[illegible] CONFIGURATION HANDBOOK

PLATO LESSON DELIVERY AND AUTHORING 1

RELEASE 34.2 - NOS 2.3 LEVEL 617

RELEASE 34.2 - NOS 2.4.1 LEVEL 630

SMD130935

Table of Contents

1 Introduction

This handbook describes the process of configuring the PLATO application. Other documentation for the PLATO application can be found in the:

PLATO Installation Guide
PLATO Operations Guide
PLATO User's Guide

This handbook is divided into two major sections.

- a description of configuration parameters and procedures which are common to all systems

- several appendices, each of which describes configuration parameters and procedures for features of the PLATO application which are only used on a few systems.

2 Deadstart File

This section describes only those things which are unique to deadstart files used with the PLATO application. In all other respects, NOS standards, as described in the NOS System Analysis Handbook, are followed.

2.2 EQPDECKS

All EQPDECKs must contain the following entries to define the mass storage device where the account log, system dayfile, error log, and maintenance log are placed:

```
    ACCOUNT
    DAYFILE
    ERRLOG
    MAINLOG
```

These are necessary to prevent the system from defaulting to the first mass storage device, which will always be the Extended Memory (EM) device for systems running the PLATO application. These system logs will interfere with the PLATO application if they are placed in EM. This will result in error messages from PP/MXX such as "local file error" or "track allocation error" when attempting to load PLATO or in a reduced amount of EM being assigned to the MASTOR control point.

Note also that allowing other uses of EM may also interfere with the PLATO application. These include using EM as an alternate system residence device, a temporary file device, a rollout file device, a checkpoint file device or a PF device. If User Extended Memory (UEM) is allocated, the "rex" and "fix" entries in the PLATO configuration file may have to be adjusted.

This is due to the fact that the PLATO application reserves all of its needed EM tracks at load time, and also because the EM tracks must be in sequential order. If NOS is making use of EM for some reason, there may be reserved tracks between the first track the PLATO application requests and reserves, and the last track it requests. Generally this will result in PP/MXX issuing a "track allocation error" message and then aborting. It may be possible to get around this situation if use of EM by other than the PLATO application is necessary by using the MSAL console command to restrict further usage of EM, and then waiting for current users to leave (e.g., allow rolled out jobs to roll in), before trying to load or reload the PLATO application.

CYBER 170-600 series mainframes require only the above EQPDECK entries and the DE EQPDECK entry used to describe the EM device for the proper operation of the PLATO application.

The following describes EQPDECK entries for devices commonly used on PLATO systems running on CYBER 170-700 mainframes only. These are in addition to the entries described above.

These parameters are used:

ord = One- to three-digit octal Equipment Status Table (EST) ordinal of equipment

st = (status) ON or OFF (usually ON)

eq = (equipment) Controller number (may vary with each system, the most commonly used number is shown)

un = Unit number (for most entries, this is not applicable, so use 0)

ch = One- or two-digit octal number of the channel to which the equipment is connected

a. SHARED LOW-SPEED PORT / DDP

```
EQord=D1,ST=st,EQ=5,UN=un,CH=ch.
```

This entry is optional, but is normally used to improve performance. If the D1 entry is not used, the PLATO configuration file parameter "ncmb" must be greater than zero. This is to allow the PLATO disk driver (PMS) to perform disk transfers through a Central Memory (CM) buffer instead of through the low-speed/DDP port.

b. ESM SIDE-DOOR PORT

```
EQord=SP,ST=st,EQ=1,UN=un,CH=ch.
```

This entry is used only when using Extended Semiconductor Memory (ESM) as the EM device.

Prior to NOS 2.1, ESM error monitoring on PLATO systems was only performed by program ESM. Now, there are two options for ESM error monitoring.

1. Use the SP EST entry as described above. This identifies the ESM maintenance channel and allows program ESM to be used for error monitoring.

2. Use the MC parameter on the ESM EST entry (DE). This allows the operating system to perform error monitoring.

Refer to the NOS V2 System Analysis Handbook for more information. Also, read the section on "ESM Management" in this Handbook before choosing the method of error monitoring to be used on your system.

