is a group notes file on all systems used for communication between authors and consultants. Normally, users in group "pso" are directors of the file and all other users have write-only access.

Consultants should read and respond to notes in this file every day.

#### 13.3 "aids"

Most of the "aids" package is available to all users and is updated with the software delivered with each release. There are a few files which are system-unique, and it is the responsibility of each site to keep this information up-to-date. With each release, instructions will be given in the PLATO Installation Guide, if these files must be changed.

These system-unique files are:

```
a0psoless
a0ssl
```

## 14 Logical Site Director Features

A "logical site" is a collection of terminals which may or may not be located together. They are related in two ways:

1. use of the terminals is controlled by the application in accordance with a set of rules for that logical site.
2. all of the users of that logical site share (compete for) the computer memory (EM) allocated for the exclusive use of that site.

Logical sites are created and EM is allocated through lesson "allocate".

### 14.1 Recommended Use

Rather than group users into very small logical sites, it has been found to be more effective to group all users into one large pool. When computer memory is so short that there is conflict within the pool, then more resources (more EM) should be obtained.

Thus, in general, the recommended use of logical sites is as follows:

site C -- site used by the general pool  
 site 03 -- site reserved for "runner" programs

The other logical sites are available for other groups, but their use is generally discouraged. However, if one group feels it is necessary or if management wishes to give one group a sort of higher EM priority when EM becomes scarce, then use of logical sites 1-62 may be necessary.

On systems which use only the ASCII (PNI) network, terminals are not permanently assigned to a particular PLATO site and station. Under these conditions, terminals cannot be assigned to many different logical sites and one logical site for all users is the only method to be used.

### 14.2 Setting Up Additional Sites

An additional logical site may be created using lesson "allocate". The site should contain ALL terminals belonging to a particular account or set of accounts. The site director should be determined by the corresponding account owner.

Since a rotary dial-up terminal may be used by several different accounts, such a terminal cannot be assigned to a logical site belonging to a particular account. Instead, such terminals must stay in the general pool.

Lesson "allocate" is used to create the new logical site



and to assign terminals and EM to it. Use lesson "site" to assign the site director.

### 14.3 Site Director Options

Normally, all site director features are used only by the system operations staff since they are the default directors of the single "pool" logical site. If additional logical sites have been created and a site director assigned to them, they may use site director features within their own logical site.

Site directors control their logical site through lesson "site". After creating the logical site, there is no further need for system controller intervention. The site directors should be able to do everything that needs to be done except for one thing. They cannot control the amount of EM allocated to their site. Only the system controllers can do that.

Since system controllers are the site directors for some of the logical sites, it is important that they understand how to use the options available to them in lesson "site".

The following options are available.

- a. Site usage display. This shows all the users on a particular logical site, what lesson they are using, the ASCII network ID assigned to a terminal when it logs onto the ASCII (PNI) network, and how much EM they are using. From this option, site directors may also send a message to, or backout a specific station on their logical site.
- b. Reserved lesson list. By entering a lesson in this list, the EM necessary to run this lesson is reserved and subtracted from the total EM available for that site. This option may be used if a site director wants to ensure that there will be enough EM when the time comes to demo a particular lesson or to run a class using a particular lesson.
- c. Restriction lists. With these restriction lists, site directors may restrict certain physical stations (which are part of their logical site) to a particular set of groups or accounts. If someone tries to sign in at a restricted terminal, a message will state that access to that port is restricted. This option is not useful on an ASCII-only system since terminals are not assigned to a particular PLATJ site and station.
- d. Author deletion. If author deletion is turned "on", students have a priority over authors and may back authors out of lessons if EM is needed for the student to be able to execute his lesson.
- e. Message to entire site. Allows site directors to send

messages to all terminals currently signed into their logical site.

- f. Backout entire site. Allows site directors to backout all terminals currently signed into their logical site.
- g. Auto sign on options. These options allow the site directors to assign a particular sign-on to a specific station on their logical site. This makes that sign-on the default sign-on for that station. A user with that sign-on only needs to enter his password to sign on to the application. Other users may use the same station but they must enter their full sign-on to use it. This option is not useful on an ASCII-only system since terminals are not assigned to a particular PLATJ site and station.
- h. Current usage controller. This option allows site directors to use the system usage controller. Through this lesson ("enforcer"), site directors may prevent users in their site from executing certain lessons (usually games). Restrictions may be limited to certain times of the week, if desired.
- 9. Site director options. These options allow site directors to give other users limited or unlimited access to logical site information and powers.

## 20 Application Utilities

## 20.1 CMDMP

This program provides a full central memory dump of a job to a specified file. It is used in all PLATO job load procedures to provide dumps to be used to investigate all PLATO-related problems.

The first record of the file is a header record which contains the name of the job which called the dump program and the date and time of the dump. The remainder of this file is a series of 1000-word records which contain the CM dump. The file is terminated by an end-of-file (EOF).

The control statement format is:

```
CMDMP(file)
```

where:

file = file to dump to (default = CMFILE)

## 20.2 CONDIX

This procedure is called to load a CONDENSOR at a specified absolute control point or to a control point relative to the control point assigned to the PLATO application with the "ENABLE,PLA,cp." IPRDECK entry.

For example, if the CP parameter in the procedure call is a simple integer, the CONDENSOR will be loaded at the specified control point.

If the relative control point form is used, the specified integer is added to or subtracted from the control point assigned to the PLATO application. For example, if the IPRDECK entry is "ENABLE,PLA,3.", entering "CONDIX(CP=3+45)" will load the CONDENSOR at control point 7.

All control point numbers are assumed to be in octal.

Dollar signs are required around the relative control point number, but not the absolute control point number.

The control statement formats are:

```
CONDIX(CP=cp,V=vrs)
CONDIX(CP=+r$,V=vrs)
CONDIX(CP=$-r$,V=vrs)
```

where:

cp = control point number for the CONDENSOR  
(default = any open control point)

vrs = software version (default = PLATDD)  
This parameter is used only on the PLATO development system.

+r = positive relative control point number  
(dollar signs are required)

-r = negative relative control point number  
(dollar signs are required)

### 20.3 CONFIG

This procedure obtains a copy of the current PLATO configuration file as a local file named CONFIG.

The control statement format is:

CONFIG(V=vrs)

where:

vrs = software version (default = PLATDD)  
This parameter is used only on the PLATO development system.

### 20.4 EFRDMP

This program is called to dump the extended flag registers used by multi-executor/multi-formatter systems which run Extended Seal-conductor Memory (ESM) in ESM mode.

The control statement formats are:

EFRDMP.

EFRDMP(base,nstat)

where:

base = base flag register to be dumped  
The default is to use the contents of CCL "register" R2, which will contain the value of the "efrp" PLATO configuration file entry.

nstat = number of stations to dump  
The default is to use the contents of CCL "register" R3, which will contain the value of the "nstat" PLATO configuration file entry. This must be a multiple of 32.

### 20.5 E4DMP

The program copies the extended memory in use by a job to a file.

The first record of the file is a header record which contains

the name of the job which called the dump program and the date and time of the dump. The remainder of this file is a series of 1000b-word records which contain the EM dump. The file is terminated by an end-of-file (EOF).

The control statement format is:

```
EADMP(file,fwa,lwa)
```

where:

file = local file to dump to

fwa = starting address to dump

lwa = ending address to dump

## 20.6 EMDTAPE

This procedure is called by all PLATO jobs to dump extended memory and submit a job which requests a tape to save all the PLATO dump files. It is called after a job has been terminated abnormally.

This procedure dumps a job's extended memory to file PLATEM under user name PLATDMF (user index 37773b) and submits a job which requests a tape to copy all PLATO dump files to.

The control statement format is:

```
EMDTAPE(DF=df,VSN=vsN,ABT@USY=xxx)
```

where:

df = EM dump file name (default = PLATEM)

vsN = tape VSN (default = PLATEM)

xxx = yes - abort if EM dump file busy (default)  
no - wait for EM dump file, then exit

## 20.7 EXEC

This procedure is used to load an additional PLATO executor job.

The control statement format is:

```
EXEC.
```

## 20.8 FORMCMD

This procedure formats PLATO dump files to be copied to a tape. File LDMAP is expected to hold the job's load map (if any), XP is expected to hold the job's exchange package dump, EFR is expected to hold the job's extended flag register dump (if any) and CMDJMP is expected to hold the job's control

memory dump. On completion, the job's dump file is formatted as follows:

```

dayfile
load map (not used by mastor or mastorn)
EFR dump (plato, frammat/format only)
exchange package dump
header record for CM dump (see CMDMP for format)
CM dump records (see CMDMP for format)

```

See COPYPD for the format of a PLATO dump tape.

The control statement format is:

```
FORMCMD.
```

## 20.9 FRAMX

This procedure is called to load FRAMAT at a specified absolute control point or to a control point relative to the control point assigned to the PLATO application with the "ENABLE,PLA,cp." IPRDECK entry.

For example, if the CP parameter in the procedure call is a simple integer, FRAMAT will be loaded at the specified control point.

If the relative control point form is used, the specified integer is added to or subtracted from the control point assigned to the PLATO application. For example, if the IPRDECK entry is "ENABLE,PLA,3.", entering "FRAMX(CP=\$+2\$)" will load FRAMAT at control point 5.

All control point numbers are assumed to be in octal.

Dollar signs are required around the relative control point number, but not the absolute control point number.

The control statement formats are:

```

FRAMX(CP=cp,V=vrs)
FRAMX(CP=$+r$,V=vrs)
FRAMX(CP=$-r$,V=vrs)

```

where:

- cp = control point number for FRAMAT  
(default = any open control point)
- vrs = software version (default = PLATDD)  
This parameter is used only on the PLATO development system.
- +r = positive relative control point number  
(dollar signs are required)
- r = negative relative control point number

(dollar signs are required)

## 20.10 MASJJB

This program is a control card translator for jobs which are submitted by MASTOR.

MASJJB reads the job card and following control statements from a specified input file and generates appropriate RFL, and SETTL control statements onto the specified output file. If the "NC" parameter is not present, the output file is copied to the current control statement file. This command is automatically inserted into all jobs submitted through MASTOR.

The following is an example of the input and output files.

input file	output file
job(cml000,t7777)	user(aaa)
user(aaa)	sett1(7777)
...	rfl(1000)
control cards	...
...	control cards
...	...

The control statement formats are:

```
MASJJB(ifile,ofile)
```

```
MASJJB(ifile,ofile,NC)
```

where:

ifile = input file

ofile = output file

NC = no-copy option

## 20.11 MFNX

This procedure is used to attach the required master files to the MASTOR control point.

The control statement format is:

```
MFNX(M=a)
```

where:

a = the desired attach mode (default = R)

This procedure is system-unique, and must be maintained locally. An example is shown below. The RESOURC and ATTACH commands will vary on each system, but the remainder should be the same. It is important that each ATTACH

command have a "+M=4" parameter so that the access mode may be passed. The PLATO division sign is the same as the equivalence sign used by NDS in CCL procedures. It is produced under O26 by using an upper case O (zero).

```

.proc,afnx,m=r.
.*
.*      attach plato master files.
.*
.*      m = access mode, default is r.
.*
sul(377773b)  $$ or "user(platoaf)"
resourc(dl=3)
.*
attach(binary/+m=m,pn=binary,r=dl)
.*
attach(system/+m=m,pn=platoa,r=dl)
attach(amast/+m=m,pn=platoa,na,r=dl)
attach(bmast/+m=m,pn=platoa,na,r=dl)
attach(cmast/+m=m,pn=platoa,na,r=dl)
.*
attach(dmast/+m=m,pn=platoa,na,r=dl)
attach(emast/+m=m,pn=platoa,na,r=dl)
attach(fmast/+m=m,pn=platoa,na,r=dl)
attach(gmast/+m=m,pn=platoa,na,r=dl)
.*
revert.

```

## 20.12 PLATX

This procedure is called to load a PLATO executor at a specified absolute control point or to a control point relative to the control point assigned to the PLATO application with the "ENABLE,PLA,cp." IPRDECK entry.

For example, if the CP parameter in the procedure call is a simple integer, the executor will be loaded at the specified control point.

If the relative control point form is used, the specified integer is added to or subtracted from the control point assigned to the PLATO application. For example, if the IPRDECK entry is "ENABLE,PLA,3.", entering "PLATX(CP=\$+13)" will load the executor at control point 4.

All control point numbers are assumed to be in octal.

Dollar signs are required around the relative control point number, but not the absolute control point number.

The control statement formats are:

```

PLATX(CP=cp,V=vrs)
PLATX(CP=$+r$,V=vrs)
PLATX(CP=$-r$,V=vrs)

```

where:



- cp = control point number for the PLATO executor  
(default = any open control point)
- vrs = software version (default = PLATDD)  
This parameter is used only on the PLATO  
development system.
- +r = positive relative control point number  
(dollar signs are required)
- r = negative relative control point number  
(dollar signs are required)

### 20.13 PNIX

This procedure is called to load the PLATO/NAM Interface program at a specified absolute control point or to a control point relative to the control point assigned to the PLATO application with the "ENABLE,PLA,cp." IPRDECK entry.

For example, if the CP parameter in the procedure call is a simple integer, PNI will be loaded at the specified control point.

If the relative control point form is used, the specified integer is added to or subtracted from the control point assigned to the PLATO application. For example, if the IPRDECK entry is "ENABLE,PLA,3.", entering "PNIX(CP=\$+3\$)" will load PNI at control point 6.

All control point numbers are assumed to be in octal.

Dollar signs are required around the relative control point number, but not the absolute control point number.

The control statement formats are:

```
PNIX(CP=cp,FAMILY=xxx,DRD=n)
PNIX(CP=$+r$,FAMILY=xxx,DRD=n)
PNIX(CP=$-r$,FAMILY=xxx,DRD=n)
```

where:

- cp = control point for PNI  
(default = any open control point)
- xxx = alternate family where terminal resident load files are to be found
- n = PNI/NAM ordinal (default=1)  
This parameter is used only in multi-mainframe configurations where multiple copies of NAM and PNI are running on more than one mainframe.
- +r = positive relative control point number

(dollar signs are required)

-r = negative relative control point number  
(dollar signs are required)

#### 20.14 PRDUTE

This program is used to schedule a job to a specified absolute control point or to a control point relative to the control point assigned to the PLATO application with "ENABLE,PLA,cp." IPRDECK entry.

For example, if the CP parameter in the program call is a simple integer, the job will be loaded at the specified control point.

If the relative control point form is used, the specified integer is added to or subtracted from the control point assigned to the PLATO application. For example, if the IPRDECK entry is "ENABLE,PLA,3.", entering "PRDUTE(X,CP=+5)" will schedule the job file X at control point 10.

All control point numbers are assumed to be in octal.

The control statement formats are:

```
PRDUTE(Job,CP=cp)
PRDUTE(Job,CP=+r)
PRDUTE(Job,CP=-r)
```

where:

job = name of a local file containing control cards to be submitted as a new job

cp = control point where new job is to run  
The default is any open control point.

+r = positive relative control point number

-r = negative relative control point number

#### 20.15 SUBMIT4

This program is used to submit a NDS file or a block of a PLATO file for execution as a NDS job. See lesson "nosajds" for the control statement format.

#### 20.16 VERSX

This procedure obtains the executable binaries for PLATO jobs, depending on what version of the software is being run. This procedure is only needed on the PLATO development system.

The control statement format is:

VERSX(V=vrs)

where:

vrs = software version (default = PLATDD)

## 21 Operations utilities

## 21.1 CONSOLE

The computer console may be used as a PLATO terminal on the CC545 console as follows. This program does not work on the CDC 721 console.

- a. Type in "X.CONSOLE." at the computer console.
- b. If this is the first time "X.CONSOLE." has been entered since the PLATO application was loaded you will be taken to a display that says "LESSON DESIRED". If "X.CONSOLE." has been entered before, you will be taken to the exact place where the person was when "X.CONSOLE." was exited.

A keyboard diagram will appear on the right hand screen. In order for the console in "X.CONSOLE." to approximate all the keys of the PLATO keyboard, there are two keyset diagrams. The "left blank" key (left arrow with bar on the CDC 721) toggles between them. It is important to note, however, that the meaning of the console keys varies according to which of the two diagrams is displayed on the right screen.

Here are some important keys to remember on the CC545 display console:

- "cr" = NEXT
- "right blank" = SHIFT
- "left blank" = an alternate keyboard diagram
- "right blank", "left blank", ")" = SHIFT-STOP
- "left blank", "right blank", "x" = drop CONSOLE job
- "left blank", "x" = drop display and return to DSD display without dropping CONSOLE job. The display may be returned to CONSOLE by pressing "\*" on the DSD display

Here are some important keys to remember on the CDC 721 display console:

- "right arrow with bar" = SHIFT
- "left arrow with bar" = an alternate keyboard diagram
- "right arrow with bar", "left arrow with bar", "divide" = SHIFT-STOP
- "left arrow with bar", "right arrow with bar", "x" = drop CONSOLE job
- "left arrow with bar", "x" = drop display and return to DSD display without dropping CONSOLE job. The display may be returned to CONSOLE by pressing the "square" key ("F15") on the DSD display.

NOTE: When you enter "X.CONSOLE.", you initially have no sign-on, and group "o" powers by default. If you wish to use your own sign-on, execute lesson "plato" and type in the desired sign-on.

## 21.2 CJPYPD

This procedure copies all PLATO dump files to a tape. It allows the operator to KILL the tape copy job submitted after a dump of a PLATO control point has completed and re-start it later. If the tape copy job is terminated by a DROP command, the job will PURGE all dump files. The format of the PLATO dump tape is as follows:

Each job has a file on the dump tape for its output records and CM dump records:

```

1  mastor / mastorn
2  plato (executor 0)
3  executor 1
4  executor 2
5  framat
6  format
7  condensor 0
8  condensor 1
9  condensor 2
10 (not used)
11 pni
12 (reserved for CDC use)

```

These files are followed by the EM dump file.

See FORMCMD for the format of each dump file.

The control statement format is:

```
CJPYPD,DF=file, VSN=vsno
```

where:

```

file = EM dump file name (defaults to PLATEM)
vsno = dump tape VSN (defaults to PLATEM)

```

## 21.3 DDPT

This is an on-line diagnostic program used to test a JDP or low-speed port concurrently with normal operations.

A "distributed data path" (DDP) allows a PP program to directly access ECS-type extended memory. A "low speed port" performs the same function for ESM whether it is run in ECS or ESM mode. The PLATO application must be running since DDPT obtains EM from the EM reserved through the "objs" PLATO configuration file entry.

The algorithm in the program is as follows:

- a. Format a pattern in PP memory using the minimum EM transfer length specified in the control statement. The patterns reside in CM and are shifted left 1 bit and written back to CM after each use.

- b. Write the patterns to EM at the relative EM FWA.
- c. Destroy the pattern in PP memory.
- d. Read the patterns back into PP memory.
- e. Verify that what was read back was correct.
- f. Repeat all previous steps with each remaining pattern.
- g. Repeat all the steps above for each of the 60 shifts.
- h. Increment the transfer length and, if less than or equal to the maximum transfer length and if the LWA is still within FLX, repeat steps a through g. If not, then one pass through the main PP loop has been completed.
- i. Increment the FWA and repeat steps a through h until the main loop has been executed "n" times as specified in the control statement.

The control statement format is:

```
DDPT(CH=cn,MN=mn,MX=mx,FL=fl,EL=el,N=n,FW=fw)
```

where:

- cn = DDP/low speed port channel number; if not specified, the program will look for equipment of type "D1" to determine the channel number
- mn = minimum EM transfer length. The default value is 1.
- mx = maximum EM transfer length. The default value is 100d.
- fl = EM FL/1000 to request from MASTER; it must be less than or equal to the amount available via the "bjecs" PLATO configuration file entry. The default value is 1.
- el = error limit, where 0 implies no limit. The default value is 100d.
- n = number of times through main PP loop, where 0 implies run until dropped. The default is 1.
- fw = relative EM FWA within your assigned EM FL. The default value is 0.

All parameters are assumed to be octal unless otherwise specified.

This procedure prints all information for a job from a PLATO dump tape or disk file. This includes the exchange package dump, a full central memory dump, the extended flag register dump (if running ESM in ESM-mode), a full load map and the job dayfile.

The control statement format is:

```
DUMPPRT(f,NJCM,TID=tid,VSN=vsn)
```

where:

f = one of the following strings:

```
mastor
mastorn
plato (or exec0)
exec1
exec2
framat
format
condan (or cond0)
cond1
cond2
pnl
```

NJCM = disable printing of the job's central memory

tid = string used to identify the tape for the operator (e.g., "e-12")

vsn = VSN of the dump tape (if needed)

If file "tape" is assigned at the beginning of this procedure, no tape request is made. File "tape" is returned by this procedure.

## 21.5 ECSTST

This program is used to test extended memory. The PLATO application must be loaded as it uses 10000 words of the extended memory reserved through the "ogecs" configuration file entry.

The control statement format is:

```
ECSTST.
```

## 21.6 EMPRT

This procedure prints requested parts of an EM dump from a PLATO dump tape or disk file.

The control statement format is:

```
EMPRT(fwa,lwa,tid,vsn)
```

where:

- fwa = the first word of the EM to be printed
- lwa = the last word + 1 of the EM to be printed
- tid = a string used to identify the tape for the operator (i.e., "e-1?")
- vsn = the VSN of the dump tape (if needed)

If file "tape" is assigned at the beginning of this procedure, no tape request is made. File "tape" is returned by this procedure.

### 21.7 ESM

This program is used to load ESM relocation memory, and to monitor and log errors. See the PLATO Configuration Handbook for information on using this program.

### 21.8 MEMPRT

This program is used to print central and extended memory from PLATO dump files.

The control statement format is:

```
MEMPRT(file,fwa,lwa,f)
```

where:

- file = the name of the dump file
- fwa = the starting address of the memory to be printed
- lwa = the ending address + 1 of the memory to be printed (default lwa = fwa + 1)
- f = the output format
  - 0 = display code (default)
  - 3 = binary

### 21.9 MFPACK

This procedure is used to reformat a master file to give it a different name, length, and type. The new master file retains all the PLATO files contained in the old master file.

The control statement format is:

```
MFPACK(MF=mf,N=n,SP=sp,PT=pt,PN=pn,R=r,hPN=hpn,NR=nr)
```

where:



**mf** = old master file (NJS) name  
**n** = new master file name - both internal and NJS name (default = old master file name)  
**sp** = new master file length (uses equipment-related default -- see MFCREAT)  
**pt** = new master file type (default = "general")  
**pn** = NJS pack which holds old master file (default = system pack)  
**r** = device type of "pn" (default = system default)  
**npn** = NJS pack to hold the new master file (default = system pack; unless "n" = "mf" where the default = "pn")  
**nr** = device type of "npn" (default = system default unless "n" = "mf" where the default = "r")

### 21.10 MFTCOPY

This procedure is used to copy master files from disk to tape for shipment to another system. This procedure also generates MFPRINT listings of the master file copied to tape upon completion of the copy.

The control statement format is:

```
MFTCOPY(MF=af,PN=pn,R=r,VSN=vsN,D=d,L=§§§,LIST=ifn)
```

where:

**af** = master file name  
**pn** = NJS pack name (default = system pack)  
**r** = device type (default = system default)  
**vsN** = tape VSN (default = "mf")  
**d** = tape density (default = PE - 1600 BPI)  
**l** = tape label - must be enclosed in "§"  
**ifn** = output file name (default = JUTPUT)

The master file is always assumed to be defined under user name PLATDMF (user index 3777730).

### 21.11 MFTLOAD

This procedure is used to copy master files from tape to disk to install lessons received from another system. This procedure also generates MFPRINT listings of the master file loaded to disk upon completion of the load. The master file is always defined under user name PLATDMF (user index 3777730). The internal master file name by which PLATD will know the master file is always set the same as the NJS file name.

The control statement format is:

```
MFTLOAD(MF=af,PN=pn,R=r,VSN=vsN,D=d,LIST=ifn,FAMILY=faa)
```

where:

**af** = destination file (default = "cut")  
**pn** = NJS pack name (default = system pack)

r = device type (default = system default)  
 vsn = tape VSN (default = value of mf parameter)  
 d = tape density (default = PE - 1600 dPI)  
 ifn = output file name (default = OUTPUT)  
 fam = family name (default = default family)

## 21.12 MFUTIL

The program MFUTIL is a collection of utilities used to create and edit PLATO master files.

MFADD - attach master file to MASTOR  
 MFALTER - change master file name and/or type  
 MFCREAT - create master file  
 MFLIST - list names of PLATO files on a master file  
 MFPRINT - print master file information  
 PFDEST - destroy PLATO files on a master file  
 PFIN - create PLATO files on a master file  
 PFOUT - copy PLATO files on a master file to a NDS file  
 PFUSE - copy PLATO files on a master file to a NDS file

While master files may be attached, destroyed and copied using standard NDS commands, there are special commands for creating, altering and printing them. These commands may be used at the computer console or in a job submitted from a PLATO terminal. To use these commands, you must first attach the master file (except when you are using MFCREAT, since no master file yet exists).

See the following sections for documentation of each of these utilities.

### 21.12.1 MFADD

The MFADD control statement is used to attach master files to the MASTOR control point. This command is used in a job submitted by lesson "ldr" when loading a master file.

The control statement format is:

```
MFADD(MF=ifn)
```

where:

ifn = local file name of master file (default = MF)

### 21.12.2 MFALTER

The MFALTER control statement enables the user to change either the internal master file name or the master file type. When a master file is created, the internal name is the same as the NDS permanent file name. It is not required that these names remain the same, although it is usually desirable. Thus, when you make a copy of a master file using NDS COPY commands, you can use MFALTER to change the internal name to agree with the NDS file name.

The control statement format is:

```
MFALTER(MF=ifn,V=p1,PT=p2)
```

where:

ifn = local file name of master file (default = MF)

p1 = new internal name for master file (default = no change)

p2 = new master file type (default = no change)  
See the description of MFCREAT for a list of valid types.

### 21.12.3 MFCREAT

The MFCREAT control statement defines a NOS permanent file to be used as a PLATO master file.

The control statement format is:

```
MFCREAT(MF=pfn,PW=p1,DT=p2,PT=p3,SP=p4)
```

where:

pfn = master file name

p1 = master file password (default = none)  
This is the standard NOS permanent file access codeword.

p2 = device type on which to define the new master file (default = system default device type)  
The permitted values are DD,DG,DI,DJ,DK,DL,DM,DQ.

p3 = master file type (default = GENERAL)  
The permitted values are GENERAL, MASTER, BINARY, BACKUP and ARCHIVE.

p4 = number of PLATO file parts in the master file (default = maximum size for the device type)  
The maximum size and the default size varies with the device type in an attempt to use sizes which best utilize the available space.

The following table shows the maximum and default master file sizes by drive type. The number of full-size master files which will fit on each device type is also shown. This number depends on the number of catalog tracks for the drive being set to 1 when the device was initialized.

Device	Maximum	Master Files/Device
DD	3680	2
DG	4400	5

DI	2410	2
DJ	5159	2
DK	2410	2
DL	5159	2
DM	5000	6
DQ	5000	6

Usually master files are created under user name PLATJMF (user index 37773b). A PACKNAM or FAMILY command should precede the MFCREAT so the master file will be created on the correct pack or family.

#### 21.12.4 MFLIST

The MFLIST command lists the names of all PLATO files on a particular master file. No other information about the individual files is provided.

The control statement format is:

```
MFLIST(MF=ifn,L=p1,OP=p2)
```

where:

ifn = local file name of master file (default = MF)

p1 = name of output file (default = LIST)

p2 = listing option (default = no extra information)  
The only valid option is H, which causes a header line to be output on the listing file.

This control statement is used to produce a list of files to be used with another control statement such as PFOUT (can be used to copy all PLATO files on a master file to a NDS file).

#### 21.12.5 MFPRINT

The MFPRINT control statement produces an output file containing information from the directory of a master file and listing of the names of all PLATO files on the master file.

The control statement format is:

```
MFPRINT(MF=ifn,L=p1)
```

where:

ifn = local file name of master file (default = MF)

p1 = name of output file (default = OUTPUT)

#### 21.12.6 PFDEST

The PFDEST control statement allows destroying a list of

PLATO files on a master file.

The control statement formats are:

```
PFDEST(MF=ifn,I=p1)
```

```
PFDEST(MF=ifn,Z)p2
```

where:

ifn = local file name of master file (default = MF)  
The master file must be attached in "write" mode.

p1 = NJS file containing the list of PLATO files to be destroyed (default = INPUT)

p2 = name of PLATO file to be destroyed  
This parameter is used only if "Z" is present.  
This also causes the "I" option to be ignored.

#### 21.12.7 PFIN

The PFIN control statement allows adding PLATO files to a master file. It does this by copying lesson source from a NJS file to a PLATO master file. To be valid, the data on the NJS file must have been generated by a previous PFOUT control statement.

The control statement format is:

```
PFIN(MF=ifn,I=p1,NA)
```

where:

ifn = local file name of master file (default = MF)  
The master file must be attached in "write" mode.

p1 = NJS file containing lesson source (default = SOURCE)  
Each PLATO file must be a single record on this file. An end-of-file signals the end of input.  
This is the "I" format used in PFOUT).

NA = if present, PFIN will not abort if there is not enough room in the master file or if the name of a PLATO file to be added is already on the master file

#### 21.12.8 PFOUT

The PFOUT control statement allows copying PLATO files from a master file to a NJS file.

The control statement formats are:

```
PFOUT(MF=ifn1,I=ifn2,O=ifn3,F=fmt,NA)
```

```
PFOUT(MF=ifn1,O=ifn3,F=fmt,NA,Z)pfn
```

where:

lfn1 = local file name of master file (default = MF)

lfn2 = NOS file containing list of files to copy

lfn3 = NOS file to copy to (default = SOURCE)

fmt = output format (default = I)

format	meaning
I	Each PLATO file is a single record with 320 words/block. The output file ends with an end-of-file.
B	Each block is a single record of 320 words. Each PLATO file ends with an end-of-file, and the output file ends with an end-of-information.

NA = if present, PFDUT will not abort if a file in the list cannot be found

pfn = name of PLATO file to be copied  
 This parameter is used only if "Z" is present  
 This also causes the "I" option to be ignored.

#### 21.12.9 PFUSE

The PFUSE control statement allows copying the contents of PLATO files from a master file to a NOS file for submittng. It is very similar to PFDUT except that only partialled-in source blocks will be copied and only the actual length of each block will be copied instead of the full 320 words per block.

The control statement formats are:

PFUSE(MF=lfn1,I=lfn2,O=lfn3,NA)

PFUSE(MF=lfn1,O=lfn3,NA,Z)pfn

where:

lfn1 = local file name of master file (default = MF)

lfn2 = NOS file containing list of files to copy

lfn3 = NOS file to copy to (default = COMPILE)

NA = if present, PFUSE will not abort if a file in the list cannot be found

pfn = name of PLATO file to be copied  
 This parameter is used only if "Z" is present.  
 This also causes the "I" option to be ignored.

## 21.13 NETPRT

This program is used to print the network database maintained by lesson "pnet". The control statement is automatically formatted by lesson "snetprt".

## 21.14 PDCAT

This procedure produces a catalog of the contents of a PLATO dump tape on file OUTPUT. It assumes that the dump to be cataloged is to be requested unless a local file named TAPE exists. File TAPE is returned if requested by this procedure, otherwise it is rewound.

The control statement formats are:

PDCAT(tid)

PDCAT(TID=tid,VSN=vsn)

where:

tid = external tape identifier

vsn = VSN for tape (default = PLATEM)

## 21.15 PDPRT

This program is used to print master file directories. See lesson "nosalds" for the control statement format.

## 21.16 REQPACK

This program allows a batch job to pause until a desired NDS pack is mounted and available.

The control statement formats are:

REQPACK(PN=pn)

REQPACK(pn)

REQPACK(PN=pn,R=dt)

REQPACK(pn,R=dt)

REQPACK(R=dt)

REQPACK. (Clears pack name and device type.)

where:

pn = name of NDS pack

dt = default device type for all future permanent file operations

This program is similar to the NDS PACKNAM command. If

the requested NDS pack is not mounted and available for use, the job will issue a flashing B-display message requesting the operator to mount the pack. The job then waits until the pack has been mounted.

#### 21.17 WAIT

This control statement is used for job timing. It allows a job to wait for a period of time, or for an operator action.

The control statement formats are:

WAIT(time)

WAIT. text

where:

time = number of seconds to pause before proceeding  
If this parameter is absent, the job waits for an operator to give the job a "GO".

text = message to flash on B-display



## 22 Print Utilities

## 22.1 ACCPRT

This program is used to generate a print of the file management logs ("acclog0", "acclog1", etc.). It is called when using the option in "accounts" to print the file management log. The control card below is automatically formatted by answering various prompts in the "print file management logs" options of "accounts".

The control statement format is:

```
ACCPRT,logname,accname,startdate,enddate,snowflg
```

where:

- logname = The file name of the particular log to be printed. This should be one of the standard log names: "acclog0", "acclog1", etc.
- accname = The name of the account for which you want to print information. If "0" is entered, information will be printed for all accounts.
- startdate = The starting date for information to be printed in the format mm/dd/yy. Information prior to this date will be ignored. If "0" is entered, information will be printed starting at the earliest date in the log.
- enddate = The ending date for information to be printed in the format mm/dd/yy. Information after this date is ignored. If "0" is entered, all information up to the current date will be printed.
- snowflg = A flag determining which operations are to be included in the print.

showflg	meaning
"all"	Print all file operations.
"set2"	Only print operations done by normal account directors (e.g., create a file, rename a file, etc.).
"set3"	Only print operations done by system controllers (e.g., create an account, change the number of subscriptions, etc.).

## 22.2 D3CPRT

This program is used to print "documentor" type files. The control card is automatically formatted by answering various

prompts in lesson "prints".

### 22.3 DPRINT

This program is used to print "student data" type files. The control card is automatically formatted by answering various prompts in lesson "prints".

### 22.4 MDDPRT

This program is used to print PLATO Learning Management "module" type files. The control card is automatically formatted by answering various prompts in lesson "prints".

### 22.5 NPRINT

This program is used to print "general notes" and "student notes" type files. The control card is automatically formatted by answering various prompts in lesson "prints".

### 22.6 PLMPRT

This program is used to print PLATO Learning Management "curriculum" type files. The control card is automatically formatted by answering various prompts in lesson "prints".

### 22.7 PPRINT

This program is used by systems with upper/lower case print trains to convert PLATO print files to ASCII. This control statement can be included in the print routines submitted by lesson "prints" by editing file "prtsub".

The control statement format is:

```
PPRINT(I=Input file,O=Output file)
```

### 22.8 TPRINT

This program is used to print TUTOR "lesson", "dataset" and "nameset" type files. The control card is automatically formatted by answering various prompts in lesson "prints".

## 30 Problem Reporting

### 30.1 Purpose

This section defines the software maintenance interface between Control Data Corporation (CDC) PLATO Software Maintenance group and PLATO software users. The fundamental purpose of this document is to ensure that:

- The software maintenance interface between CDC and the user is defined and understood by both parties.
- The interface conforms with CDC standards covering software maintenance organizations.
- The necessary internal controls for effective software maintenance process are established and maintained.
- The responsibilities of CDC and the user in a cooperative effort to ensure responsive solution of problems are established.

Terminology specific to this document will be found under the "Definitions" section.

### 30.2 Definitions

#### Application

In this document, "application" and "network" both refer to the PLATO application.

#### Local CDC Representative

The CDC sales representative or manager who is responsible for your application.

#### Local PLATO Systems Staff

That group of people on each system who have the responsibility for reporting problems to CDC. This usually includes the operations and consulting groups.

#### PLATO Development System

The "home" system for the PLATO Software Maintenance organization. (Currently the "pdev" system.)

#### PLATO Hotline

Assistance to customers using the PLATO application may be obtained by calling the "PLATO Hotline". The PLATO Hotline number is 1-800-328-4915 (outside Minnesota) or 1-612-375-8111 (Minnesota only).

If you call the Hotline, give the person on duty the following information:

- \* Your name and location (city, state), and phone

number.

- \* The name of the system you use.
- \* A description of your problem.
- \* If you are experiencing communications problems, such as the error (err) light going on, the staff member helping you may ask your site, or station number. You will find this information by pressing DATA from the Welcome display.

#### PLATO Software

PLATO software includes all system software and CDC Published Courseware. See PLATO System/Courseware Maintenance.

#### PLATO Software Maintenance

See PLATO System/Courseware Maintenance.

#### PLATO System/Courseware Maintenance

This CDC group is part of PLATO Development and is responsible for maintaining all PLATO system software and CDC Published Courseware. Throughout this document this group will be referred to as "PLATO Software Maintenance."

#### Programming System Report (PSR)

A "Programming System Report" (PSR) is the name of the report submitted by a particular system to the PLATO Software Maintenance organization.

### 30.3 User Responsibilities

#### 30.3.1 Monitoring Your System

Your system should be monitored on a regular basis by designated site personnel. Each problem should be evaluated to determine if it really is a system software or courseware problem, or a problem of unknown origin which should be reported to PLATO Software Maintenance.

Notes file "sysin" is the default file for errors and user comments on system lessons. Notes file "lessnotes" is the default file for errors and user comments on CDC published courseware. These two files should be read daily.

Error messages for which the documented action is to call PLATO Software Maintenance should be reported, as well as any message for which the action is unknown.

System crashes may or may not be the result of PLATO software. Site analysts should investigate each crash as completely as possible. If it appears that PLATO software is involved in some way, the crash should be reported immediately. If a problem appears to be due to other causes (i.e., non-PLATO software, operational errors, suspected hardware problems), it should only be reported to PLATO Software Maintenance if assistance is being requested to help the local system.

staff resolve the problem.

Refer to the "Problem Reporting Procedures" and "Information Required" sections for guidelines on reporting crashes and other problems. The format used for reporting problems is called the "Programming System Report" (PSR.)

### 30.3.2 Reporting Problems

#### 30.3.2.1 Problem Reporting Procedures

Problems reported by all supported PLATO systems must be prepared as follows:

- a. Enter a concise, complete description of the problem on a standard PSR form, or on a Courseware Action Request form (see the section on Information Required). These are the only recognized mechanisms for reporting problems. The following are not valid means of reporting problems: personal contact with PLATO Software Maintenance analysts through telephone calls, person-to-person discussions, personal notes, or TERM-talk.
- b. Assign a priority to the problem which reflects its impact (see section "Submitter's Priority Level").

For all CRITICAL problems the local PLATO system staff must TWX/Telex or call the PLATO Hotline (refer to the information given earlier). This allows PLATO Software Maintenance analysts to begin working on the problem before the PSR is received.

NOTE: even though PLATO Software Maintenance is contacted directly, a PSR containing a concise description of the CRITICAL problem must be written to assure inclusion in status reporting. Refer to the section on "Information Required" for reporting guidelines.

- c. The PSR submitter is responsible for contacting users to obtain any additional information requested by PLATO Software Maintenance.

#### 30.3.2.2 Information Required

The guidelines in this section have been designed to help you include information which is necessary for the timely and accurate resolution of a PSR.

All PSRs should follow the general format outlined below:

- State only the facts in your PSR. Avoid offering personal opinions or comments that might tend to be confused with the actual problem.
- Isolate the location of an error to the smallest area possible. For example, to a specific file,

lesson, module, unit, or set of commands in a lesson.

- List the sequence to follow to duplicate the error. List any unique conditions which would affect the duplication of the error.
- Submit all PSRs in English or provide an English translation.
- Report only ONE problem in each PSR. DO NOT report several different problems in the same note, no matter how small the errors appear. If it cannot be determined if a problem is the same as another, write two different PSRs.

The following guidelines are categorized to help you include information appropriate to your problem:

#### HARDWARE PRODUCT AND MODEL INVOLVED

- Extended memory type.
- Mainframe type.
- Terminal type (Viking, IST-III, etc...)
- Resident type
- Network name (CIU, Direct ASCII (PNI), etc...)
- Peripheral equipment (disk drive, modem, printer, etc...)

#### OPERATING SYSTEM/LEVEL INVOLVED

- Version/Level (NDS 2.3 - 617)
- Release Level (PLATO Release 33.2)
- Published courseware delivery number.