2.3 LIBDECKs

The following is a list of LIBDECK entries required for the PLATO application. See the PLATO Operations Guide for more information on the following procedures. These entries are delivered with the initial release of the PLATO application and placed in file LIBDIR under the PLATO user name by the installation procedures.

```
*cm       pp/mas
*cm       pp/4qf,4qg,4qh,4qi,4qj,4qk                           (mrq)
*cm       pp/4ql,4qm,4qn,4qo,4qp,4qq                           (mrq)
*cm       pp/4qr,4qs,4qt                                       (mrq)
*cm       pp/4qu          (mrq - 700 series mainframes only)
*cm       pp/4pa,4pb                                           (pas)


*proc     versx                               (version load)
*proc     configx                             (get config)
*proc     mfnx                                (attach master files)
*proc     platx                               (load plato executor)
*proc     framx                               (load framat)
*proc     pnlx                                (load pnl)
*proc     condx                               (load condensor)
*proc     formcmd                             (format cm dump)
*proc     emdtape                             (dump em for plato)
*proc     copyod                              (copy plato dump)
*proc     dumpprt                             (print cm dump)
*proc     emprt                               (print em dump)
*proc     pdcat                               (catalog plato dump)
*proc     mftload                             (copy mf to disk)
*proc     mftcopy                             (copy mf to tape)
*proc     mfpack                              (mf pack utility)
*proc     bkstart                             (backups)
*proc     backdmp                             (backups)
*proc     mfix                                (backups)
*proc     recoval                             (recover master files)
*proc     recovmf                             (recover master file)
```

2.4 IPRDECKs

The following are required IPRDECK entries for systems using the PLATO application.

1. An "ENABLE,SCP." entry must be present to enable the system control point facility if the PLATO ASCII network is to be used on your system.

2. An "ENABLE,PLA,cp." entry must be present in the IPRDECK to specify "cp" as the control point where MASTOR will be loaded by the operating system when the "PLATO." DSD command is entered at the computer console. The PLATO application occupies this control point and the following four control points. To minimize storage moves of the programs which make up the PLATO application, it is recommended that you use the following control point configuration:

   1. IAF (must be at control point 1)
   2. NAM (controlled by ENABLE,NAM,cp. entry)
   3. MAS1 (controlled by ENABLE,PLA,cp. entry)
   4. PLA1 (controlled by submit file PLATODD)
   5. FOR1 (controlled by submit file PLATODD)
   6. PNI1 (controlled by submit file PLATODD)
   7. COA1 (controlled by submit file PLATODD)

3. A "DELAY" entry must be used to change the default values of the CPU recall time parameters.

   All sites should set the "MX" parameter on the "DELAY" entry to 10.

   The following recommendations should be considered when setting the value of the "CR" parameter on the "DELAY" entry.

   - Sites for which batch processing time is not a critical and/or scarce resource should use a setting of 153 milliseconds.

   - Single-CPU sites which use NAM for part or all of their communications should set their CPU recall period at 303 milliseconds (NOS standard) if they have 50 or fewer PNI users; dual-CPU sites should do so with PNI loads of up to 60 users. Sites with larger PNI loads should use a setting of 153 milliseconds for the CPU recall period.

     Sites which deviate from this recommendation may estimate their mean PNI echo time, using the following approximation:

	CPU Recall Period		
	153	308	453
1-CPU	155	197	220

2-CPU	145	185	221

- Sites with less than 150 users which use only CIU communications should set their CPU recall period to 30B milliseconds (NOS standard) or higher, up to 45B. If a site has more than 150 users, or if a setting greater than 45B milliseconds is desired, it may be necessary to experiment to find the highest acceptable setting.

4. Systems which are using ESM in ESM mode (as opposed to ECS mode) should add "X.ESM(NK)." as a DSD entry to ensure that this job runs as soon as the operating system is loaded to reload the relocation memory.