#### FILE / COURSE / LESSON / MODULE INVOLVED

- Name of the course and/or lesson involved (Beginning Biology, lesson "s(doc", etc...)

#### FLEXIBLE DISK INVOLVED

- Disk serial number
- Disk title
- Publication number
- Manufacture date
- Specific lesson or module name you were studying.

#### HARDCOPY MATERIALS INVOLVED

- Publication number
- Title of the course, manual, audiotape, etc...
- Page, chapter, or section number.

#### ERROR MESSAGES INVOLVED

- Quote error messages exactly, as a small difference between the report and actual wording can be critical to the analyst solving the problem.

#### FILES OR DATA LOST

- Make copies of affected files and include their names in the PSR so that PLATO Software Maintenance analysts may inspect them if necessary.

#### SUPPORTING MATERIALS

- For problems classified as CRITICAL, send any paper dumps and/or tape copies of PLATO dump files or express deadstart dump (EDD) tapes to the following address:

Control Data Corporation  
PLATO Software Maintenance  
MEVJ3P  
ATTN: PSR Manager  
511 11th Ave. So.  
Minneapolis, Mn. 55415-1579

#### 30.3.2.3 Setting a Priority Level

It is the responsibility of the local system staff to assign a priority number to each PSR sent. This priority number reflects the impact of the problem and will affect the speed with which a problem will be resolved (refer to the section on the "Resolution Schedule").

Published courseware users should consider that the priority of a problem may be higher if the problem is impacting a large number of users.

There is no commitment that CDC will assign the same priority to the problem. Changes made by CDC to the PSR priority will be communicated to the submitter. If the submitter does not agree with a change in priority, a request to re-evaluate the change should be made to the local CDC representative.

#### PRIORITY 4 - CRITICAL

CRITICAL problems include:

- ONLY repeated or lengthy system interruptions such as MASTOK or PLATO crashes.
- Major lessons, courses, or features being unusable.
- Massive permanent loss of data (destroyed files, data stored incorrectly).
- No known "work-around" is available.
- Possible liability against CDC or customer. E.g., the user is instructed to do something that could cause bodily harm to himself or others.

#### PRIORITY 3 - URGENT

URGENT problems include:

- Isolated full-system service interruptions.
- PLATO-related problems which do not cause a full-system interruption of service (such as in CONDENSOR, or in FRAMAT), unless the problem occurs frequently, or full-system service interruptions are required to fix the problem.
- Major user difficulties in using a courseware package or released PLATO feature. E.g., a student is not automatically given credit for a completed lesson.
- Permanent loss of data occurring infrequently.
- A "work-around" is available but would cause substantial difficulty.
- Possible liability against CDC or customer. E.g., content/instructions are contrary to existing laws or regulations.

#### PRIORITY 2 - SERIOUS

##### SERIOUS problems include:

- A portion of a courseware package or released PLATO feature is not working correctly or is documented incorrectly.
- A "work-around" may be available.
- A problem or system interruption which can be circumvented or which occurs rarely or under unusual circumstances.

#### PRIORITY 1 - MINOR

##### MINOR problems include:

- Ambiguous error messages or documentation or HELP sequences are incomplete or outdated. This includes hardcopy manuals such as PLATO Operations Guide or Administrative (Ad) Guide.
- Display problems (misplaced text, overwrites).
- Errors in grammar, spelling, or punctuation.
- Inconsistencies or irregularities in the function of a released PLATO feature or courseware.
- Minor content disparities.

#### CRITICAL/URGENT GUIDELINES

Many problems may seem CRITICAL or URGENT. Therefore, the following tests may be helpful in classification of the problem.

- a. If the problem can wait for a full test of corrective code (the next PLATO software release or Courseware release) rather than risk the implementation of uncertified code, the problem is less than CRITICAL or URGENT and should be classified as SERIOUS.
- b. If it is possible to tolerate a problem, rather than generate a new system or course after the corrective



binaries or course are available, the priority should be classified as SERIOUS or MINOR.

### 30.4 CDC Responsibilities

#### 30.4.1 User Interface Organization

It is a CDC policy that each organization responsible for software maintenance shall provide a User Interface organization to which the local system staff or Professional Service Division (PSD) field analyst can interface for problem assistance. This organization shall be responsible for:

- Administering the critical PSR process.
- Providing remote assistance on problem identification, work-around, and/or resolution.
- Providing on-site assistance for catastrophic situations.

These functions are provided for by standard CDC policies and procedures.

#### 30.4.2 PSR Status Reporting

##### 30.4.2.1 Resolution Schedule

Problems will normally be resolved in the timeframes shown below:

Priority	Turnaround
CRITICAL	10 working days
URGENT	30 working days
SERIOUS	60 working days
MINOR	120 working days

Turnaround for CRITICAL and URGENT problems is defined as the time from receipt of the PSR to the time the fix is available for delivery to the submitting system.

Updated software which corrects problems classified as CRITICAL or URGENT will be delivered to those systems reporting the problems as soon as the software has been corrected. All other systems will receive the updated software with the next scheduled release.

Turnaround for SERIOUS and MINOR problems is defined as the time from receipt of the PSR to the time the fix is made on the PLATO Development system. Updated software will be available to all systems with the next major release following resolution of the problem.

##### 30.4.2.2 PSR Responses

On a release basis, a PSR status report is sent to the submitting local PLATO system staff. Once the report has been distributed, all resolved problems will be deleted from the problem tracking and reporting utility. I.e., once a problem has been reported as "resolved", future status reports will not contain any references to the resolved problems.

The release PSR status report will include a print of resolved PSRs for individual systems. Also, a list of all current resolved and unresolved PSRs will be sent.

The following types of responses may appear in the list of resolved PSRs or on the printed resolved PSRs:

a. DUPLICATE

PSR number of other report(s) of the same problem from the same system will be stated.

b. CANNOT DUPLICATE

The problem cannot be duplicated on the PLATO Development system when following instructions given in the PSR. For CRITICAL or URGENT problems, this response means that the problem could not be duplicated on the PSR submitter's system either.

c. REQUEST FOR SYSTEM MODIFICATION (RSM)

The PSR is considered to be a request for changes to correctly functioning software. If the submitter feels the need for this change is critical, the request should be sent to the local CDC representative.

d. FIXED

Corrective code has been generated to fix the problem. The PLATO release level or the published courseware delivery level containing the corrective code will be stated if known.

e. HARDWARE ERROR

Known details of the error will be stated.

f. USER ERROR

An explanation will be given.

g. REQUEST FOR INFORMATION

An explanation will be given.

h. DEFERRED

The problem has been deferred because:

- current or planned development will fix problem
- development has not been planned
- this problem has a minor impact on users versus a major effort to fix
- changes may be desirable but would negatively affect existing programs
- the problem involves non-supported courseware or system software

#### 40 Add-on Products

This section describes the operation procedures necessary for the separately ordered add-on PLATO products. These products include:

- a. PLATO Courseware Development and Delivery (PCD2)

40.1 PCD2

PLATO COURSEWARE DEVELOPMENT AND DELIVERY (PCD2) APPLICATION

INTRODUCTION

PLATO Courseware Development and Delivery (PCD2) utilizes major capabilities of the PLATO Lesson Authoring and Delivery Application to assist authors in developing Computer-Based Instruction (CBI) for presentation on microcomputers which include the Control Data 110, the PLATO Personal Training Station, the IBM PC, and the Zenith Z-150.

Students executing courseware written using PCD2 will notice no difference between this courseware and that written in the PLATO Authoring Language or the Micro PLATO Language.

PCD2 authors can use the following PCD2 capabilities:

- o Text (both Normal and Bold sizes)
- o Line graphics (Boxes, Circles, Arcs, Vectors, etc)
- o Author and student controlled branching
- o Conditional branching
- o Questions (student responses can be collected and evaluated)
- o Variable Test Structures (based on User and System-defined variables, for example)
- o Arithmetic calculation (of these same variables)
- o Alternate text characters (Character Sets)
- o Animation (flashing and straight-line movement of figures, for example)
- o Color commands (within delivery micro capabilities)
- o Programming language "add-ons", including jumpouts to and from Micro PLATO language generated lessons.

Authors are guided through the authoring process with prompts and menus, and an extensive on-line reference manual called "PCD2 Aids". Function-specific editors; such as the Reply Specification Editor, Frame Linking Editors, and the familiar Micro PLATO Editor; help make PCD2 easy to learn and use. A PCD2 Seminar (ASE number DP4005) is also available to help familiarize users quickly with all the features and capabilities of the PCD2 authoring system.

PCD2 is an add-on product to the PLATO Lesson Authoring and Delivery Application 1.

IBM is a registered trademark of International Business Machines Corporation.

Z-150 is a trademark of Zenith Data Systems &R. Zenith and Zenith Data Systems are registered trademarks of Zenith Radio Corporation.

PLATO is a registered trademark and PCD2 is a trademark of Control Data Corporation.

#### 40.1.1 PCD2 Operation

#### PCD2 OPERATIONS PROCEDURES

Outside of the normal operation of your PLATO application, the only operational responsibility for PCD2 is that of controlling user access to PCD2 on your system. Should you wish to control user access on a user-by-user basis, the following procedure should be followed to customize the PCD2 access list:

- a. Sign on to your system with a group "p" sign-on.
- b. Execute lesson "lpedit".
- c. Choose the option to "edit/inspect system access lists". This takes you to lesson "s0calutil".
- d. From the list of options presented, press LAB to edit or inspect an access list, and at the arrow, type "pcd2".
- e. From this point, proceed as normal to edit the access list, just as you would any other access list.

To give users access to the product, they should be given "Execute" access. Initially, "Other / Other" has this access.

To allow other people to give access to others, their access should be set to "Director". Initially, only "Other / p" and "Other / s" have this access.

To prevent someone from either gaining access to PCD2 or from providing access to someone else, their access should be set to "None".

If you wish to allow access to PCD2 only to selected users, set "Other / Other" to "None" access, and then set "Execute" access for the desired users (by sign-on, group, or account, as with any other access list).

## 6) Error Messages

Error messages may be found:

- a. On the computer console B-display.
- b. In the system or job dayfiles.
- c. On the third message line on the Author Mode display.
- d. In group notes files "sysin" and "s0sysmsg".
- e. While executing a lesson.

This section documents many of these errors. For the most part, the messages are listed in alphabetic order. However, some messages are preceded by the issuing program name or some other variable word. If you cannot find a message you are looking for, check the list for a message beginning with the second word displayed.

The programs and lessons which generate each message are normally listed in parentheses immediately following the corresponding message.

## 60.1 Messages A

## ABORT (mas)

The ppu program MAS waits 10 milliseconds for the calling program to go into recall following a request to set the caller's RAE and FLE. This wait has timed out.

ACTION: Report via PSR.

## ADDR PE DN FLAG FNC (mrq)

See ADDRESS PE DN FLAG FUNCTION.

## ADDRESS OUT OF RANGE (epe)

The CN address passed to PP program EPE is incorrect.

ACTION: Report via PSR.

## ADDRESS PE DN FLAG FUNCTION (master,mastorn,mrq,mas)

A program has attempted to clear a bit in the EM flag register, and the controller has returned an abort status. The only condition which can cause this is a parity error in the EM address supplied.

ACTION: Call Customer Engineering.

## nnnnnnnnnn ALREADY EXISTS (pfin)

The current PLATO file on the input file cannot be created because the master file already has a PLATO file by this name.

ACTION: User error.

## xxxxxx ALREADY RUNNING (master,pnl)

An attempt was made to bring up PLATO job xxxxxx when it was already at a control point. Operational procedure.

ACTION: Correct PLATO load procedures.

## ALL EXECUTORS ACTIVE (plato)

This message is issued if you attempt to bring up more executors than allowed in your configuration file.

ACTION: Increase configuration file parameter "nexec" if more executors are desired or correct PLATO load procedures.

## ARGUMENT ERROR (tform,ppack,rcode,majjob,pf,ddpt)

The control statement is incorrect.

ACTION: See lesson "nosaid" or the PLATO Operations Guide for the correct format.

## ARGUMENT ERRORS (plato)

An executor has an incorrect executor identifier.

ACTION: Correct PLATO load procedures.



**ATTACH ERROR (aprint,dprint)**

Unable to open the student datafile or account management log file for which a print was requested.

**ACTION:** Report via PSR.

**AUTO-RECALL ERROR (epe)**

The program which called PPU program "ape" did not have the auto-recall bit set in the call.

**ACTION:** Report via PSR.

**60.2 Messages B**

**BAD ADDRESS (ddp)**

An incorrect CM address has been passed from DDPT to DDP.

**ACTION:** Report via PSR.

**BAD CHANNEL (ddp)**

The channel number to be tested given on the DDPT control statement is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**BAD CONTROL CARD (tprint)**

The format of the control card is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**BAD DEVICE TYPE (afcreat)**

Device type specified for new master file is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**BAD EXTENDED MEMORY POINTERS (mrq,mas,pio,pas)**

Certain extended memory info (i.e., rae, fie) for MASTOR and MASTORN are stored in word 63b of the job communication area. Word 64b contains the program name. Word 62b contains a checksum of these two words. PP programs read these words and verify the checksum before proceeding. If the checksum does not verify, this message is issued.

**ACTION:** This problem can be due to hardware errors. Call Customer Engineering.

**BAD FIELD LENGTH**

The field length passed to MASTOR by a PLATO job during initialization was incorrect.

**ACTION:** Report via PSR.

**BAD OVERLAY FILE** (mastor,mastorn,plato,framaf,conden)  
The format of the file on the deadstart tape which holds overlays for the program is incorrect.

**ACTION:** This problem could be due to a bad deadstart tape. Check the procedure used to build the current deadstart tape. If this does not resolved the problem, report via PSR.

**BAD PACK TYPE** (afalter,mfcreat)  
Master file type specified for master file is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**BAD PATTERN ADDRESS** (ddp)  
One of the parameters passed from DDPT to DDP is incorrect.

**ACTION:** Report via PSR.

#### 60.2.1 Messages BAD S

**BAD SECTOR** Eeeee Mmmmm Ttttt Sssss (pms)  
This message indicates that PMS detected an incorrect sector on a master file at the given location.

eeee = the EST ordinal (in octal) of the NDS device which contained the affected master file.

mmmm = the master file slot number (in octal) of the affected master file at the time of the error. The slot number may change with a later reload.

tttt = NDS logical track number of the error (in octal).

ssss = NDS logical sector number (in octal).

The first two bytes of each sector (64 CH words) of a disk file contain a pointer to the next sector and the logical sector length. These bytes are verified each time a disk sector is read into EM. If the bytes differ from expected values, this message is issued and the transfer is aborted.

**ACTION:** This condition should be corrected as soon as possible because it can cause all backup copies of this master file to be invalid. To correct this error:

- a. Note the numbers given in the error message.
- b. Convert the master file slot number "mmmm" to decimal and go to lesson "lcr" to determine the corresponding master file name.
- c. Go to the "verification options" in lesson "utility" and choose to run the "disk error check" option.
- d. Run the disk error check on the affected master file.

- e. When the bad sector is found, an error message will appear and the dayfile message will again appear. The name of the file containing the invalid sector will be shown.
- f. Rerun the disk error test on this file.
- g. When the error appears again, you will be given an option to rewrite the sector. Do so.
- h. After restoring the pointers, the file should be recovered from the most recent backup.

If you do not find the error using this procedure, run the disk error check on all master files.

**BAD SPACE COUNT (afcreat)**

Requested size in disk parts for the new master file is too large for the device type requested.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**6G.2.2 Messages BAT**

**BATCH JOB LIMIT (submit)**

The user name used for the requested submit already has the maximum number of submitted batch jobs allowed, as determined by the user number's validation file entry.

**ACTION:** Re-submit the job or increase the batch job limit for this user name.

**BINARY DIRECT BAD - nnnnn (plato)**

Validity checks on a PLATO binary failed while it was being loaded from disk due to one of the following reasons. "nnnnn" is the name of the TUTOR file whose binary was being loaded.

1. The name of the TUTOR file does not agree with the name stored in the binary file.
2. The name of the binary file and the name in the first word of the binary file directory read do not agree.
3. The PLATO file which should have contained the binary for this lesson does not have a file type of "binary" in its directory.
4. The binary file directory format is invalid.
5. The number of disk blocks to be read from the binary file is not reasonable.

The binary file is destroyed and the TUTOR file is condensed to form a new binary.

**ACTION:** Case 1 is usually caused by having two type "general" or "master" master files whose names have the same first four characters. Check all master files for this and correct. If this is not the case, report via PSR. For all other cases, report via PSR.

**BINARY READ ERROR - nnnnn (plato)**

While loading a binary from disk, a disk error was detected while attempting to read the binary file itself or the directory of one of the -use- files. The binary file is destroyed, and the source for the lesson is condensed to form a new binary. "nnnnn" is the name of the TUTOR file whose binary was being loaded.

**ACTION:** Call Customer Engineering.

**BINARY SUMCHECK ERROR (plato)**

When a binary is created, a sumcheck is performed and stored in the disk copy. When a binary is being loaded from disk, the sumcheck is verified. This message is issued when the comparison fails. This is probably caused by hardware problems. Immediately following this line in the dayfile, there will be a line which gives the name of the TUTOR and the binary file, and two lines which give the original and current sumchecks.

**ACTION:** Call Customer Engineering.

**BINARY TRUNCATED -- nnnnn (plato)**

PLATO software detected that the binary for lesson "nnnnn" had an incorrect length. The binary file is destroyed and the lesson is condensed to form a new binary.

**ACTION:** Report via PSR.

**BLOCK NOT FOUND (pf)**

When using an option which requires that a block name be given, a block of the specified name and type could not be found in the file specified.

**ACTION:** See lesson "nosalds" for the correct format for this control statement.

**BUFFERS OVERFLOW 12 BITS (master)**

Buffers in master have become too large for the software to handle.

**ACTION:** This can be caused by specifying too many master files ("ndsus") and CM-disk transfer buffers ("ncap") in the PLATO configuration file. Attempt to correct by changing the PLATO configuration file. If this is not possible, report via PSR.

**BUILT FOR ENHANCED NAM (pnl)**

Informative message. PNI was built to run with an enhanced, non-standard version of NAM.

ACTION: None.

BUILT FOR STANDARD NAM (pnl)  
 Informative message. PNI was built to run with the  
 standard release NAM.

ACTION: None.

### 60.3 Messages C

CANNOT CREATE FILE (pfin)  
 The current PLATO file on the input file cannot be  
 created because of a shortage of disk space or file name  
 table space on the master file.

ACTION: Disk space must be increased.

CANNOT DESTROY BINARY "b"; pack "p" (n) ("binary")  
 Lesson "binary" attached binary file "b" on master file  
 "p" and determined it was a candidate for deletion. A  
 "zreturn" of "n" was received when an attempt was made to  
 destroy the binary using the -sysfile- command. See lesson  
 "sysaids" for a description of this command and possible  
 "zreturn" values.

ACTION: This can be caused by unloading a binary master  
 file, but it should not occur with any frequency. If this  
 problem continues, report via PSR.

CANNOT RELOAD PLATO (mastor)  
 MASTOR has attempted to load PLATO, but the attempt has  
 failed for one of the following reasons.

1. Load submit file name on MASTOR K-display was cleared  
 or changed by operator action at computer console.
2. Load submit file cannot be submitted by MASTOR.
3. MASTOR has waited 90 seconds for PLATO to notify it  
 that the PLATO system is successfully running. PLATO  
 is probably aborting or hanging during initialization.

ACTION:

1. Re-enter the correct submit file name on the MASTOR  
 K-display.
2. Attempt to correct the cause of the failure. The most  
 likely cause for this error is that the user name  
 specified by the "subun" entry in the PLATO configuration  
 file is not validated to submit system service class  
 jobs. Another cause could be that the submit file  
 specified is not under the user name specified by "subun".
3. Determine what is causing PLATO to abort.

CANNOT RETURN TO MASTOR (mq)

The PP program MRQ switches to other control points for certain job management and information functions. It then switches back to MASTOR's control point when done. While MRQ was at the other control point, MASTOR was aborted, or MTR refused the switch request.

ACTION: Report via PSR.

#### CARDFILES ATTACHED ("transfer")

This message may be seen during courseware installation.