3 The PLATO Configuration File

The PLATO configuration ("config") file contains software parameters which may be unique on each system. It allows the local site personnel control over resources which have considerable impact on performance.

The "config" file is a text record on the deadstart tape. The first line in the "config" file must be the name of the record. The PLATO application checks that the first six characters of the "config" file are the required letters "config". However, multiple "config" files may exist, named "config1", "config2", etc.

Comments may be added as lines starting with "*". All lines that are not entirely comments must follow this format:

```
    parameter=value.  comment
```

where:

```
    parameter = a valid keyword described in the
                following sections

    value     = an appropriate setting for your system

    comment   = any additional comment you wish to make
```

3.1 Types of Keywords

The defined keywords can be divided into several different "types" dependent on what is controlled by that keyword.

The following keywords control parameters affecting the PLATO Application as a whole and its interface to the operating system.

Application load/drop control

    nampd

Security

    passw
    secur
    syot

Identification

    famly
    prtun
    subun

Extended memory allocation

    fix
    rax

Application interface to network

    nam
    namto

Background PLATO batch jobs

    njob

The following keywords control parameters affecting the PLATO Application as a whole internally.

Application availability

    instl

Internal security

    ptlim
    sysac

PLATO system identification

    rid

sld

Central Processor speed

cpspd

PLATO disk system resources

ncmo
ndsus
npms

Internal extended memory allocation

Parameters dependent on memory size

cblth
cdisk
fastl
forml
lesns
nparc

Parameters dependent on number of users

jbnks
nlob
nnlsl
nsite
nlslt
quesz
users

Author deletion parameters

edel1
edel2
edel3
sysdl

EM manager parameters

emgr1
emgr2
emgr3
emgr4

PLATO feature availability

cmp
confr
cstat
astat
acond
ncond
pdcnk
pnet

PLATO account parameters

nacnt
nalog

Runner program parameters

nrunr

Data formats

datef
timef

Background PLATO batch job resources

bgecs
bgpct

PLATO Inter-system Link parameters

netms

See the "Keyword Definitions" section for an alphabetic list of keywords and their definitions.

Keywords for the CIU network and for multi-mainframe systems are described in appendices later in this Handbook.

3.2 Keyword Definitions

The description of each configuration file entry includes the following information:

keyword: short definition of keyword

complete definition of keyword

restrictions on the value assigned to this keyword (if any apply)

usual values for different configurations

default value if this keyword is omitted from the configuration file

The values assigned to keywords may be:

a positive decimal integer
a positive octal integer (by specifying "o", i.e., 10o)
the string ON
the string OFF
alphanumeric string of one to three characters
alphanumeric string of one to seven characters

When a parameter is dependent on memory size, this means the Extended Memory actually assigned to the PLATO application, not the entire physical EM size.

The recommended values for all keywords should be considered as guidelines only, particularly those which are dependent on memory size. All values may have to be adjusted for the maximum performance of the system.

3.2.1 Keywords: A - B

BGECS: background batch job extended memory

The value of "bgecs" is the number of 1000b-word blocks of extended memory within the extended memory field length of the application to be reserved for batch jobs (i.e., PLATO print jobs, PF commands, etc.).

For systems with ESM, this should be a multiple of 10b or it will be rounded up to the nearest multiple of 10b.

For single mainframe systems, a value of 20b is usually sufficient.

Default value: 20b

BGPCT: excess processing time (background) percentage

The value of "bgpct" is used to specify what percentage of excess processing time should be made available to users running with the -backgnd- TUTOR command in effect. The remainder of the processing time, if any, will be made available for batch jobs.

The value of "bgpct" must be an integer greater than 0 but less than 100. The greater the value of "bgpct" the more excess processing time is given to users of lessons with -backgnd- commands.

On systems with light batch loads, it may be best to give -backgnd- users priority (i.e., set "bgpct" to a high value) to minimize CPU idle time.

Default value: 50

3.2.2 Keywords: C

CBLTH: CONDENSOR source buffer length

As the CONDENSOR source buffer is made larger, more EM is required for each CONDENSOR, but less disk activity is required since more blocks of the source file can be read into the buffer with a single disk access.