ACTION: See the "Courseware Installation" section of the PLATO Operations Guide for more information.

#### CENTRAL MICRO PLATO text (conden)

There is an error on one of the "s0cmpN" files used by the CMP feature, where N is a number 0 or greater. The "text" is additional text describing the error.

ACTION: For the case where "text" = "PLATO OUTPUT FILE OVERFLOW", the code produced by CMP will not fit into an 18-part file. The user must make his lesson smaller.

For all other cases:

- a. Review the CMP initialization procedures and make sure you have created enough "s0cmpN" files. Make sure the files are the proper length and type and that appropriate codewords exist.
- b. Run "utility" file directory and disk error tests on all "s0cmpN" files.
- c. If unable to resolve, report via PSR.

#### 60.3.1 Messages CH

##### CHECK DATE/TIME (plato)

This message is issued by lesson "stats1" during system initialization if the last time the PLATO system was brought up was later than the current time or if the last time was more than 25 hours before the current time. This check is to insure that the operator entered the correct the correct date and time at Jeadstart time.

ACTION: Check the date and time. If they are incorrect, drop PLATO immediately and correct them, then reload the application. It is very important that the date and time be correct as many user programs rely on them.

DO NOT RESET THE OPERATING SYSTEM DATE AND TIME WHILE PLATO IS RUNNING.

##### CHK-SUM ERR IN PPT TABLES (plato)

This message is issued if an error is detected while reading the terminal definitions database from file "s0pcon". This database is checksummed to verify its integrity.

**ACTION:** Recover file "s0pcom" from the most recent backup. If this does not resolve the problem, report via PSR.

**CIU RESERVED (pio)**

Informational message seen only at initialization to indicate that the CIU has been found in the EST, and that both input and output channels have been reserved.

**ACTION:** None.

**CLOCK RESET (mrq)**

This message will only be seen on multi-mainframe systems. The copies of MRQ on each mainframe keep the millisecond clock of each mainframe in synchronization. This message is issued on a mainframe when a negative elapsed time or a positive elapsed time greater than 25 milliseconds was generated during a clock update.

**ACTION:** None.

**CM ADDRESS OUT OF RANGE (mfutil)**

A request was made to a PP program, but the request address was not within the CPU program's field length.

**ACTION:** Report via PSR.

**CODE WORD ERROR (pf)**

The requested file cannot be opened because the job does not have the correct codeword for the file.

**ACTION:** Use the PCJDE control statement before the PF statement to set the correct codewords. See lesson "nosaid" for the correct format for this control statement.

**CONDENSE ERRORS IN lesson (plato)**

System lesson "lesson" has condense errors. Unless the software has actually been changed by someone in the PLATO support group, this usually indicates a hardware error.

**ACTION:** If the errors are in a lesson such as "plato" or "edit", normal users should be backed out until the problem is resolved. Condense the lesson and try to determine why there are condense errors. Depending on the cause of the errors, call Customer Engineering or report via PSR.

**CONDENSOR/OVERLAY MISMATCH (conden)**

This occurs if an attempt was made to load a version of the condensor which does not match the condensor overlays currently in EM.

**ACTION:** If there is a condensor already running and you are just trying to load an additional one, then stop and do nothing. If no condensers are running, then report via PSR.

**CONTROL CARD ERROR (mfutil)**

An attempt was made to use one of the master file utility programs, but an unrecognized parameter was used on the control statement.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**COURSEWARE ACCESS LIMIT NOT FOUND ("s0init")**

This message is issued when the current system cannot be found in the courseware access limit table (located in file "0caco", block "cacw"). The access limit is set to one and only one person may be executing a published lesson at any time.

**ACTION:** Contact PLATO Courseware Delivery.

**60.4 Messages D****DATA TRUNCATED (pf)**

When using a PF "write" option, the data in the local file overflows the PLATO file.

**ACTION:** lengthen PLATO file.

**DATE NOT AVAILABLE (mastorn)**

This message will be seen only on multi-mainframe systems. During initialization of MASTORN, it requests the date and time from MASTOR on mainframe zero so the mainframes will have exactly the same time and date. If there is any error during this process, this message is issued.

**ACTION:** Check the status of mainframe zero for more information.

**60.4.1 Messages DDP****DDP CHAN ERR C=cc E=tn (pio,ddp)**

This problem can be caused by a variety of channel problems detected while attempting to read or write EM through the DDP or low-speed port. The DDP port, the channel or EM itself could have a problem. The parameters, if present, have the following meanings:

cc = channel number

t = type ("R" for read; "W" for write)

n = error number - when combined with the type, it determines what function is failing

x1 Following a channel function, the PP waits for the channel to become inactive. This never happened.

x2 After functioning and then activating the channel, it was unexpectedly full. This



indicates that the controller is not accepting the function.

- R3 The PP issues a function to read from EM, outputs the EM transfer address, then waits for the channel to become full, indicating that the data from the controller is ready to be read in. The channel failed to become full.
- R4 The PP loads the number of words to be transferred into the accumulator and begins the transfer with an "lax" or "oax" instruction. When the transfer completes successfully, the controller disconnects the channel, which returns control to the PP with the accumulator set to zero. In this case, the channel was disconnected and control returned to the PP before the transfer word count in the accumulator reached zero.
- R5 Following a successful transfer from EM, the controller puts a status word on the channel, setting it active and full. The status word did not appear.
- W1 See the explanation for "R1".
- W2 See explanation for "R4".
- W3 See the explanation for "R1". This error indicates that the function to get the status word following a write to EM was not accepted by the controller.
- W4 Following a write to EM, the PP sends a request for the status word, waits for the channel to become inactive (indicating that the controller accepted the function), activates the channel and waits for the channel to become full (indicating that the status word is available). The channel failed to become full.
- W5 An abort status was returned following a write to EM.

ACTION: Call Customer Engineering.

DDP CHANNEL ERROR (mq)  
See "DDP CHAN ERR".

DDP RESERVED (pio)  
Informational message seen only at initialization to indicate that the DDP or low-speed port assigned to PIO has been found and reserved.

ACTION: None.

xxxx = DDP PORT STATUS (pas)

An error was detected while attempting a "master clear port" function through the DDP. "xxxx" is the DDP port status word. This indicates a hardware problem with the DDP, the channel or EM.

ACTION: Call Customer Engineering.

60.4.2 Messages DE

DELETING - nnnnnnnnnn (pfdest)

Informative message. "nnnnnnnnnn" is the name of the PLATO file now being deleted from the master file.

ACTION: None.

DISK ERROR (eprint,dprint)

A disk error was detected while trying to print the account file management logs or a student datafile.

ACTION: Call Customer Engineering.

DISK REQ POST ERROR (mastor)

A PLATO disk request was rejected by MASTOR for one of the following reasons:

1. The master file slot number on which the request was made was illegal.
2. Too many disk requests have been made on at least one master file slot and a queue has overflowed.

ACTION: Case 1 is normal when a PDPRT control statement is used. If this is not the case, report via PSR.

DISK STACK DJR (pms)

An internal check for a valid parameter passed from MASTOR to PMS during initialization has failed. The disk request area address in MASTOR's FL is out of range.

ACTION: Report via PSR.

DROPPED AT nnnn. (mrq)

This is an informational message used for debugging.

ACTION: None.

DROPPING PLATO (pni)

Informative message only. PNI is dropping PLATO.

ACTION: None.

DUMPING - nnnnnnnnnn (pfout,ecdump)

Informative message. "nnnnnnnnnn" is the name of the PLATO file now being dumped from the master file (PFOUT) or the beginning EM address of the block being dumped (ECDUMP).

ACTION: None.

DUPLICATE LOGIC (plato,framat,conden)

A program attempted to identify itself to MASTJR as part of the PLATO system, but that particular section of the system was already active.

ACTION: Correct PLATO load procedures.

60.5 Messages E

EARLY END OF FILE IN COM. BUFFER (backone)

The communications buffer contains information about fewer files than are actually on the master file being dumped. This can be caused by not inhibiting PLATO file options while master files are being dumped.

ACTION: Purge file CJMBUF under user name PLATDMF, re-define this file and begin the dumps again with PLATO file options inhibited.

ECS ABORT (pms,plo)

A parity error was detected during an EM write. There could be a hardware problem with the DDP or low-speed port, channel or EM itself.

ACTION: Call Customer Engineering.

ECS ERROR (aprint,ecdmp,dprint,tprint)

An ECS parity error was detected during an EM read or write. The EM, the controller or the DDP port or low-speed port may have a problem. ECUMP also reports the beginning address of the 10000-word block in which the error occurred.

ACTION: Call Customer Engineering.

ECS PARITY ERROR (mrq,pms,mas,plo)

A parity error was detected during an EM read. The EM, the controller or the DDP port or low-speed port may have a problem. The PP program PMS also gives the absolute address of the beginning of the transfer being performed. The transfer length is always 64 CH words for PMS.

ACTION: Call Customer Engineering.

60.5.1 Messages EM

++++ (DDP) EM PARITY ERROR +++ (ddp)
+
+ CHANNEL CC
+ ABS FWA aaaaaaaa
+ LENGTH 1111
+ PATTERN xxxxxxxxxxxxxxxxxxxxxx
+
+++++

A parity error was detected when DDP attempted to read data

from EM through the DDP / low-speed port. See "EM VERIFY ERROR" for the meaning of the symbols in this message.

ACTION: Contact Customer Engineering.

++++ (DDP) EM READ ABORT +++++ (ddp)

+  
 + CHANNEL cc  
 + ABS FWA aaaaaaaa  
 + LENGTH llll  
 + PATTERN xxxxxxxxxxxxxxxxxxxxxx  
 +

+++++

An EM transfer abort signal was detected when DDP attempted to read data from EM through the DDP / low-speed port. See "EM VERIFY ERROR" for the meaning of the symbols in this message.

ACTION: Contact Customer Engineering.

EM REQUEST FAILURE (conden)

The condensor was unable to obtain an EM buffer from PLATO in which to put the binary of the current lesson being condensed. The condense request is aborted and the user sees an error message. This message may occur because of a temporary shortage of EM, or it could mean a more serious problem if it continues for a few minutes or more.

ACTION: If the message occurs frequently, refer to the PLATO Configuration Handbook for information on how to attempt to resolve the memory shortage.

++++ (DDP) EM VERIFY ERROR +++ (ddp)

+  
 + CHANNEL cc  
 + ABS FWA aaaaaaaaa  
 + LENGTH llll  
 + OFFSET oooo  
 + PATTERN xxxxxxxxxxxxxxxxxxxxxx  
 + READ AS yyyyyyyyyyyyyyyyyyyy  
 +

+++++

DDP writes patterned data to EM through the DDP or low-speed port, reads it back and verifies that it received the same data it wrote. When this message is issued to the dayfile, the data verification failed. The meaning of the various fields in the message follows.

- cc = channel number of the DDP / low-speed port being tested when the error was detected
- aaaaaaaa = 24-bit EM address where first word of the data was being written/read when the error was detected
- llll = length of the data block being written/read when the error was detected
- oooo = offset into the data block of the failing word
- xx.xx = 60-bit pattern written to/read from the failing

**ACTION:** The error in the "course access" common is normal if the system does not have published courseware installed. For all other errors, check for missing master files and correct PROC/MFNX to attach all required master files. If this does not resolve the problem, recover the appropriate file from the most recent backup.

**ERR IN LOADING MICRO-TUTOR BINARY (plato)**

This message is issued if an error is detected while reading a Micro PLATO Interpreter from disk to load it into a terminal. The user receives an error message if this happens, but Micro PLATO remains available to users of other Micro PLATO levels.

**ACTION:** Check for missing master files and correct PROC/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

**ERR IN LOADING PPT TABLES (plato)**

This message is issued if an error is detected while reading the terminal definitions database from file "s0pcom".

**ACTION:** Check for missing master files and correct PROC/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

**ERRJR IN PARAMETERS (backlib,backlist,backmod,copyf,copypf)**  
An incorrect parameter or value was entered.

**ACTION:** See the PLATO Operations Guide for the correct format of this control statement.

**ERRJR IN READING A BLOCK (tprint)**

A disk error was detected while trying to read a block from a PLATO file.

**ACTION:** Call Customer Engineering.

**ERRJK IN SPELL DATASET (plato)**

This message is issued by lesson "s0init" during the PLATO system initialization if the database associated with TERM-spell cannot be attached.

**ACTION:** Check for missing master files and correct PROC/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

**ERROR IN WRITING (backcopy)**  
DATASET = ffff  
ERROR STATUS = ss  
BEGINNING RECORD = dd  
ENDING RECORD = ee

- b. For each error you find, go to lesson "utility" and rerun the same test on the same file or master file. This should give you a more detailed description of the error. The test to run is determined by the test name as recorded in "sdulog".
  - mfdfvy - master file directory integrity test
  - fdfvy - file directory integrity test
  - devfy - disk error test
- c. Using this detailed description as well as any additional information provided in the HELP section, attempt to repair the file.
- d. If unable to make repairs, obtain a backup. If no suitable backup exists, it may be necessary to destroy the file without a backup.
- e. If the problem appears to be the result of an unreported and unresolved software problem, then report via P&R.
- f. Initialize "sdulog" to make room for more errors.

ERROR WHILE READING TERMINAL RESIDENT (plato)

FILE f - r

An error was detected while reading the terminal resident for "r" from file "f" during application initialization.

ACTION: Check for missing master files and correct PROC/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

ERROR WITH FILE xxx ("accountp")

An error has been detected in file "xxx" when a user attempted to run the "last editor" option in his account.

ACTION: Run account cleanups on this account.

60.5.4 Messages ERROR (number)

ERROR xx INFO = xinfo

This error message may appear immediately after the

UNIT uuu afname aftype

master file load message if an error occurs. "xx" is the error number shown below and "xinfo" is extra information, if applicable for the given error number. This message may be seen during initialization of the PLATO application or when attempting to load a master file through lesson "ldr".

Error numbers:

- 1 MASTOR disk request failed. "xinfo" is the reply code from MASTOR in octal. This reply code is the disk error code returned by PMS. See "disk error"

more files than are actually on the master file being dumped. This can be caused by not inhibiting PLATO file options while master files are being dumped.

**ACTION:** Purge file CD43UF under user name PLATDMF, re-define this file and begin the dumps again with PLATO file options inhibited.

## 60.6 Messages F

**FATAL I/O ERROR** nnnn (mfcreat)

MFCREAT has detected a fatal disk error while allocating and zeroing disk space for a new master file. "nnnn" is the error code as returned by the operating system.

**ACTION:** The disk pack probably needs to be flawed.

**FILE ALREADY BUSY** (mfadd)

Master file is attached in write mode at another control point so it cannot be attached to MASTOR.

**ACTION:** Wait a few minutes and try again.

**FILE LIMIT** (pfin)

An attempt was made to create a PLATO file on a master file, but the master file was at its file limit (either 1080 or 2160 files, depending on total master file size).

**ACTION:** No more files may be added to this master file. There is no corrective action.

**FILE NOT IN WRITE MODE** (mfalter,pfin,pfdest)

Master file not attached in write mode.

**ACTION:** Re-attach master file in write mode and re-try.

**FILE NOT ON MASS STORAGE** (mfaod)

File which was supposed to be attached to MASTOR is not resident on mass storage.

**ACTION:** Move the desired file to a mass-storage device and re-try.

**FLAG INTERLOCK** (mrq)

MRQ has been trying to reserve shared EM tables for one second. This usually indicates a hardware problem which has resulted in the flag register indicating it is reserved at all times (170-700 series machines only) or some type of software problem.

**ACTION:** Call Customer Engineering. If no problem is found or if your system is not on a 170-700 series mainframe, report via PSR.

**FLX REQ xxxx / ACT yyyy** (mastor)

The EM field length reserved is different from the field length requested by the "flx" configuration file entry.

- in lesson "sysalds" for a list of these error codes.
- 2 Invalid masterfile type. "xinfo" is not used.
  - 3 Invalid number of files on master file. "xinfo" is the file/space information word from the master file directory.
  - 4 Invalid master file directory length. "xinfo" is the size in decimal from the master file directory.

**ACTION:** Either the file to be loaded was not a master file or the master file directory has been damaged. Make sure the correct file is being loaded. If it is, recover the master file from a backup. If the problem continues, report via PSR.

#### 60.5.5 Messages EX

##### EXECUTION ERROR IN LESSON \*nnnnnnnn\* (plato)

This message is issued by lesson "execerr" when a TUTOR execution error is detected in system lesson "nnnnnnnn". The following additional lines are issued if the error is in one of the special system library lessons. This is done because no other automatic error reporting scheme can work without these lessons. This method will avoid infinite loops between the lesson with the error and "execerr".

PJINTERS - xxxxxxxxxxxxxxxxxxxx

- 1/user lesson had data collection turned on
- 1/user lesson had -route error- unit
- 1/automatic error recording
- 15/unused
- 9/argument number
- 6/contingency
- 12/command number within unit
- 15/execution error number

COMMAND - xxxxxxxx

UNIT - xxxxxxxx

X1 - xxxxxxxxxxxxxxxxxxxx

X2 - xxxxxxxxxxxxxxxxxxxx

**ACTION:** A copy of the information from the display in lesson "execerr" or the lines issued to the system dayfile must be reported via PSR. A copy of the information displayed by lesson "execerr" is also automatically written into notes-file "sysin".

##### EXECUTOR DEAD (plato)

This message may only be seen on multi-executor systems. Each executor maintains a real-time clock in EA to indicate to other executors that it is still running. One of the clocks is at least sixty seconds behind, indicating that that executor is no longer running.

**ACTION:** Determine what is wrong by checking the system dayfile and/or 3-display for other messages.

##### EXTRA WORDS IN COMM. BUFFER (backone)

The communications buffer contains information about



more files than are actually on the master file being dumped. This can be caused by not inhibiting PLATO file options while master files are being dumped.

**ACTION:** Purge file CD43UF under user name PLATOMF, re-define this file and begin the dumps again with PLATO file options inhibited.

## 60.6 Messages F

### FATAL I/O ERROR nnnn (mfcreat)

MFCREAT has detected a fatal disk error while allocating and zeroing disk space for a new master file. "nnnn" is the error code as returned by the operating system.

**ACTION:** The disk pack probably needs to be flawed.

### FILE ALREADY BUSY (mfadd)

Master file is attached in write mode at another control point so it cannot be attached to MASTOR.

**ACTION:** Wait a few minutes and try again.

### FILE LIMIT (pfin)

An attempt was made to create a PLATO file on a master file, but the master file was at its file limit (either 1080 or 2160 files, depending on total master file size).

**ACTION:** No more files may be added to this master file. There is no corrective action.

### FILE NOT IN WRITE MODE (mfalter,pfin,pfdest)

Master file not attached in write mode.

**ACTION:** Re-attach master file in write mode and re-try.

### FILE NOT ON MASS STORAGE (mfaod)

File which was supposed to be attached to MASTOR is not resident on mass storage.

**ACTION:** Move the desired file to a mass-storage device and re-try.

### FLAG INTERLOCK (mrq)

MRQ has been trying to reserve shared EM tables for one second. This usually indicates a hardware problem which has resulted in the flag register indicating it is reserved at all times (170-700 series machines only) or some type of software problem.

**ACTION:** Call Customer Engineering. If no problem is found or if your system is not on a 170-700 series mainframe, report via PSR.

### FLX REQ xxxx / ACT yyyy (mastor)

The EM field length reserved is different from the field length requested by the "fix" configuration file entry.

The requested EM field length was "xxxx" and the actual field length is "yyyy". Both "xxxx" and "yyyy" are in terms of 10000 words. If the requested EM field length and the actual field length reserved are the same, this dayfile message will not be issued. These values may differ due to another job having part of EM reserved, which reduces the amount of EM available for PLATO below the requested EM field length. This message is a warning that the EM being used by PLATO is less than that requested. The PLATO application may still be able to run.

ACTION: None.

FNT NOT FOUND (mfadd)

The file which is supposed to be attached to MASTOR cannot be found at the control point.

ACTION: Correct the job so that the file to be attached to MASTOR is attached before using the MFADD control statement.

FOR PLATO CRASH DUMPS TO FINISH. (copypd)

See WAIT. FOR PLATO CRASH DUMPS TO FINISH.

FORMAT DROPPED (format)

This message is only used on systems which are running multi-formatters. One of the formatters has detected a fatal error condition and has aborted, or it has exited normally through a request by FRAMAT.

ACTION: Check the system dayfile and 3-display for other error messages.

FRAME OVERFLOW (frawat)

There is too much formatted output to fit into a frame of output. This indicates that the FRAMER is falling behind, probably due to system overload. The excess output is discarded.

ACTION: If this problem does not appear to be due to a system overload, report via PSR.

## 60.7 Messages G

---GETUNIT ABORT--- (plsto)

-ACCOUNT aaaaaaa

-LESSON 1111111111

---GETUNIT ABORT---

A parity error has been detected while attempting to read the EM copy of lesson "1111111111", account "aaaaaaa". The user gets an error message, and must restart his lesson. The lesson which got the error is marked so it may not be used, moved, or deleted. This is done to prevent another user from using the bad area in EM. The name of the lesson is changed to "//accerr//". This may be seen in lesson "system1", "ECS Usage" option.

ACTION: Continue to monitor the system for further errors

and call Customer Engineering.

**6D TO OVERWRITE DUMP** (all dump procedures)

The required dump file for this job already exists. This message will be preceded by a dayfile message indicating which dump file already exists.

**ACTION:** The operator should enter "GD" for this job, if the previous dump file is not wanted or when the previous dump file is safely copied to a tape.

**6D.9 Messages I**

**ILLEGAL ARGUMENT** (submta, pf)

An argument specified on the control statement was not recognized.

**ACTION:** See lesson "nosalds" for the correct format of this control statement.

**ILLEGAL BLOCK TYPE** (pf)

When using an option which requires a block name and type be specified, an invalid block type was used.

**ACTION:** See lesson "nosalds" for the correct format of this control statement.

**ILLEGAL ECS INSTRUCTION AT xxxxxx** (epe)

**INSTRUCTION - #####**

This message is issued when PP program EPE has been called to process an EM parity error and it detects that the machine instruction involved is illegal. This is probably a hardware error.

**ACTION:** Call Customer Engineering.

**ILLEGAL FILE TYPE** (tprint)

The requested file is not one which this program can print. This program is used to print the contents of TUTDR, dataset and nameset files.

**ACTION:** See the PLATO Operations Guide for the correct print program for the file type to be printed.

**ILLEGAL MAIN-FRAME** (mastorn)

This message is issued during initialization of MASTORN if the mainframe ordinal specified by the "mford" PLATO configuration file entry is greater than the number of mainframes specified by the "naf" entry or if the NOS file "lconfi" exists on the mainframe under user index 3777730 and the "mford" entry is set to zero, indicating that this mainframe is the primary mainframe.

**ACTION:** If this mainframe is to be the primary mainframe, purge NOS file "lconfi". If this mainframe is to be a secondary mainframe, set "mford" to a non-zero number less than the "naf" entry or increase the "naf" entry.

**ILLEGAL PARAMETER (submit)**

An invalid parameter has been specified on the SUBMIT control statement.

**ACTION:** See lesson "nosaid" for the correct format for this control statement.

**ILLEGAL REQUEST (ppack, pcode)**

1. The request from PPACK to set a master file name for future file references was refused by MASTOR.
2. The request from PCODE to set a codeword for future file references was refused by MASTOR.

**ACTION:** Report via PSR.

**ILLEGAL USER ACCESS (pf, epe)**

1. A non-system origin job tried to use the "bin" option of PF.
2. A non-system origin job attempted to call PP program EPE.

**ACTION:** The first case is due to user error. The job should be re-submitted under a user name which has system origin privileges. The second case should be reported via PSR.

**ILLEGAL USER CARD (xasjob)**

The user statement supplied by the submit file was not in the correct format, or had an illegal user name.

**ACTION:** Correct the user statement and rerun the job.

**ILLEGAL USER NUM (submit)**

The user name in a request passed to MASTOR is incorrect. This is probably a user error, due to no USER control statement preceding the SUBMIT control statement.

**ACTION:** Correct and rerun the job.

**60.9.1 Messages IM****IMPROPER REQUEST (mrq, mas, submit, ddpt)**

An incorrect request has been made to a program. This may be one of the following:

1. Computing the absolute address of an address relative to the caller's RA has resulted in an illegal address.
2. MRQ was called by a job which was not system origin.
2. MAS was not called under one of the following conditions:
  - a. calling job is system origin
  - b. user name of calling job has system-origin privileges
  - c. operating system is running in "DEBUG" mode
  - d. calling program has SSU= special entry point
4. An incorrect request code has been passed to MAS.
5. A request from SUBMIT or DDPT to MASTOR contains incorrect information.



This is probably a hardware problem.

ACTION: Call Customer Engineering.

INVALID SUBMIT USER NAME (mastor)

The user name or family in the PLATO configuration file or in a request to submit a job through MASTER cannot be found.

ACTION: Correct the PLATO configuration file or user request or use MODVAL to create the proper user name.

60.10 Messages J

JOB CARD ERROR (masjob)

The job statement in the submit file given as an argument on the MASJOB control statement is incorrect.

ACTION: Correct the submit file and re-submit the job.

JOB HAS NO EM (jdp)

PP program DDP has been called by a job which did not request EM prior to the call.

ACTION: Report via PSR.

JOB NOT IN ECSTAB (mastorn)

During initialization, MASTORN searches MASTER's table of jobs which are using EM for its own job name to obtain its RAE and FLE relative to MASTER's RAE and FLE. The correct job name could not be found in the table.

ACTION: Report via PSR.

JOB TABLE FULL (mastor,mastorn)

The job name table, with which MASTER tracks batch jobs submitted through the PLATO system, is full. Any user who wants to submit another job must wait.

ACTION: If this happens frequently, increase the "njob" PLATO configuration file entry.

60.11 Messages K

KEY BUFFER OVERFLOW xxxx (pio)

PIO reads all pending keys from the CIU and stores them in a buffer in its memory. This buffer has overflowed "xxxx" times since the last PLATO reload. Informational message.

ACTION: Normally, none. If the number displayed is very high, this can indicate a problem with the CIU channel, a site controller or the CIU, and you should call Customer Engineering or look for a communications network problem.

KEY DATA OUT OF SEQ (pio)

The input keys are received by PIO as two separate data

words, one is the station number and the other is the value of the key being sent. This message is issued when PIO expected a station number word but got a key data word instead. This is probably a CIU, site controller or channel problem.

ACTION: Call Customer Engineering.

KEY IGNORED STATION xxxx3 (pio)

In this message, "xxxx" is the station number in octal. There is a buffer in EM which contains the last four keys for each station. If an additional key comes in before PLATO has processed one or more of the four previous keys, there is no place in the buffer to store the new key. It is consequently ignored and this message is issued to the B-display. Since PLATO should process keys faster than humans can type, this message should only occur when a station is getting line errors and flooding the system with keys. This message is usually of no concern, as communications people should have better means of determining when a station is generating errors.

ACTION: Normally, none. If the station number in this message is changing rapidly, there may be a CIU, site controller or channel problem and you should call Customer Engineering or look for a communications network problem.

60.12 Messages L

LAST PATTERN ADDR UJR (ddp)

One of the parameters passed from DDPT to DDP is incorrect.

ACTION: Report via PSR.

LESSON ENTRY ERROR nn - xxxxxxxxx (plato)

A fatal lesson entry error "nn" has occurred in lesson "xxxxxxx". The possible values of "nn" are shown below and discussed more fully in lesson "ajds" under "zieserr". Note that other values of "zieserr" exist, but should not appear in this message.

- 0 unknown error
- 1 condensor not available
- 2 no such lesson
- 3 lesson source is too long
- 4 no computer memory available
- 6 disk error
- 7 unit too long
- 10 no room in memory for common
- 11 common not found
- 12 not enough common blocks
- 13 lesson tables full (system error)
- 14 codeword mismatch for common
- 15 tag too long
- 16 lesson binary too long
- 17 not a tutor lesson
- 18 lesson temporarily unavailable

19 memory allocation exceeded  
 20 lesson directory error (system error)  
 21 next physical unit missing (system error)  
 23 -jumpout- codeword error  
 24 common in EM has different length  
 25 -jumpout- to wrong router  
 27 condensor failure (system error)  
 28 published courseware access error  
 30 obsolete lesson  
 31 too many -use- blocks  
 32 invalid processor lesson  
 43 unit name table full  
 44 published courseware access limit

**ACTION:** Attempt to find and correct the cause of the problem. If you are unable to do so, or if the value of "nn" is not in the above list, report via PSR.

**LESSON PLATE FATAL CONDERR (plato)**

While condensing lesson "plato", a fatal condense error was found (ie, bad -use- file, etc). If this is during system initialization, this will prevent everyone from signing in, except station 0-0 (the Cyber console).

**ACTION:** Attempt to find and correct the cause of the problem. If you are unable to do so, report via PSR.

**LESSON TABLE FULL (plato)**

A user was unable to get storage for a lesson because the EM lesson table was full.

**ACTION:** Increase the value of the "lesns" configuration file keyword.

**LISTING BLOCKS ONLY (tprint)**

This message is placed in the job dayfile when a print is made of a TUTOR or CODE file which had the special option to print only the listing blocks in the file set in the file directory. This is only an informational message.

**ACTION:** None.

**LOAD FILE FORMAT ERROR (pni)**

There is an error in a PNI terminal load file.

**ACTION:** Rebuild the PNI load files using lesson "s0pnlif", then drop and reload PNI. If the error persists, report via PSR.

**LOADING - nnnnnnnnn (pfin)**

Informative message. Lesson "nnnnnnnn" is being loaded onto the master file.

**ACTION:** None.

**LOADING PLATJ (pni)**

PNI is loading PLATJ. Informative B-display message.



ACTION: None.

### 60.13 Messages M

#### MAIN-FRAME DEAD (submit)

The mainframe specified in the submit request is no longer active.

ACTION: Check other mainframe for more information. This also could be a user error in specifying wrong mainframe.

#### MASTER FILE SUMCHECK ERROR - nnnnnnn (plato)

This message will only be seen on systems which have master file sumchecking enabled by the "pdchk" configuration file entry. When a master file directory is about to be changed (creating a PLATO file, checkpointing the master file directory, etc.), a sumcheck is performed and compared to the previous one. This message is issued, and PLATO aborts with a CPU ERRJR EXIT if the comparison fails, indicating that the EM copy of the master file directory has been illegally changed. When this happens, certain hardware registers in the PLATO dump contain useful data:

- (x1) = master file ordinal
- (x2) = current sumcheck
- (x3) = previous sumcheck

This problem is usually caused by a hardware error.

ACTION: Call Customer Engineering.

#### MASTOR DRDP (mastor,mastorn)

This message is issued at the end of a successful shutdown of the PLATO system following an operator "K.STOP" request to MASTOR.

ACTION: None.

#### MASTOR DUMP FILE ALREADY EXISTS (mastor,mastorn)

This is an informative dayfile message issued when the MASTOR dump procedure detects that the required dump file already exists. After this message is issued, a flashing WAIT control statement appears on the B-display along with the message "GO TO OVERWRITE DUMP".

ACTION: See "GO TO OVERWRITE DUMP".

#### MASTOR MF=0 DEAD (mastorn)

This message will only be seen on multi-mainframe systems. It is issued on the secondary mainframes when MASTOR on the primary mainframe aborts or hangs.

ACTION: Check the status of the primary mainframe for more information.

#### MASTOR NOT ACTIVE (mas,mfu,mrqp,plc)

This message is issued during initialization of the PP program MRQ if one of the following conditions exists:

1. A parity error is detected while reading pointer tables or the MASTOR millisecond clock in EM.
2. The name of the job this copy of MRQ is assigned to does not begin with the characters "MAS".
3. MRQ waits for MASTOR's millisecond clock to update as a check for activity. This check has timed out.

This message is also issued by other PP programs if the MASTOR control point cannot be found or is inactive.

ACTION: Check for other messages for more information.

MASTOR NOT LOGIC (mastorn)

During initialization, MASTORN attempted to identify itself to the MASTOR on the primary mainframe, but the request was rejected.

ACTION: Report via PSR.

MASTOR REQ BUFF ERROR (plato)

A fatal error condition was detected in the communications buffer between PLATJ and MASTOR.

ACTION: Report via PSR.

MASTOR REQUEST = 0 (mas)

The request passed to MASTOR by MAS has been zeroed.

ACTION: Report via PSR.

MASTOR SETR CLEAR FAILED (unknown origin)

ACTION: Report via PSR.

MASTOR SETR REJECT (conden)

During initialization of the condensor, a communications buffer between MASTOR and the condensor is set up in EM. The request to do this was rejected by MASTOR.

ACTION: Report via PSR.

MASTOR TIMEOUT (mas)

After passing a request to MASTOR, MAS will wait for eight seconds for a reply that the request is complete or instructions from MASTOR for an action to complete the original request. This time limit has been exceeded.

ACTION: Check for more messages for more information.

60.13.1 Messages MAX

MAX TRANSFER LT MIN (dup)

One of the parameters "MX" or "MN" specified on the DJPT control statement is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**MAX TRANSFER TOO LARGE (ddp)**

The "MX" parameter specified on the DDPT control statement is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**MF INFO TABLE DDR (pms)**

An internal check for a valid parameter passed from MASTOR to PHS during initialization has failed. The master file information table address in the MASTOR FL is invalid.

**ACTION:** Report via PSR.

**MFORO .NE. 0 (mastor)**

The "local PLATO configuration file" ("iconfig") does not exist under user index 377773b on this mainframe, which indicates that this should be the primary mainframe, but the "mford" entry in the PLATO configuration file is set to a number other than zero.

**ACTION:** If this mainframe is to be the primary mainframe, change the "mford" configuration file entry to zero. If this is to be a secondary mainframe, create the NDS file "iconfig" and add an "mford" entry with a non-zero value.

**\*\*\*\*\* - MF DUMPED MORE THAN ONCE (backone)**

During the file backup process, master file "\*\*\*\*\*" was dumped two or more times.

**ACTION:** Change PRDC/MFDX so that the master file is only listed one time.

**MISSING MICROTUTOR BINARY...nnnnnn (plato)**

This message is issued by PLATO during initialization if one of the Micro PLATO interpreter binaries cannot be loaded from disk in order to initialize entry point tables. "nnnnnn" is the missing file name. A flag is set which disables Micro PLATO for all users of the system.

**ACTION:** Check for missing master files and correct PRDC/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

**MISSING SOME REQUIRED PACKS (plato)**

This message is issued by lesson "s0init" when the PLATO system is being initialized if at least one of the master files listed in the required packs list was not loaded. All account file operations are turned off so duplicate file names cannot be created. Either the contents of PRDC/MFNX does not agree with the required master file list or a master file could not be attached.

ACTION: Check the contents of PROC/MFNX against the required master file list and correct. If this does not resolve the problem, check the PLATD dayfile for more information on master files not being attached.

MISSING SUBOVERLAY--xxxxxxx (plato,framat,conden)  
An overlay of name "xxxxxxx" is missing from the load file.

ACTION: Report via PSR.

MODIFY FILE NAME MISMATCH (pf)  
This message will only be seen when using the "modify" option of PF. The message is issued when the first word of the local file does not agree with the name of the PLATD file. Probably user error.

ACTION: Correct and rerun the job.

### 60.13.2 Messages MR

MRQ CANNOT GET MASTER INTLK (mrq)  
This message is used only on 170-800 series mainframes. This message means that MRQ cannot reserve the software interlock for MASTER because the channel flag on the maintenance channel is reserved or another program has the interlock word in MASTER's CM reserved.

ACTION: Take an express deadstart dump and report via PSR.

MUST BE SYSTEM ORIGIN (mfutil)  
One of the master file utility programs was called from a non-system-origin job.

ACTION: Rerun the job under a user name which has system origin privileges.

MXR BAD OPTION (mxx)  
An invalid request has been issued to PP program MXR.

ACTION: Report via PSR.

MXR BAD PP WORD (mxx)  
An error occurred during a pause for storage move.

ACTION: Drop the PLATD load job and initiate PLATD again. If the problem persists, report via PSR.

MXR ECS EQUIP NOT AVAILABLE (mxx)  
The "DE" or "DP" equipment could not be found in the NDS equipment status table (EST).

ACTION: Correct the EQPDECK.

MXR ECS TRACKS NOT SEQ (mxx)  
One of the following conditions has occurred:

1. MXX is unable to obtain sufficient contiguous tracks of EM to initialize PLATO.
2. While reserving tracks of EM for MASTOR, a track was reserved which was not in increasing sequential order. This condition must be checked so that MASTOR has a contiguous block of EM reserved.

ACTION: Deadstart the system, initialize EM and initiate PLATO again. If the problem persists, report via PSR.

**MXX IMPROPER REQUEST (mxx)**

An improper request has been made to MXX. This could be due to one of the following.

1. Computing the absolute address of an address relative to the caller's RA has resulted in an illegal address.
2. MXX was called by a job which was not system origin.
3. Request parameters passed to MXX are illegal.

ACTION: Report via PSR.

**MXX LOCAL FILE ERROR (mxx)**

One of the following error conditions exists:

1. A file name used by PP program MXX to reserve tracks of EM for MASTOR is already in use at the control point.
2. Some of the ECS tracks requested by MASTOR are already reserved by some other job.

ACTION: In the first case, report via PSR. In the second case, drop the PLATO load job and initiate it again. If the problem persists, deadstart the system, initialize EM and initiate PLATO again. If the problem still persists, report via PSR.

**MXX TRACK ALLOC ERROR (mxx)**

A request to the system to reserve an EM track was rejected. Usually this means that the tracks determined by the "rax" and "fix" PLATO configuration file entries are in use by the operating system or by some other job.

ACTION:

- a. Change "rax" and "fix" to avoid the conflict.
- b. If tracks that should not be in use are still reserved by the system, it may be necessary to deadstart and initialize EM to release the tracks.

**60.14 Messages n**

**NAME NOT FOUND (pf)**

The specified "name" in a nameset-type file does not exist.

ACTION: PF cannot add names to a nameset, they must be added via a nameset editor (system-supplied or otherwise).

**NETON COMPLETE (pnl)**

Connection established with NA1. Informative message.

ACTION: None.

NETOFF COMPLETE (pni)

Disconnection with NAM complete. Apparently, NAM has been dropped.

ACTION: Check NAM status.

NETWORK TABLE MUST BE LENGTHENED (plato)

The common block "link" in file "sysfile", which contains the network system table, is too short to hold the number of systems specified by the value of the "netms" PLATO configuration file keyword.

ACTION: Lengthen the common as described in the "Adding a system" subsection of the "Network Management" section of the PLATO Configuration Handbook.

NETWORK TABLE OBSOLETE--RUN SONETSYS (plato)

This message may be seen when installing PLATO release 34.2 or release 35. The network system table must be converted during the installation.

ACTION: Execute the conversion program in lesson "sUnetsys" when instructed to do so in the installation procedure.

NO DDP REPLY - PORT STATUS (pas)

The DDP or low-speed port is not responding to a request to return the port status word to the PP program. This indicates a hardware problem with the DDP/low-speed port or the channel.

ACTION: Call Customer Engineering.

NO DDP REPLY - MASTER CLEAR PORT (pas)

The DDP or low-speed port is not responding to a "master clear port" function before being used by the PP program for an EM transfer. This probably indicates a hardware problem with the DDP/low-speed port or the channel.

ACTION: Call Customer Engineering.

NO DDP/LSP AVAILABLE (pio)

PLATO attempted to load the CIU driver, PIO, but there was no DDP or LSP defined in the EST (equipment 01). PIO is loaded only when the CIU network is defined in the PLATO configuration file ("nc051" keyword has a non-zero value).

ACTION: Correct the EQPDECK or the PLATO configuration file.

NO DISK SPACE ("account2", "account2a")

This message is found in "s0sysmsg." The system could not find enough physical disk space to create a file of the specified size.

ACTION: Add more disk space to the system.

**NO EM AVAILABLE (ddpt)**

No EM is available for DDPT to do its testing. This could be due to configuration problems or to other batch jobs holding all available EM.