The value of "cblth" must be a positive integer less then or equal to 21 and is in units of the TUTOR block length.

Default value:
 3 for systems with less than 1000K of EM
21 for systems with 1000K or more of EM

CDISK: CONDENSOR overlays on disk

If the value of "cdisk" is ON, the main CONDENSOR overlays (PLATO Author Language, each level of the Micro PLATO Language and Central Micro PLATO) are loaded from disk rather than from EM. This will save a large amount of EM, but condense times for Micro PLATO lessons will be greatly increased.

If the value of "cdisk" is ON, the value of the "ncond" keyword will be set to 1.

The usual value of "cdisk" is:

a. ON for systems with less than 1000K of EM
b. OFF for systems with 1000K or more of EM

Default value: OFF

CMP: Central Micro PLATO availability

The value of "cmp" determines the availability of the Central Micro PLATO (CMP) Executor.

If the value of "cmp" is OFF, the CMP overlay in the CONDENSOR is not loaded into EM, saving memory. If the value of "cdisk" is ON, there will be no savings and this keyword will simply make execution of CMP lessons impossible.

The usual value of "cmp" is OFF.

Default value: OFF

CONFR: TERM-confer (Teleconferencing) availability

The value of "confr" determines the availability of the Teleconferencing feature, including TERM-confer and lesson "s(confer". This control is necessary for countries in which this feature would be in violation of laws governing communications.

The usual value of "confr" is ON.

Default value: ON

CPSPD: Central Processor speed

Setting this keyword to the correct value is necessary to insure that lessons execute the same on all machines, regardless of the actual machine speed. The following figures are rough estimates only. Use lesson "s0cpspd" to get a more accurate estimate.

The value of "cpspd" should be set as follows:

Cyber 73	cpspd=1000.
Cyber 171	cpspd=620.
Cyber 172	cpspd=900.
Cyber 173	cpspd=1500.
Cyber 174	cpspd=1500.
Cyber 175	cpspd=4000.
Cyber 720	cpspd=900.
Cyber 730	cpspd=1500.
Cyber 810	cpspd=1400.
Cyber 815	cpspd=1400.
Cyber 825	cpspd=1700.
Cyber 830	cpspd=1700.
Cyber 835	cpspd=1700.
Cyber 840	cpspd=3000.
Cyber 845	cpspd=3000.
Cyber 850	cpspd=4200.
Cyber 855	cpspd=4200.

Default value: 1400

CSTAT: CONDENSOR statistics

The value of "cstat" determines the availability of an EM buffer for the collection of CONDENSOR statistics. If a buffer is available, lesson "system1" may be used to collect and view TUTOR command condensing statistics.

The value of "cstat" is usually OFF.

Default value: OFF

3.2.3 Keywords: D - E

DATEF: date format

The value of "datef" controls the format of the data produced by the -cdate- TUTOR command. This command is used in many system lessons to display the current date.

The value of "datef" must be one of the integers 1, 2 or 3. The format produced by each of these values is:

a. datef=1. mm/dd/yy
b. datef=2. dd/mm/yy
c. datef=3. yy/mm/dd

The usual value of "datef" depends on the local site.

Default value: 1

EDELL1: EM deletion pass 1

When a student needs EM, and there is not enough available, authors using more than "edel1" words of EM may be backed out of their current lesson.

See the section on "Controlling EM deletion" for more information.

The usual value of "edel1" is 8000.

Default value: 8000

EDEL2: EM deletion pass 2

If not enough EM is obtained after pass 1, a second pass is made, backing out authors using more than "edel2" words of EM.

See the section on "Controlling EM deletion" for more information.

To be effective, "edel2" must be less than "edel1".

The usual value of "edel2" is 5500.

Default value: 5500

EDEL3: EM deletion pass 3

If enough EM is not obtained by the first two passes, a third pass is made backing out authors using more than "edel3" words of EM. No more passes are made, even if pass 3 is not successful.