**ACTION:** Rerun the job later. You may need to increase the "ogecs" PLATO configuration file entry.

**NO EQUIPMENT (ddp)**

A channel number was not specified on the DDPT control statement so DDP attempted to find equipment "D1" in the EST and failed.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

**NO FREE COMMUNICATION AREA (conden)**

This message is issued during initialization of the condensor if there is no condensor / PLATO communications area available. This is probably due to trying to bring up a condensor when the maximum number are already active.

**ACTION:** Correct PLATO load procedures.

**NO JUDGE BUFFER AVAILABLE (plato)**

There was no buffer available for a user to store TUTOR judging information in during an auto-break.

**ACTION:** Increase the "jbnks" configuration file entry. If this problem persists, report via PSR.

**60.14.1 Messages NO L**

**NO LESSON BUFFER ENTRY AVAILABLE (plato)**

While loading a master file, EM was not available to read the master file directory into.

**ACTION:** Try again later.

**NO PORT STATUS WORD RECEIVED (pms)**

The DDP or low-speed port is responding to a request to return the port status word to the PP program, but the PP times out while waiting for the DDP/low-speed port to output the status word. This indicates a hardware problem with the DDP/low-speed port or the channel.

**ACTION:** Call Customer Engineering.

**NO RECORDS IN NAME (pf)**

There are no records to be read/written in the specified "name" of a nameset-type file.

**ACTION:** Add the necessary records to the name via a nameset editor (system-supplied or otherwise).

**NO REPLY FROM CIU (pio)**

The CIU would not respond during an attempt to output the

"PLATO OFF" message to all CIU terminals. This indicates a problem with the CIU or channel.

ACTION: Call Customer Engineering.

NO RESOURCES (submit)

There is no PP available to perform the submit or the MASTOR job table is full. The job will continue trying until it succeeds.

ACTION: Normally, none. If this happens often, you may want to increase the "njob" PLATO configuration file entry.

NO RESPONSE CIU INPUT (pio)

The input side of the CIU or its channel is not responding. This indicates a problem with the CIU or the channel.

ACTION: Call Customer Engineering.

NO RESPONSE CIU OUTPUT (pio)

The output side of the CIU or its channel is not responding. This indicates a problem with the CIU or the channel.

ACTION: Call Customer Engineering.

NO REQ AREA (mas)

There is no request area in EM available for MAS-to-MASTOR communication.

ACTION: Rerun the job later. If this happens often, report via PSR.

NO ROOM FOR ANOTHER ARCHIVED FILE IN ACCOUNT xxxx ("account3")  
The archive file table in account "xxxx" is full.

ACTION: Lengthen the account file.

NO TRANSFER PATH AVAILABLE (pas)

This message is only used on 170-700 mainframes. It is issued during the initialization of MASTOR when no UDP or low-speed port (equipment "u2") is defined in the system EQPDECK and the "ncap" PLATO configuration file entry is set to zero. There is no data transfer path between disk and EM.

ACTION: Correct the EQPDECK or the configuration file.

NO USER NUMBER (submit)

There was no user number in a request passed to MASTOR. This is probably a user error due to no USER control statement preceding the SUBMIT control statement.

ACTION: Correct and rerun the job.

60.14.2 Messages NJA

NON-REQUIRED PACKS PRESENT ("s0init")



Some of the master files ATTACHED when PLATO was loaded are not listed in the required packs table.

**ACTION:** Check notesfile "s0sysasg" for a note entitled "extra packs". This note will list master files which should either be added to the required packs table or removed from procedure MFNX.

**NOT ENOUGH DISK SPACE TO CREATE A n PART FILE** ("account2", "account2a")  
This message found in "s0sysasg." The system could not find enough contiguous disk space to create a file of the specified size.

**ACTION:** Add more disk space to the system.

**NOT ENOUGH DUMP DIRECTORY DATASETS** (backcopy)  
The dump directory is stored on-line in datasets. This message means there are not enough datasets to hold the entire directory.

**ACTION:** Use the procedure described in the "Setting up dump directory datasets" section of the PLATO Operations Guide to add the next dataset in sequence and rerun the job.

**NOT ENOUGH ECS TO RUN** (plato)  
Due to configuration or hardware problems, there is not enough EM available to support ten terminals.

**ACTION:** Check the system for other messages indicating a hardware problem. If none, check the PLATO configuration file. You may need to deadstart, initialize EM and initiate PLATO again.

**nnnnnnnnn NOT FOUND** (pfout, pfdest)  
The current PLATO file "nnnnnnnnn" to be dumped from or purged from the master file cannot be found.

**ACTION:** Correct and rerun the job.

**NOT SYGT JOB** (sdp)  
DDPT may only be used in a system origin job.

**ACTION:** Rerun the job under a user name with system origin privileges.

**NUMBER OF RECORDS INCORRECT** (pf)  
The number of records specified is incorrect.

**ACTION:** Verify that the records exist before attempting to access them via PF. PF cannot add records, they must be added via a nameset editor.

60.15 Messages D

**ONLY ONE EXECUTOR** (exec)  
PLATO has been configured to only run one executor, but

an attempt was made to load an executor with a non-zero executor identifier.

**ACTION:** Correct the PLATO load procedures or increase the "nexec" PLATO configuration file entry.

**OVERFLOW--**        **config (vvvvvvv)       (plato)**

This message occurs when a buffer which is controllable by a PLATO configuration file entry overflows. "config" is the relevant entry name and "vvvvvvv" is the amount by which the buffer overflowed. For most overflows, the system will wait for space in the buffer before proceeding. However, if the buffer overflows frequently, this waiting will cause a degradation noticeable to users.

**ACTION:** If the message occurs rarely, or under unusual circumstances (e.g., while installing a release), ignore it. If the message occurs frequently, increase the corresponding PLATO configuration file entry by a minimum of "vvvvvvv". See the PLATO Configuration Handbook for a description of the relevant entry before changing it.

**OVERLAYS NOT CM-RESIDENT       (pms)**

One or both of the PMS overlays, 4PA and 4PB, are not central memory resident as required.

**ACTION:** Add a "\*cm pp/4pa,4pb" entry to the system LIBDECK.

60.16        Messages P

**PACK DIRECTORY ERROR       (plato)**

**mfname        mfnum**

This means that a not-ready status was received for a "master" or "general" type master file of name "mfname" at slot "mfnum" in the master file list. PLATO will abort in this case. Most likely the disk unit that this master file is on was/is not ready, and has spun down.

**ACTION:** Start the unit spinning again, restart PLATO, and contact Customer Engineering, if necessary.

**PACK DISMOUNTED       mfname       (plato)**

This means that a not-ready status was received for a "binary" or "backup" type master file of name "mfname". PLATO will not abort in this case, but will turn off this master file. Most likely, the disk unit this master file is on was/is not ready, and has spun down.

**ACTION:** Start the unit spinning again. If the master file was of type "binary", use lesson "ldr" to initialize it, then turn it back on. For any other master file type, use lesson "ldr" to turn the master file back on. Call Customer Engineering if necessary.

**PACK NOT LOADED       (ppack)**

The master file requested for future file references by

this batch job (via PF) is not currently active on the PLATO system.

**ACTION:** Load the master file through lesson "ldr" and run the job again.

**PARAMETER WORD DDR (pms)**

An internal check for a valid parameter passed from MASTOR to PMS during initialization has failed. The parameter word address in the PP input register is incorrect.

**ACTION:** Report via PSR.

**PARCEL OVERFLOW (fraeat)**

Almost all of the output parcels available are in use.

**ACTION:** It is sometimes possible that terminals pumping keys may cause this problem. You may want to check the various system error displays to see if any station has a high number of errors. If this message occurs frequently, increase the number of output parcels available by increasing the "nparc" PLATO configuration file entry. If the system runs out of parcels altogether, the system will hang.

**PARTIAL ECS (mastor)**

Due to configuration or hardware problems, there is a smaller than usual amount of ECS available for the PLATO system. PLATO will continue to run on half of its usual amount of ECS.

**ACTION:** Attempt to correct by dropping all jobs with EM reserved or checking/correcting the PLATO configuration file. Call Customer Engineering if there are other messages indicating an EM error.

#### 60.16.1 Messages PL

**PLATO ALREADY ACTIVE (exec)**

This message is issued if you try to bring up a second primary executor.

**ACTION:** Correct the PLATO load procedures.

**PLATO DUMP FILE ALREADY EXISTS (plato,exec)**

This is an informative dayfile message issued when the PLATO dump procedure detects that the required dump file already exists. After this message is issued, a flashing WAIT control statement appears on the 3-display along with the message "GO TO OVERWRITE DUMP".

**ACTION:** See "GO TO OVERWRITE DUMP".

**PLATO FILE DIRECTORY BAD (pf)**

When the requested PLATO file is opened, certain validity checks are performed on the file directory. This test has failed.

**ACTION:** Recover the file from the most recent backup.

**PLATO FILE HEADER BAD (esm)**

This message is issued if the first three words of the first record of dataset "sueserr" have incorrect values.

**ACTION:** See the section titled "Automatic Loading and Monitoring" in the PLATO Configuration Handbook for the correct values and correct the file.

**PLATO FILE NOT AVAILABLE (pf)**

The PLATO file requested cannot be opened. Possibly user error.

**ACTION:** Check Jayfile for more information. Correct and rerun the job.

**PLATO FILE READ ERROR (pf)**

A disk error was detected while trying to read from the requested file.

**ACTION:** Call Customer Engineering.

**PLATO FILE READ RETRY (pf)**

A disk error was detected while trying to read from the requested file. The job will attempt three re-tries before giving up.

**ACTION:** Call Customer Engineering.

**PLATO FILE TYPE ERROR (pf)**

The file requested is of an illegal type for the option being used.

**ACTION:** See lesson "nosalds" for the correct use of this control statement.

**PLATO FILE TYPE NOT DATASET (pf)**

The file requested is not a dataset when a file of that type is required for the option being used. User error.

**ACTION:** See lesson "nosalds" for the correct use of this control statement.

**PLATO FILE TYPE NOT NAMESET (pf)**

The file requested is not a nameset when a file of that type is required for the option being used. User error.

**ACTION:** See lesson "nosalds" for the correct use of this control statement.

**PLATO FILE WRITE ERROR (pf)**

A disk error was detected while trying to write to the requested file.

**ACTION:** Call Customer Engineering.

## 60.16.2 PNA

PNA - xxxxxxxx ABORTED (pna)  
 PNI aborted job "xxxxxxx" via PP program PNA. Informative message.

ACTION: None.

PNA CANNOT FIND MASTOR (pna)  
 This message is issued when PNA attempts to drop PLATO via a K-display message to MASTOR and the control point cannot be found.

ACTION: Report via PSR.

PNA - xxxxxxxx CANNOT BE DROPPED (pna)  
 PNA could not find job "xxxxxxx", so it was not dropped. Informative message.

ACTION: None.

PNA - xxxxxxxx PURGED FROM QUEUE (pna)  
 PNA purged job "xxxxxxx" from the input or rollout queue. Informative message.

ACTION: None.

PNI - ABORTR TIMEJUT (framot)  
 Informational message. This message displays a count of the number of time an "abort" message was sent out to the network, but no response came back in the expected time.

ACTION: None. Report via PSR if the number displayed is very high.

PNI DROPPING PLATO (pni)  
 PNI is dropping PLATO. Informative message.

ACTION: None.

PNI DUMP FILE ALREADY EXISTS (pni)  
 This is an informative dayfile message issued when the PNI dump procedure detects that the required dump file already exists. After this message is issued, a flashing WAIT control statement appears on the 3-display along with the message "GO TO OVERWRITE DUMP".

ACTION: See "GO TO OVERWRITE DUMP".

## 60.16.3 Messages PNI E

PNI ERROR xxxx-yyzz ON ttttt sssss (pni)  
 xxxx = primary error code  
 yyzz = detailed error information (only supplied with certain primary error codes)  
 ttttt = terminal identifier  
 sssss = PLATO station number

## Primary error codes:

- 1 = NAM reported a connection broken for a connection that does not exist
- 2 = FC/ACK received with no outstanding blocks
- 3 = FC/INIT received out of sequence
- 4 = FC/NAK received
- 5 = unknown asynchronous message code (PFC/SFC)
- 6 = Input block not delivered (I3J bit set)
- 7 = parity error occurred
- 8 = Input received on a connection not initialized for PLATO data
- 9 = unknown synchronous message code (PFC/SFC)
- 10 = unknown input block type received
- 11 = NAM requested connection of a port to PLATO and when no more PLATO ports are available  
ACTION: The PLATO configuration file entry "nXsit" should be increased to accommodate more users.
- 12 = NAM reported that a connection to PLATO was terminated abnormally  
ACTION: This may be caused by communications errors or faulty communications hardware. Check the NDS system error log. This may also be due to the user disconnecting the communications line by hanging up the phone without signing out first.
- 13 = PNI attempted to send more message blocks to the network than allowed (see NAM "napi" parameter in the NAM Network Definition Language Processor Reference Manual)
- 14 = break (FC/BRK) received from NAM
- 15 = an invalid connection number was received from NAM
- 16 = PNI rejects a request to access PLATO  
ACTION: Normally, none. This error code often occurs in conjunction with error code 17.
- 17 = PNI rejects a request to access PLATO because the PLATO port is currently active  
ACTION: Normally, none. The active PLATO port is backed out so the new connection can be made. Report via PSR if this error code is seen very often in a short period.
- 18 = error logical received, "yy" is the ERR/LGL error code (see NAM reference manual for codes)
- 19 = a terminal resident load block is being retransmitted to the terminal  
"xx" = load file number  
"yy" = load file block number  
ACTION: None. Informative message.
- 20 = PNI received a NAK on a terminal resident load block because the block number is out of range
- 21 = PNI has received an unknown output request from FRAMAT

ACTION: Since many of these messages are the result of communications problems, occasional errors should be expected as part of normal daily operations. If any of the following errors occurs frequently, and is correlated with reported communications or systems problems such as hung terminals, users unable to log in, slow response time, lost input or output, contact Customer Engineering and check the system

error log for NAM/CCP messages which indicate the source of the problem.

A PSR should not be written solely on the basis of one occurrence of a message. Only write a PSR if a demonstrable problem occurs which cannot be explained by hardware or operational errors. When writing such a PSR, be sure to follow standard PSR procedures and include all appropriate dumps and dayfiles. The PSR should describe the problem which occurred, not only the error message(s) seen, since the messages may or may not be directly related to the actual problem.

60.15.4 Messages PNI F

PNI LOADING PLATO (pni)  
PNI is loading PLATO. Informative message.

ACTION: None.

PNI NOT SUPPORTED (pni)  
PNI is not available on this system.

ACTION: If PNI is desired, change the "nam" PLATO configuration file entry.

PNI - REQUEST DROPPED JJJJJJJ (pni)  
PNI dropped a completed request for job "JJJJJJJ". This means that the user aborted the job after a request was made, but before it completed. PNI eventually clears the completed request, if the job doesn't pick up the request in a reasonable time.

ACTION: None.

PNI. TRACE ACTIVATED. (pni)  
Informative message in response to the CFD type-in "DBG,ON.", indicating that logging of data and supervisory messages has begun.

ACTION: None.

PNI. TRACE DEACTIVATED. (pni)  
Informative message in response to the CFD type-in "DBG,OFF.", indicating that logging of data and supervisory messages has been suspended.

ACTION: None.

PNI. TRACE FILE RELEASED. (pni)  
Informative message in response to the CFD type-in "DBG,DJT.", indicating that the trace file "ZZZZZD1" has been routed to the input queue.

ACTION: None.

PNI. TRACE NOT AVAILABLE (pni)

Informative dayfile message in response to CFU commands "DBG,ON.", "DBG,OFF." or "DBG,OUT.", indicating that the running version of PNI has not been loaded with the debug version of the network library, NETIOD.

ACTION: None.

PNI UNKNOWN CFD REQUEST (pni)  
Informative dayfile message in response to any unrecognized CFD entry for PNI.

ACTION: See the "Network Management" section of the PLATO Operations Guide for correct CFU entries.

#### 60.16.5 Messages PJ

POSSIBLE SECURITY BREACH ("plato","authors")  
The following note is written into "s0sysmsg" by lesson "plato":

```
Possible security breach--
signon = plato name/group
tries at password = 10
station = 32-0, net port pni03e
```

The same message is issued by "authors" but the sign-on of the person attempting to break security is stored in the note header.

ACTION: Operations should contact the director of the account involved and warn them of the attempt to breach security.

PPU DROPPED (pas,pib)  
This message is issued when a ppu program has completed processing and is about to drop from its control point. This could be a normal completion following a "K.STOP" request to MASTER, or due to an error exit in the calling program.

ACTION: Check dayfile for more information.

PREMATURE END-OF-FILE (pfin)  
The input data to be written to the new PLATO file was truncated. The number of blocks read from the input file did not agree with the number of blocks in the file as read from the directory block. This is probably due to the input file not having been created by PFJUT or PF(BIN).

ACTION: Correct the job and rerun.

PREVIOUS LINE IN ERROR (master,plato)  
This messages occurs only during initialization while the PLATO configuration file is being processed. Each entry in the configuration is copied into the job dayfile as it is processed. This message will follow any entry which cannot be processed due to one of the following



## errors:

1. Display-codes to binary conversion error. A number which has an octal post-radix contains an 8 or 9.
2. An entry which should have an "on/off" value has some other value.
3. An entry which should have a three-character value has some other value.

ACTION: Correct the configuration file.

PRJUTE COMPLETE (proute)  
Informative message.

ACTION: None.

PRJUTE CONTROL CARD ERRJR (proute)  
The PRJUTE control statement is incorrect.

ACTION: See the PLATO Operations Guide for the correct format for this control statement.

PRJUTE INVALID KEYWORD (proute)  
An argument on the PRJUTE control statement is incorrect.

ACTION: See the PLATO Operations Guide for the correct format for this control statement.

PRJUTE INVALID \*CP\* ARGUMENT (proute)  
The parameter specified for the "cp" keyword on the PRJUTE control statement is incorrect.

ACTION: See the PLATO Operations Guide for the correct format for this control statement.

PRJUTE - RSB ABORT (proute)  
The PRJUTE program was unable to read central memory via the RSB monitor call.

ACTION: Report via PSR.

## 60.17 Messages Q

QUESIZ NOT A POWER OF 2 (plato)  
The "quesz" PLATO configuration file entry is incorrect.

ACTION: See the PLATO Configuration Handbook for the correct format for this entry.

## 60.18 Messages R

RAX REQ xxxx / ACT yyyy (xastor)  
The starting address of the EM reserved for PLATO (RAX) is different from the RAX requested by the "rax" PLATO configuration file entry. The requested RAX is "xxxx" while the actual RAX is "yyyy". The values "xxxx" and "yyyy" are in terms of 1000B words. If the actual value of RAX and the

requested value of RAX are the same, this message will not be issued.

**ACTION:** None. The PLATJ application will continue to run on the reduced amount of EM unless that amount is below a critical value. In that case, PLATO will abort with other messages indicating the problem.

**RECALL TIMEOUT (djp)**

DDP waits 50 milliseconds for DDPT to go into recall before reading the exchange package. This wait has timed out.

**ACTION:** Report via PSR.

**RECOVERED ECS PARITY ERROR (master,plato,format,conden)**

```
(1) ++++++ yy/mm/dd ++++++
(2) RECOVERED ECS PARITY ERROR
(3) ECS ADDR (7 digit octal number)
(4) CONTENTS (20 digit octal number)
(5) TEST DATA (20 digit octal number)
(6) READ AS (20 digit octal number)
(7) FWA TRANS (7 digit octal number)
(8) LTH TRANS (7 digit octal number)
(9) CALL ADDR (20 digit octal number) xx
(10) ++++++
```

EM error messages are written to the job dayfile whenever an error occurs. Part or all of the information above may be included in the message depending on the circumstances.

- 1) This line always appears with the date the message occurred.
- 2) This line always occurs. However, under some circumstances the messages will say "UNRECOVERED ECS ERROR". For an unrecovered error, the job aborts after displaying the message (it may however restart itself). When a recovered error occurs, the software will repeat the same instruction, hopefully without error.
- 3) & (4) PLATO may have requested a transfer of several words. The software will loop, reading a word at a time, trying to isolate the problem to the exact word. If it is able to do so, these lines will show the location of the bad word and its contents.
- 5) & (6) If the software is able to isolate the error to a single word, it will next try out some test patterns to see if it can isolate the error still further. It first writes the pattern to EM and then tries to read it back. Several patterns are tested until one fails (assuming it does). The pattern itself, and what was read instead, are both displayed if an error occurs. Patterns are tested in the following sequence:
  - (a) All zeros.
  - (b) All ones.