See the section on "Controlling EM deletion" for more information.

To be effective, "edel3" must be less than "edel2". The minimum allowable value is 1500.

The usual value of "edel3" is 3000.

Default value: 3000

EMGR1: maximum short EM request

See the section on "EM Manager parameters" for more information.

The usual value of "emgr1" is 2000.

Default value: 2000

EMGR2: minimum EM free for long request

See the section on "EM Manager parameters" for more information.

The usual value of "emgr2" is 10000.

Default value: 10000

EMGR3: EM available scan threshold

See the section on "EM Manager parameters" for more information.

Default value:
40000 for systems with less than 1000K of EM
80000 for systems with 1000K or more of EM

EMGR4: normal free EM desired

See the section on "EM Manager parameters" for more information.

Default value:
30000 for systems with less than 1000K of EM
50000 for systems with 1000K or more of EM

ESTAT: execution statistics

The value of "estat" determines the availability of an EM buffer for the collection of execution statistics. If a buffer is available, lesson "systeml" may be used to collect and view TUTOR command execution statistics.

The value of "estat" is usually OFF.

Default value: OFF

3.2.4 Keywords: F

FAMLY: NOS family

The value of "famly" is the NOS family to be used by the PLATO application load procedures and submitted jobs. Any user-submitted jobs will use this family unless it is overridden by a family name stored in the user's signon record. More information regarding NOS families may be found in the NOS System Analysis Handbook.

The value of "famly" must be a one to seven character alphanumeric string. If the form "famly=." is used, the default NOS family will be used.

Default value: default NOS family

FASTL: fast output buffer length

The value of "fastl" determines the size of the FRAMAT fast output buffer, which contains output with the highest priority. Very short output strings such as key echoes are kept in this buffer.

The system will issue a dayfile message when this buffer is too short so that it may be lengthened. If this buffer length must be increased much beyond the usual range, it may indicate that a terminal is sending in too many keys.

Default value:
 500 for systems with less than 750K of EM
 1000 for systems with 750K to 1500K of EM
 2000 for systems with more than 1500K of EM

FLX: extended memory field length (FL)

The value of "flx" determines the amount of extended memory to be reserved for the PLATO application. The actual EM FL is set to (value of "flx") * 1000b.

The PLATO application is usually given all extended memory starting at "rax" and continuing through the entire physical EM available. Typical values of physical EM sizes are:

	ECS-1	ECS-2/ESM/UEM
500K system	1730b	2000b
750K system	2704b	3000b
1000K system	3660b	4000b
1500K system	5610b	6000b
2000K system	7540b	10000b

In general, the method to determine "flx" is to subtract "rax" from the extended memory field lengths shown above. For example, if you have 750K of UEM and set "rax" to 30b, then "flx" should be set to 3000b - 30b or 2750b.

If the value of "flx" is set to 0, all available contiguous EM beginning at "rax" will be reserved.

Note: NOS supports only up to 2000K words of EM. If your system has more than 2000K words of EM and you are reserving it for PLATO, you must set "flx" to do so. Setting the value of "flx" to 0 will reserve only the first 2000K - "rax" words.

Default value: 0

FORML: slow output buffer length

The value of "forml" determines the length of the FRAMAT slow output buffer, which contains output with the lowest priority. Medium to long output strings are kept in this buffer.

The system will issue a dayfile message when this buffer is too short so that it may be lengthened. If this buffer length must be increased much beyond the usual range, it may indicate that a terminal is sending in too many keys.

Default value:
 960 for systems with less than 750K of EM
1920 for systems with 750K to 1500K of EM
2400 for systems with 1500K or more of EM

3.2.5 Keywords: G - M

INSTL: installation mode

The value of "instl" determines the operating mode of the PLATO application, either installation mode or normal mode. Under installation mode, only certain system groups are allowed to sign in.

The value of "instl" must be either 0 for normal operation or 1 for installation mode.

The usual value of "instl" is 0. During PLATO application installation, it is set to 1 whenever the PLATO application is loaded via the "PLAUPD." or "PLAINS." DSD-commands.