- (c) Alternating ones and zeros.
- (d) The original words with each bit complemented.

7) through (10) These will always be displayed. The call address is the address in PLATO's field length from which the original read/write ecs occurred. The "FWA TRANS" and "LTH TRANS" are the first word and length. As part of the call address line, you will be shown the type of instruction which caused the error (RE, WE, RX or WX will replace "xx").

**ACTION:** Call Customer Engineering. In the event of unrecovered errors, immediate action may be necessary as it might prevent PLATO from running.

#### 60.18.1 Messages RED

##### REDUCED ECS (plato)

Due to configuration or hardware problems, there is a smaller than usual amount of ECS available for the PLATO system. PLATO will continue to run on half of its usual amount of ECS.

**ACTION:** Attempt to correct by dropping all jobs with EM reserved or checking/correcting the PLATO configuration file. Call Customer Engineering if there are other messages indicating an EM error.

##### RELATIVE FWA DDR (ddp)

The "FW" parameter on the DDPT control statement is incorrect.

**ACTION:** See the PLATO Operations Guide for the correct format for this control statement.

##### \*\*\*\*\* - REQUIRED MF NOT DUMPED (backone)

Master file "\*\*\*\*\*" is listed in the dump-required table but was not included in the master files to be dumped as listed in PROC/MFDX.

**ACTION:** Change PROC/MFDX or use BACKMOD to change the dump-required table so the two agree.

##### ROLLOUT REQUESTED (ddp)

Informative message - the operator or operating system has requested the rollout of the DDP/low-speed port test program.

**ACTION:** None.

##### RUN ABORTED (backone)

An error has been detected during the backups database merge.

**ACTION:** Check the job dayfile for more information.

##### "runrexec" - RUNNER, DETECTED ERRORS ("runrexec")

This message found in "s0sys.sg." It means that one of

the runner program has found an error.

**ACTION:**

- a. Execute lasso "runner".
- b. Choose the option to "execute the runner executive".
- c. Any error will be shown on the screen. Use the information given to try to fix. It may be necessary to obtain a backup of the corresponding comcon or namaset.

60.19 Messages 5

**SECURITY COUNT ZERO (submit)**

The security count word in the user's validation file entry has been decremented to zero. The user name will not be usable until this count has been increased. See the NDS Administration Handbook for more information.

**ACTION:** Reset the user's security count.

**SETFRAM - CIU OVERFLOW - nnnn (frammat)**

The output buffer used by FRAMAT to send data to the CIU network has overflowed. "nnnn" is the PLATO port number.

**ACTION:** Check that the CIU network is still operational. If it is, ignore this message. If not, report via PSR.

**SETFRAM - PNI OVERFLOW - nnnn (frammat)**

The output buffer used by FRAMAT to send data to the PNI network has overflowed. "nnnn" is the PLATO port number.

**ACTION:** Check that the PNI network is still operational. If it is, ignore this message. If not, report via PSR.

**SITE ASSIGNED TO DISABLED NETWORK (plato)**

This message can only appear during processing of the configuration file. A physical site is assigned to a given network by the configuration file, but that network is disabled by an "on/off" entry or by a "C" numeric entry.

**ACTION:** Correct the configuration file.

**SITE DUPLICATEDLY DEFINED (plato)**

This message can only appear during processing of the configuration file. A physical site is assigned to two different networks by the configuration file.

**ACTION:** Correct the configuration file.

**SITE NOT ASSIGNED TO NETWORK (plato)**

This message can only appear during processing of the configuration file. A physical site is not assigned to any network by the configuration file. The sum of all defined networks is not equal to the total number of physical sites defined.

**ACTION:** Correct the configuration file.

**SITES DEFINED OUT OF RANGE (plato)**

This message can only appear during processing of the configuration file. A physical site number assigned to a network by a configuration file entry is greater than the total number of sites defined by the configuration file.

**ACTION:** Correct the configuration file.

**SLOT NUMBER OUT OF RANGE (backmod,backone,backtwo)**

A slot number in the dump cycle slot table is out of range (less than 1 or greater than 30).

**ACTION:** Use BACKMOD to correct the dump cycle slot table.

**SOMETHING IS WRONG WITH THE AIDS NOTES FILE ("aids")**

This message is found in "slsysmsg". An error was detected when "aids" attempted to attach the TERM-comments notesfile via the -notes- command. The "zreturn" from the -notes- command is given.

**ACTION:** Determine the cause of the problem and fix it if possible. Otherwise, report via PSR.

**60.19.1 Messages ST****STATION TOO BIG (plo)**

The station number word part of an input key is larger than the number of stations on the system. This could be a problem of configuration or of hardware.

**ACTION:** Correct PLATO configuration file or call Customer Engineering to check out CIU.

**STOPPING MASTOR (mastor)**

This is a temporary B-display message which is seen while the PLATO system is being shut down following an operator "K.STOP" request to MASTOR.

**ACTION:** None.

**SUBFILE LOAD ERROR (plato)**

This message may be seen during initialization of the PLATO system. It is issued when one of the following databases cannot be loaded. The file and block names are given immediately following this error message.

- |            |             |                           |
|------------|-------------|---------------------------|
| 1. micro   | / standard  | Standard micro table      |
| 2. sysfile | / sizechars | Standard sized characters |
| 3. sysfile | / ecsallot  | ECOS allocation table     |
| 4. sysfile | / signoncom | Auto-signon table         |
| 5. sysfile | / lessons   | Reserved lesson list      |

**ACTION:** Check for missing master files and correct PKDD/MFNX to attach all required master files. If this does not resolve the problem, recover the file from the most recent backup.

**SUBMIT ABORT (submitm)**

A request to MASTOR to submit a job was rejected by the operating system. See the NDS Reference Manual (ROUTE macro) for more information.

**ACTION:** Correct the cause of the problem and rerun the job.

**SUBMIT ERROR xx. (mastor)**

An error has been detected by MASTOR when it attempted to submit the job which loads the other PLATO control points. "xx" is an error code which may have these octal values:

- 10 = submit abort - the operating system has aborted the submit request, usually because the user name for submitting PLATO jobs (specified by the "subun" configuration file keyword) is not allowed to submit system service class jobs.
- 11 = security count for the user name used to submit PLATO jobs (specified by the "subun" configuration file keyword) is zero.
- 12 = deferred batch job limit for the user name used to submit PLATO jobs (specified by the "subun" configuration file keyword) has been reached.

**ACTION:**

- For xx=10, set the MODVAL "VM" parameter to "ALL".
- For xx=11, set the MODVAL "SC" parameter to "UNLIMITED".
- For xx=12, set the MODVAL "DB" parameter to "UNLIMITED".
- For any value of "xx" other than the above, report via PSR.

**SYSTEM ID .NE. ROUTING ID (plato)**

SYSTEM ID = sss

ROUTING ID = rrr

The routing id ("rrr") as determined by the "rid" PLATO configuration file entry is different from the system identifier ("sss") stored in PLATO file "s0file".

**ACTION:**

- a. If this is a new installation, use the network options in lesson "ipedit" to initialize the system identifier.
- b. Verify that "rid" is set correctly in the PLATO configuration file.
- c. Get a backup of file "s0file" if it has been damaged.

**SYSTEM \*xxxxxxx\* NOT IN NETWORK TABLE (plato)**

The system name "xxxxxxx" defined by the "sid" PLATO configuration file entry could not be found in the network table.

**ACTION:**

- a. Verify that the "sid" configuration file entry is set to the desired value.

- b. If "sid" is correct, use lesson "ipedit" to add system "xxxxxxx" to the network table.

60.20 Messages T

TABLE NOT FOUND (mastorn)

During initialization of MASTORN, EM addresses of several tables are read from EM and stored in CH. MASTOR should set up these pointers and tables before MASTORN is brought up. At least one of the tables pointed to was not initialized.

ACTION: Report via PSR.

xxxx TERMINALS ACTIVE (pnl)

Informative message. "xxxx" is the decimal number of terminals currently active in the network.

ACTION: None.

THE FOLLOWING PACKS ARE NOT LISTED IN THE REQUIRED PACKS TABLE IN "IPEDIT" BUT ARE CURRENTLY ATTACHED ("s0init")

This message is found in "s0sysmsg." The note also lists the names of all active master files which are not in the required master file table.

ACTION: Add the missing master file names to the required master files table or remove them from PROJ/MFNX.

TIMEDOUT WAITING FOR CPM (mastor)

Something went wrong with certain PP/CPM functions which do not set completion bits. The chance of seeing this message is extremely remote. If it does occur, it is because the user number for a submit was deleted while the submit was taking place or because the NDS validation file or system pack are damaged.

ACTION: Check NDS validation files and system pack. If everything appear to be correct, report via PSR.

TOO MANY DEFINE NAMES (tprint)

A lesson has too many defined symbols for the print program to handle. The symbol cross-reference table has overflowed

ACTION: Report via PSR.

TOO MANY DEFINE SETS (tprint)

A lesson has too many define sets for the print program to handle. The define set name table has overflowed.

ACTION: Report via PSR.

TOO MANY FORMATTERS (format)

This message is only used on systems running multiple formatters. An attempt has been made to bring up more formatters than the system can handle.

**ACTION:** Correct the PLATO load procedures.

**TOO MANY MASTER FILES (pms)**

There are more master files attached to MASTOR's control point than MASTOR is set up to handle.

**ACTION:** Remove extra master files from PRDC/MFNX or increase the "ndsus" PLATO configuration file entry.

**TOO MANY UNITS (tprint)**

A lesson has too many units for the print program to handle. The unit cross-reference table has overflowed.

**ACTION:** Report via PSR.

**TRACK LIMIT (mfcreat)**

MFCREAT has run out of disk space while allocating space for the new master file.

**ACTION:** The initialization of the file is incomplete. It must be purged and recreated, either with a smaller space request or on a different disk pack.

**TRACK xxxx nnnnnnnn (mfu)**

Informative message. This message is issued by PF program MFU when it is called by MFFIND to search a disk pack for possible master files. "xxxx" is the NDS logical track where the master file begins, and "nnnnnnn" is the first word of the file (which should should be the master file name).

**ACTION:** None.

**60.21 Messages U**

**ULIB RECORD MISSING (mastor, mastorn, plato, framat, conden)**

The format of the file on the deadstart tape which holds overlays for this control point is incorrect.

**ACTION:** Report via PSR.

**UNABLE TO RELEASE ECS (pf)**

This message is used only by the nameset options in PF. During initializations, PF releases all EM it currently has allocated to it before requesting more. The request to release the EM failed.

**ACTION:** Report via PSR.

**UNABLE TO RESET TO FAMILY (mastor)**

After a job was submitted to NDS under an alternate family, MASTOR was unable to reset to its own family. This is probably due to the "mass storage table" (MST) for the family device and/or the family device itself having been damaged, or the "device access table" (DAT) having been overwritten.



ACTION: Report via PSR if the above suggestions do not resolve the problem.

UNEXPECTED OVERLAY--xxxxxxx (plato,frammat,conden)  
An attempt was made to load an overlay with an incorrect name. Overlay load file is probably incorrect.

ACTION: Report via PSR.

++++ (DDP) UNKNOWN ERROR ++++++ (ddp)  
+  
+ CHANNEL cc  
+  
+++++

An attempt was made by DDP to read (write) a certain number of PP words of data from (to) EM. The transfer was terminated and control returned to the PP before the word count had gone to zero.

ACTION: Contact Customer Engineering.

UNRECOVERED ECS PARITY ERROR (master,plato,frammat)  
This is described in detail in the section on RECOVERED ECS PARITY ERROR messages.

ACTION: Call Customer Engineering.

60.23 Messages w

WAIT FOR ECS (pf)  
WAIT FOR ECS (xxxxx) (terint,dprint,nprint)  
The amount of EM this batch job needs is not available. The amount of EM requested may be displayed as "xxxxx". The job will wait until the EM is available.

ACTION: None.

WAIT FOR MASTER (pf)  
MASTER is not responding to a request from PF during initialization. PF will continue to loop until MASTER responds, or MAS aborts the job.

ACTION: Check dayfile for reason for MASTER failing to respond.

WAIT FOR PLATO CRASH DUMPS TO FINISH. (copyed)  
This message is issued by the job which copies PLATO dump files to tape.

ACTION: The operator must wait until all PLATO jobs have completed their dump procedures, then give this job a "GO".

WAIT FRAMAT (pio)  
This message is issued when FRAMAT is initializing, either following an initial PLATO load or when FRAMAT

has been restarted following an EM error or other abort.

ACTION: Wait for a while to see if FRAMAT comes up. If this message is still being issued after a minute, report via PSR.

WAITING FOR CLOCK TO STOP (exec)

This message will be shown on the B-display by an executor which is being dropped through the "executor drop" option on multi-executor systems. If this message is displayed for longer than a few seconds, the executor may be hung.

ACTION: None. If the message is displayed for more than a few seconds, report via PSR.

WAITING FOR ECS (plato,tform)

During initialization of the program, it requested allocation of EM from MASTOR but the request was refused. The request will be repeated until it succeeds.

ACTION: None.

WAITING FOR EXECUTOR (plato)

This message is used only on multi-executor systems. The inter-executor communications buffer in EM is almost full so PLATO pauses to let the other executor catch up. If this message persists, the executor may be hung.

ACTION: Normally, none. If this message persists, report via PSR.

WAITING FOR FILE - ffffffff (backcopy)

A file operation is pending on file "fffffff" but the file is currently busy elsewhere. The job will wait until the file is released.

ACTION: None.

WAITING FOR FILE TABLE SPACE (pf)

MASTOR's active file table is full so the requested file cannot be opened. The job will wait until the file is opened.

ACTION: None.

WAITING FOR FRAMAT (plato)

See "WAIT FRAMAT".

WAITING FOR MASTOR (plato)

The MASTOR / PLATO communications buffer in EM is almost full. PLATO pauses to let MASTOR catch up to insure that the next user will have a slot available if he needs it. If this message persists, either MASTOR or PLATO may be hung.

ACTION: Normally, none. If this message persists, report

via PSR.

**WAITING FOR PLATO (exec,pnl)**

The control point is waiting for the main PLATO executor to complete initializations. If the message does not go away after a few minutes, the primary executor may be hung. If the "naa" PLATO configuration file entry is not set correctly, PNI may hang on this message.

**ACTION:** Normally, none. If the message persists, report via PSR.

**WAITING FOR PLATO FILE (pf)**

The requested file is currently attached. The job will wait until the file is available.

**ACTION:** None.

**WAITING FOR TABLE --- tttttt (mastorn)**

MASTORN is waiting for the initialization of table "tttttt" to complete before proceeding with its initialization.

**ACTION:** Normally, none. If the message persists, report via PSR.

**WARNING "placom/plusers" HAS BEEN INITIALIZED**

This message is found in "scsyslog". Informative message.

**ACTION:** None.

**60.23.1 Messages W3**

**xxx - WRITE ECS ERROR (wrq, mas)**

An error has been detected while trying to write to EM. The EM, DDP or low-speed port or the controller may have a problem. The transfer length varies from 1-5 words.

**ACTION:** Call Customer Engineering.

**WXC3 OVERFLOW CALL ADDRESS = xxxxxx (freat,foraet)**

The FORMAT to FRAMER circular buffer has overflowed. "xxxxxx" gives the address of the calling program within the issuing job.

**ACTION:** If the error only happens during software installation or only occurs occasionally, it can be ignored. If it occurs frequently, check to see if there is a terminal which is passing keys. If you are unable to find the source of the problem, report via PSR.

**60.26 Messages Z**

**ZERO PATTERNS (ddp)**

One of the parameters passed from DDPT to DDP is incorrect.

**ACTION:** Report via PSR.

## 70 Release Changes

## 70.1 Release Changes

The following changes have been made to the PLATO Operations Guide for the current release.

TOPIC	SECTION	CHANGE	DELETE	ADD
PLATO Inter-system Link	1.2	X		
	5.1	X		
	9			X
	40	X		
	40.2		X	
Mini-release installation procedures moved to PLATO Installation Guide	15		X	
			X	
Add ISD II (936) disk support	21.12.3	X		
Add dayfile message	60.1	X		
Add dayfile message	50.3.1	X		
Add dayfile message	50.3.2	X		
Add dayfile message	60.12	X		
Add dayfile message	60.19.1	X		

83 Alphabetical Index

ALPHABETICAL CROSS-REFERENCE INDEX

This index supplements the Table of Contents. Items will be found in alphabetical order. Entries beginning with a number follow "z" and entries with a number as the second character will be found at the end of that alphabetical list. For example, "sGarch" will be found following "sz".

Error messages are listed in alphabetical order in section "Error Messages". They are not repeated in the index.

Special symbols:

Lesson names and PLAT3 configuration file entries are delimited by double quotes ("").

Program names, procedure names and NDS file or user names are capitalized.

## 80.1 Index: A

"accerrlog" .....	2.8.4
"acclog0" .....	22.1
account (see "PLATO account" or "NJS account file")	
account access list .....	4
	4.2
	4.7
account dayfile (see "NJS account file")	
account director .....	1.5
	4
	4.2
	4.6
	22.1
account file management log (see "file management log")	
account owner .....	1.5
	4
	4.1
	4.2
	4.7.3
	6.1
	10.2.1
	10.2.2
	14.2
account summary data .....	10.2
	10.2.1
	10.2.2
"accounts" .....	1.1.9
	1.3.2.2
	4
	4.1
	4.4
	4.7.3
	4.7
	10.2
	10.2.1
	22.1
"accountu" .....	1.1.9
	2.8.4
	4.7.2
	4.8
"account1" .....	1.1.3.1
ACCPRT .....	1.1.8.1
	22.1
add-on PLATO products .....	15.7
	40
"aids" .....	1.1.9
	7.8
	13.3
"alarm" .....	5.1
"allocate" .....	1.1.9
	14
	14.2
application utilities .....	1.1.3.4
ARCHIVE file .....	5.1
ARCHIVE master file .....	1.3.2.1

	6.2
	6.4
	21.12.3
archive processing .....	1.1.9
	6.4
	6.5
archive rights .....	4.2
	6
	6.3
	10.2.1
	10.2.2
"archiver" .....	1.1.9
	6.4
	6.5
ASH1 .....	1.1.8.6
	10.3.3.1
	10.3.3.2
audit trail .....	1.1.8.5
	7.2
	7.10.1
	7.10.3
	7.10.6
author deletion (see "logical site")	
auto-signon (see "logical site")	
availability data (see "usage data")	
"aUpsoless" .....	13.3
"aOss1" .....	13.3
80.2 Index: B	
BACKCOPY .....	1.1.8.5
	7.2
	7.3
	7.4
	7.5
	7.6
	7.7
	7.10.1
BACKDMP .....	1.1.8.5
	7.2
	7.7
	7.10.2
BACKLIB .....	1.1.8.5
	7.10.3
BACKLST .....	1.1.8.5
	7.10.4
BACKMOD .....	1.1.8.5
	7.3
	7.4
	7.5
	7.6
	7.10.5
BACKONE .....	1.1.8.5
	7.2
	7.10.6
	7.10.7
	7.10.9

backout .....	2.2.3 2.3.6 2.3.13 2.8.1 10.3.1 14.3
backout entire site (see "logical site")	
BACKTND .....	1.1.8.5 7.2 7.10.7 7.10.9
BACKUP master file .....	1.3.2.1 1.3.2.2 21.12.5
"backups" .....	1.1.8.5 1.1.9 7.3
backups utilities .....	1.1.8.5
batch job utilities .....	1.1.8.2 1.1.8.3
"bgecs" .....	21.3 21.5
billing cycle .....	1.1.8.6 10.1.3 10.1.5 10.2 10.3.1 10.3.2 10.3.4 10.3.5 10.3.7
"binary" (see also "lesson binary")	1.1.9 1.3.2.1 5.1
BINARY master file .....	1.1.9 1.3.2.1 1.3.3 2.3.2.1 5 5.1 21.12.3
BKSTART .....	1.1.8.5 7.10.8
broadcast unit .....	2.1.2 2.2.5 2.3.7 2.3.12 2.5
"bulfile" .....	12
80.3 Index: C	
"c" .....	1.1.9 13.1
"catalogs" .....	11.1
CCP .....	3.1.1