Default value: 0

JBNKS: auto-break judge buffers

The value of "jbnks" determines the number of buffers to be defined for the system to store information used for judging student responses when the user must auto-break at the end of a time-slice.

The value of "jbnks" must be a postive integer exactly zero or greater than 4 but less than or equal to 48.

The usual value of "jbnks" is determined by the number of users of the PLATO application. There should be one buffer defined for every 16 users signed on at once. If the value of "jbnks" is set to 0, the system will change it to: (value of "users")/16 + 4.

Default value: 0

LESNS: lesson buffer entries

The value of "lesns" determines the number of entries which can be made in the EM lesson buffer. These entries are the executable binaries of lessons, storage, commons and other subfiles, router variables, local variables, student banks, etc.

Default value:
 500 for systems with less than 750K of EM
 1000 for systems with 750K to 1500K of EM
 2000 for systems with more than 1500K of EM

MCOND: minimum number of CONDENSORs

The value of "mcond" determines the minimum number of CONDENSORs which will be left at control points when the condense queue is lower than the minimum at which the PLATO application will load additional CONDENSORs. This is most useful in multi-mainframe systems and systems which are dedicated to running PLATO with a large number of courseware authors where there is a demand for faster response to condense requests.

This keyword has no effect if "ncond" is set to 1.

The usual value for "mcond" is 1.

Default value: 1

3.2.6 Keywords: NA - NB

NACNT: highest PLATO account number

The value of "nacnt" determines the largest integer which may be assigned as an account number to any PLATO account in your system. Note that this is NOT the same as the number of accounts existing on the system since there may be account numbers which are not currently assigned to an account. However, "nacnt" does determine the maximum number of accounts which may be created.

The value of "nacnt" must be less than or equal to 500.

There is no usual value for "nacnt" since it depends on the policies of the local site.

Default value: 200

NALOG: number of account file management logs

The value of "nalog" determines the number of account file management logs ("acclog0", "acclog1", etc.), on the system.

Contact PLATO Support before changing this value.

The usual value of "nalog" is 4.

Default value: 4

NAM: PLATO-NAM Interface (PNI) availability

The value of "nam" is set to the number of copies of NAM and PNI running on a system when the ASCII network is in use for PLATO.

On single mainframe systems, the value of "nam" can be the integers 0 or 1.

On systems which use the ASCII network for PLATO, the value of "nam" must be 1.

Default value: 1

NAMPD: PLATO drop time

The value of "nampd" is the time in seconds between the time the last user of PLATO through the ASCII network signs out and the time PNI drops the PLATO application. This can be used to release the CM in use by the PLATO application when it is not being used.

If the value of "nampd" is set to 0, PNI will never drop PLATO. If the site wishes PLATO to load and drop as users sign on and off, a non-zero value should be used. Normally, it takes the PLATO application two minutes to initialize, so a site should not specify this value such that the application drops too soon. A suggested value might be 2 minutes (120 seconds), but it is highly dependent on the user sign-on activity at the site.

This keyword should never be used on systems which use both the ASCII and CIU networks.

Default value: 0

NAMTO: PNI log-out timeout

The value of "namto" determines the length of time between the time a user presses SHIFT-STOP to return to the "Press NEXT to Begin" display and the time the user is logged out of NAM.

PNI checks for this timeout only once every 30 seconds, so any number specified may actually have a range of plus or minus 30 seconds.

If the value of "namto" is set to 0, PNI will immediately log out a user when SHIFT-STOP is pressed to return to the "Press NEXT to Begin" display.

There is no usual value for "namto" since it depends on the policies of the local site.

Default value: 120

3.2.6.1 Keywords: NC - NM

NCMB: number of PMS CM buffers

The value of "ncmb" determines the number of CM paths for disk-to-EM transfers when the PMS low-speed port/DDP is busy or not defined in the NOS EQPDECK.

The value of "ncmb" must be an integer greater than or equal to 0 but less than or equal to "npms".

Systems on 800-series mainframes must set this value to 0 or use the default value.