CFD commands (PNI) .....	3.2.2
"checkpt" .....	1.3.2.1
	5.1
CIU .....	1.1.3
	1.1.4
	2.1.2
	2.2.5
	2.3.4
	2.3.7
	2.3.12
	2.5
	10.1.1.6
CM dumps (see "dumps")	
CMDMP .....	1.1.8.4
	20.1
	20.8
COA1 (see "CONDENSOR")	
COMBUF (see "communication buffer")	
80.3.1 Index: CO	
communications buffer .....	1.1.8.5
	7.2
	7.10.6
	7.10.8
	7.10.9
Computer Interface Unit (see "CIU")	
CONDENSOR .....	1.1
	1.1.3
	1.1.6
	1.1.7
	1.1.8.4
	2.3.5
	2.3.6
	20.2
	30.3.2.3
CONDX .....	1.1.6
	1.1.8.4
	2.3.5
	20.2
configuration files .....	1.1
	1.1.1
	1.1.3.4
	1.2
	1.3.2.1
	4.6
	10.1.1.1
	10.1.1.7
	20.3
	21.3
	21.5
	70.1
Configuration Handbook (see "PLATJ Configuration Handbook")	
CONFIGX .....	1.1.8.4
	20.3
CONSOLE .....	1.1.3.2
	21.1

consultant (see "System Consultants")	
consultant features .....	1.4.6
	13
"convert11" .....	8.1
	8.1.2
"convert10j" .....	8.1.2
	8.1.4
COPYMF .....	7.2
	7.5
	7.10.2
	7.10.9
	7.10.11
COPYPD .....	1.1.8.2
	20.8
	21.2
COPYPF .....	7.10.10
copying dumps (see "dumps")	
copying master files (see "master files")	
copying PLATO files to tape .....	2.9
COPYRF .....	1.1.8.5
COPYRF .....	1.1.8.5
courseware .....	1.4.2
	4.1
	10.1.1.7
	11
	11.1
	11.2
	30.2
	30.3.1
courseware access bits (see "Lesson access bits")	
courseware action request (see "PSR")	
courseware installation .....	1.1.8.2
Courseware Services .....	1.4.2
"cpspd" .....	10.1.1.7
crash log .....	1.1.9
	2.3.13
current usage controller (see "enforcer")	
Customer Engineering .....	1.4.5
	2.3.3
	2.3.11
	2.3.12
	2.7
CYBER 170-700 .....	1.1
CYBER 170-800 .....	1.1
	1.1.3
80.4 Index: D	
DATISON .....	1.1.8.6
	10.3.7
dayfile (see "system dayfile")	
DDPT .....	1.1.8.2
	21.3
deadstart .....	2.1.1
	2.1.3
	2.2.8
	2.3.0

	2.3.11
	2.3.12
	2.8.3.1
	10.3.7
deadstart dump .....	2.3.6
	2.3.9
	2.3.10
	30.3.2.2
DFTERM .....	7.7
	10.1.2
	10.3.1.2
disconnect accounts common .....	4.8
disk errors (see "hardware errors")	
disk space management .....	1.1.8.2
	1.1.9
	1.3.2.1
DOCPRT .....	1.1.8.1
	22.2
documentation .....	1
	1.1.9
DPRINT .....	1.1.8.1
	22.3
dump cycle .....	7.1
	7.3
dump directory .....	1.1.8.5
	7.2
	7.4
	7.10.1
	7.10.4
	7.10.5
	7.10.7
dump utilities (see "utility programs")	
DUMPPRT .....	1.1.8.2
	21.4
dumps .....	1.1.8.2
	1.1.8.4
	2.3.4
	20.6
	20.7
	21.2
	21.4
	21.6
	21.3
	21.14
	30.3.2.2
duplicate PLATO files .....	2.3.4
80.5 Index: E	
ECSTST .....	1.1.8.2
	21.5
EDD (see "deadstart dump")	
EFRDMP .....	1.1.8.4
	20.4
EM dump (see "dumps")	
EMDMP .....	1.1.8.4
	20.5

E4DTAPE .....	1.1.8.4 20.6
EMPRT .....	1.1.8.2 21.5
ENABLE IPRDECK entry .....	20.2 20.9 20.12 20.13 20.14
"enforcer" .....	1.1.7 5.1 14.3
EQPDECK .....	2.8.3.1
EPE .....	1.1.1
error log .....	2.3.11
ESM .....	1.1.8.2 21.7
ESM management .....	1.1.8.2
EXEC .....	1.1.8.4 20.7
executor (see "PLATJ executor")	
express deadstart dump (see "deadstart dump")	
80.6 Index: F	
file dumps .....	2.2.1 2.2.6
file management log .....	1.1.8.1 4.6 22.1
FORMAT .....	1.1 1.1.4 1.1.8.4 20.9
FORMAT (NDS command) .....	2.7.2 2.7.3
formatting disk packs .....	2.7 2.7.2 2.7.3 2.7.4
FORMCMD .....	1.1.8.4 20.8 21.2
FOR1 (see "FRAMAT")	
FRAMAT .....	1.1 1.1.3 1.1.4 1.1.5 1.1.7 1.1.8.4 2.3.5 20.9 30.3.2.3
frame .....	1.1.4
FRAMER .....	1.1.4
FRAMX .....	1.1.4 1.1.8.4

	10.2.2
"lpedit" .....	1.1.9
	8.1.5
IPRDECK .....	20.2
	20.9
	20.12
	20.13
	20.14
80.10 Index: J	
80.11 Index: K	
"K.STDP" .....	1.1.1
	2.2.4
	10.3.1
80.12 Index: L	
"ldr" .....	1.1.9
	1.3.1
	1.3.2.2
	2.3.1
	2.9
	6.4
	11.1
	11.2
	21.12.1
"lessnotes" .....	30.3.1
lesson access bits .....	4.1
	4.2
	4.7.3
	10.1.1.7
	10.2.1
	10.2.2
lesson binary .....	1.1.6
	1.3.2.1
	1.3.3
	5
	5.1
lesson usage data (see "usage data")	
load file (see "resident")	
logical site .....	1.1.9
	14
	14.1
	14.2
	14.3
LURBC .....	1.1.8.5
	10.3.5
80.13 Index: M	
MAS .....	1.1.7
MASJOB .....	1.1.8.4
	20.10
master file .....	1.1.1

	1.1.8.2
	1.1.8.3
	1.1.8.4
	1.1.8.5
	1.1.9
	1.2
	1.3
	1.3.1
	1.3.2
	1.3.2.1
	1.3.2.2
	1.3.3
	1.3.4
	2.3
	2.8.1
	2.3.2
	2.3.2.2
	2.8.3.2
	2.8.4
	2.9
	4.3
	6.2
	7.2
	7.9
	7.10.2
	7.10.13
	21.9
	21.10
	21.11
	21.12
	21.12.1
	21.12.2
	21.12.3
	21.12.4
	21.12.5
	21.12.6
	21.12.7
	21.12.8
	21.12.9
	21.15
master file names .....	1.3.3
master file types .....	1.3.2
	1.3.2.1
	1.3.2.2
	21.12.2
	21.12.3
master file utilities (see "4FUTIL")	
master files to be dumped table .....	7.5
	7.10.5
	7.10.6
MASTER master file .....	1.3.2
	1.3.2.2
	1.3.3
	1.3.4
	2.9
	21.12.3

## 80.13.1 Index: MASTOR

MASTOR .....	1.1
	1.1.1
	1.1.2
	1.1.3
	1.1.4
	1.1.5
	1.1.7
	1.2
	1.3.1
	2.1.3
	2.3.4
	2.3.5
	20.10
	20.11
	21.12.1
	30.3.2.3
MASTOR K-display .....	1.1.1
	2.2.4
MASTORN .....	1.1
	1.1.2
	1.1.7
	2.1.3
MASUSER .....	20.10
MASI (see "MASTOR")	
memory dump (see "dumps")	
MEMPRT .....	1.1.8.2
	21.8
message to entire site (see "logical site")	
messages to users .....	2.2.2
	2.2.3
	14.3

## 80.13.2 Index: MF

MFADD .....	21.12
	21.12.1
MFALTER .....	2.8.1
	21.12.2
MFCREAT .....	1.3.3
	2.8.1
	2.8.2.1
	2.9
	21.9
	21.12.2
	21.12.3
MFDX .....	1.1.8.5
	7.2
	7.5
	7.10.2
	7.10.11
MFLINK .....	9.1
MFLIST .....	21.12
	21.12.4
MFNX .....	1.1.8.4
	1.3.1

	1.3.2.2
	1.3.4
	7.2
	7.10.2
	20.11
"aford" .....	1.1
MFPACK .....	1.1.8.2
	21.9
MFPRIAT .....	11.1
	21.10
	21.11
	21.12.5
MFTCOPY .....	1.1.8.2
	2.9
	21.10
MFTLOAD .....	1.1.8.2
	11.1
	21.11
MFUTIL .....	1.1.8.2
	21.12
missing PLATO files .....	2.8.4
MDDPRT .....	1.1.8.1
	22.4
MDDVAL .....	1.2
MRQ .....	1.1.1
	1.1.2
MST .....	2.7.2
	2.7.3
	2.7.4
multi-mainframe .....	1.1
	1.1.2
	1.1.7
	1.1.8.4
	2.1.3
	20.13
MXX .....	1.1.1
80.14 Index N	
"alog" .....	4.6
NETPRT .....	1.1.8.2
	21.13
network database .....	1.1.8.2
	1.1.9
	3.3
	3.3.1
	3.3.2
network management .....	1.1.8.2
networking (see "inter-system link")	
NJS account file .....	1.1.8.0
	2.3.11
	7.7
	10.1.1
	10.1.1.1
	10.1.1.2
	10.1.1.3
	10.1.1.4



	10.1.1.5
	10.1.1.6
	10.1.1.7
	10.1.1.8
	10.1.1.9
	10.1.1.10
	10.1.1.11
	10.1.2
	10.1.3
	10.1.4
	10.3.1
	10.3.3.1
NJS dayfile (see "system dayfile")	
NJS error log (see "error log")	
"nosalds" .....	1.1.8.3
	1.1.9
	20.15
	21.15
NJS user name (see the specific user name)	
notesfile director .....	1.5
"notesys" .....	5.1
NPRINT .....	1.1.8.1
	22.5

80.15 Index: D

Operations Guide (see "PLATO Operations Guide")	
operations utilities .....	1.1.8.2
"operator" .....	1.1.9
	1.3.2.2
	4.8
	4.10
"opjobs" .....	1.1.9
	10.1.4
	10.1.5
	10.2
	10.2.2
	10.3.7

80.16 Index: P

PA entry in NJS account file .....	10.1.1.2
	10.3.1.2
packing master files (see "disk space management")	
parcel .....	1.1.4
PC entry in NJS account file .....	10.1.1.4
PCD2 .....	40
PJ entry in NJS account file .....	10.1.1.11
PJCAT .....	1.1.8.2
	21.14
PDPRT .....	1.1.3.2
	21.15
PFDEST .....	21.12
	21.12.5
PFIN .....	21.12
	21.12.7
PFOUT .....	21.12

	21.12.4
	21.12.7
	21.12.8
PFUSE .....	21.12
	21.12.9
PI entry in NDS account file .....	10.1.1.5
	10.3.1.2
PIO .....	1.1.3
	1.1.4
80.16.1 Index: PL	
PL entry in NDS account file .....	10.1.1.7
"PLAINS." DSD-command .....	1.2
	8.1
	8.1.1
PLARECV user name .....	1.2
PLASEND user name .....	1.2
PLATEM .....	20.5
	21.2
	21.14
PLATO (see "PLATO executor")	
PLATO. (see " "PLATJ." DSD-command")	
PLATO account .....	1.1.9
	4
	4.1
	4.2
	4.3
	4.4
	4.6
	4.7
	4.8
	4.9
	5.1
PLATO Author Language (see "TUTOR")	
PLATO binary (see "Lesson binary")	
PLATO binary tape .....	3.1.1
PLATO configuration files (see "configuration file")	
PLATO Configuration handbook .....	1
	1.1.9
	1.4.3
	4.5
	21.7
	70.1
80.16.1.1	
PLATJ Courseware Development and Delivery (see PC02)	
PLATJ documentation (see "documentation")	
PLATJ dumps (see "dumps")	
"PLATJ." DSD-command .....	1.1.1
	1.1.2
	1.2
	2.1.3
	2.3.5
	8.1.1
PLATJ executor .....	1.1

	1.1.3
	1.1.4
	1.1.7
	1.1.8.4
	20.7
	20.12
PLATO hotline .....	12
	30
PLATO Installation Guide .....	1
	3.1.1
	3.3.1
	7.4
	13.3
	70.1
PLATO Inter-system Link (see "Inter-system link")	
PLATO master file (see "master file")	
PLATO/NA4 Interface (see "PNI")	
PLATO Operations Guide .....	1
	1.1.9
	1.4.4
	2.3.3
	30.3.2.3
PLATO Support .....	1.4.1
	2.3.4
	8.2
	30.2
	30.3.1
	30.3.2.2
PLATO User's Guide .....	1
	1.5
PLATO utilities (see "utility programs")	
PLATD .....	1.2
PLATOMF user name .....	1.2
	20.6
	21.10
	21.11
	21.12.3
PLATX .....	1.1.8.4
	20.12
PLAT1A (see "PLATO binary tape")	
"PLAUPD." DSD-command .....	1.2
PLA1 (see "PLATO executor")	
PLMPRT .....	1.1.8.1
	22.6
80.16.2 Index: P4	
P4 entry in NDS account file .....	10.1.1.8
P4S .....	1.1.1
PNA .....	1.1.5
"pnet" .....	1.1.8.2
	1.1.9
	3.3
	3.3.1
	3.3.2
	5.1
	21.13

PNI .....	1.1
	1.1.3
	1.1.4
	1.1.5
	1.1.7
	3.2
	3.2.1
	3.2.2
	10.1.1.6
	14.1
	14.3
	20.13
PNI load file (see "resident")	
PNIX .....	1.1.5
	1.1.8.4
	20.13
"pnotes" .....	5.1
PO entry in NDS account file .....	10.1.1.9
port usage data (see "usage data")	
PORTX .....	1.1.8.6
	10.3.2
PPRINT .....	1.1.8.1
	22.7
PR entry in NDS account file .....	10.1.1.3
	10.3.1.2
print access .....	4.3
	4.4
print requests .....	1.1.9
	2.0
print utilities .....	1.1.8.1
printing files (see "print utilities")	
printing file management log .....	1.1.8.1
	4.6
	22.1
printing dumps (see "dumps")	
printing master file directories .....	1.1.8.2
	21.12.5
printing network database .....	1.1.8.2
"prints" .....	1.1.8.1
	1.1.9
	2.6
	22.2
	22.3
	22.4
	22.5
	22.6
	22.7
	22.3
PRINTS user name .....	1.2
problem reporting .....	2.4
	8.2
	30
processing print requests (see "print requests")	
Programming System Report (see "PSR")	
PROUTE .....	1.1.8.4
	20.14
"prtsub" .....	22.7

"prtun" .....	1.2
80.16.3 Index: PS	
PS entry in NDS account file .....	10.1.1.1 10.3.1.2
"psonotes" .....	13.2
PSR .....	2.4 30.2 30.3.1 30.3.2.1 30.3.2.2 30.3.2.3 30.4.1 30.4.2.1 30.4.2.2
PTF .....	9
PU entry in NDS account file .....	10.1.1.10
published courseware (see "courseware")	
publishing account .....	4.2 10.2.1 10.2.2
PX entry in NDS account file .....	10.1.1.0
80.17 Index: Q	
80.18 Index: R	
RAFMON (see "raw account file")	
RAFPBC .....	1.1.8.6 10.3.3.1 10.3.3.2
RAFPDD .....	1.1.8.6 10.3.1
raw account file .....	10.1.3 10.3.1 10.3.2 10.3.3 10.3.3.1 10.3.3.2 10.3.7
rebuilding 2550 load file (see "CCP")	
RECDVAL .....	1.1.8.5 7.10.12
recovering disk space (see "disk space management")	
recovering from disk errors (see "hardware errors")	
recovery integrity tests .....	2.8.2.2 2.8.3.2 2.8.4
RECDVME .....	1.1.8.5 7.9 7.10.13
REQPACK .....	1.1.8.2 21.16
required master files .....	1.1.8.4 1.3.1

	1.3.2.2
	1.3.4
	2.9
	8.1.5
	20.11
reserved lesson list (see "logical site"):	
resident .....	1.1.5
	1.1.9
	1.2
	3.2.1
	20.13
resource management .....	1.1.8.2
	1.1.9
	1.3.2.1
	1.4.3
	14
restricted lesson .....	1.1.9
	5.1
	14.3
restriction lists (see "logical site")	
RETRIEVAL file .....	6.1
retrieval processing .....	1.1.9
	6.4
	6.5
RHP .....	9
	9.1
routing ID .....	6.2
runner program .....	1.1.9
	1.3.2.1
	5
	5.1
	14.1
"runnersys" .....	1.1.9
	5
"runrexec" .....	5
80.19 Index: S	
sending messages (see "messages to users")	
shutdown .....	2.2.1
	2.2.2
	2.2.4
	2.2.7
	2.3.13
"sid" .....	10.1.1.1
signing off users (see "backout")	
"site" .....	1.1.9
	14.2
	14.3
site director .....	14.2
	14.3
site director options (see "logical site")	
slot table .....	7.1
	7.3
	7.10.5
software errors .....	2.4
Software Maintenance (see "PLATO Support")	

sorted billing cycle file ..... 10.3.3  
 10.3.3.2  
 10.3.3.3  
 10.3.4  
 10.3.5

special groups (see also the specific group)  
 1.4  
 1.4.1  
 1.4.2  
 1.4.3  
 1.4.4  
 1.4.5  
 1.4.6

"stats" ..... 1.1.9  
 2.3.13  
 5

"stats1" ..... 5

"STDP" K-display entry (see "K.STDP")  
 1.1.8.4

SUBMIT ..... 20.15

submitting jobs ..... 1.1.1  
 1.1.2  
 1.1.3.4  
 1.2  
 2.1.3  
 2.3.4  
 4.6  
 20.15

subscriptions ..... 4.1  
 4.2  
 4.6  
 4.7.3  
 10.2.1  
 10.2.2

80.19.1 Index: SY

SYS user name ..... 1.2

"sysalids" ..... 1.1.9

"sysin" ..... 2.4  
 30.3.1  
 60

"sysatr" ..... 1.1.9  
 2.1.4  
 2.3.1  
 2.3.2  
 2.3.4  
 2.3.6

"sysopts" ..... 1.1.2  
 1.1.3  
 1.1.4  
 1.1.6  
 1.1.9  
 2.2.2  
 2.2.3

system account file (see "NDS account file")

System Consultants .....	1.1.9
	1.4.6
	12
	13
System Controllers .....	1.1.9
	1.4.3
	2.7
	4
	4.6
	14.3
	22.1
system dayfile .....	2.3.11
	7.7
system error log (see "error log")	
system ID (see "sid")	
system shutdown (see "shutdown")	
SYSTEMX user name .....	1.2
	10.1.2
	10.1.3
"s0arcn" .....	6.1
"s0calutil" .....	1.1.9
	1.4
	40.2
"s0cpustat" .....	5.1
"s0files" .....	7.4
"s0netprt" .....	21.13
"s0notrun" .....	5.1
"s0pnetrn" .....	5.1
"s0pnllf" .....	1.1.9
	3.2.1
"s0crhp" .....	5.1
"s0sysmsj" .....	2.4
	60

80.20 Index: T

temporary master files .....	1.3.1
	1.3.2.2
	2.3.1
	2.9
TERM-consult .....	1.4.6
	13.1
terminal load file (see "resident")	
terminal resident (see "resident")	
testing utilities (see "utility programs")	
TPRINT .....	1.1.3.1
	22.3
"transfer" .....	1.1.9
	1.3.2.2
	2.3.1
	2.9
	11.1
TRANSMIT .....	4.2
	4.9
	10.2.1
	10.2.2
TUTOR .....	1.1.3