Systems on 700-series mainframes with a "D1" equipment defined in the EQPDECK should set "ncmb" to "npms"-1.

Systems on 700-series mainframes with no "D1" equipment defined in the EQPDECK should set "ncmb" equal to "npms".

See the section on "Settng up your disk system" for more information.

Default value: 0

NCOND: number of CONDENSORs

The value of "ncond" determines the maximum number of CONDENSORs which are allowed to run. If more than one CONDENSOR is allowed, additional ones will be started automatically when the condense queue builds up. The additional CONDENSORs will be dropped when the condense queue goes below the threshold where additional CONDENSORs are loaded unless the "mcond" keyword is used to allow the additional CONDENSORs to remain running.

The value of "ncond" must be an integer greater than 0 but less than or equal to 3. This keyword has no effect if the value of "cdisk" is ON. In this case, the system will support only one CONDENSOR.

The usual value of "ncond" is 1.

Default value: 1

NDSUS: maximum number of master files

The value of "ndsus" determines the maximum number of PLATO master files which may be attached at one time.

The value of "ndsus" must be a positive integer less than 116.

There is no usual value for "ndsus" since it is dependent on the amount of disk space desired for the local site.

Default value: 12

NETMS: network system table size

The value of "netms" determines the length of the network system table contained in common "link" of file "sysfile". This common must be at least (1 + 10 * "netms") words long. See the "Network Management" section for more information before changing the value of this keyword.

The value of "netms" should be set to the number of systems in the network for which an "authors" database or a PLATO inter-system link exists.

There is no usual value for "netms" since it is dependent on the local site's network configuration.

Default value: 10

NIOB: number of subfile I/O buffers

The value of "niob" determines the number of "subfile I/O buffers" which are used when reading commons, charsets, linesets, microtables, access lists, dataset records, nameset records or TUTOR file blocks from disk.

The usual value of "niob" is:
a. 4 for systems with less than 64 users
b. 8 for systems with 64 - 127 users
c. 12 for systems with 128 - 255 users
d. 16 for systems with more than 255 users

When the value of "niob" is set to zero, the system will automatically set it to the usual values given above.

Default value: 0

NJOB: number of PLATO batch jobs

The value of "njob" determines the maximum number of jobs submitted through PLATO which can be running at the same time. This includes the PLATO system control points.

The usual value of "njob" is 64.

Default value: 64

3.2.6.2 Keywords: NN - NR

NN1SI: number of NAM (PNI) sites

The value of "nn1si" determines the number of PLATO sites (groups of 32 terminals) for use with the ASCII network.

The value of "nn1si" must be an integer greater than or equal to 0 but less than or equal to "nsite".

There is no usual value for "nn1si" since it is dependent on the number of users on the system at the same time.

Default value: 2

NPARC: number of output parcels

The value of "nparc" determines the number of buffers of formatted output which are waiting to be sent to the network.

Default value:

```
     500 for systems with less than 750K of EM
     750 for systems with 750K - 1500K of EM
    1000 for systems with more than 1500K of EM
```

NPMS: number of copies of PMS

The value of "npms" determines the number of copies of the PLATO disk driver PP, PMS, to be run. Using more than one copy of PMS will improve disk performance, but keep in mind that each additional copy requires another dedicated PP while the application is active. The number of disk channels available and the disk drive configuration on the available channels will also affect the number of PPs which will optimize performance. See the section on "Setting up your disk system" for more information.

The value of "npms" must be 1, 2, or 4.

The usual value for "npms" is 1.

Default value: 1

NRUNR: number of runner stations

The value of "nrunr" determines the maximum number of runner programs which may be running at the same time. See the PLATO Operations Guide for more information on runner programs.

The value of "nrunr" must be at least 3 greater than the number of runners which must be active continuously throughout the day. Each additional runner station will cost about 500 words of EM. Changing this value will affect the station numbers which runner stations occupy and could affect the logical site assignments of these and other stations. See the PLATO Operations Guide and the "EM management" section of this Handbook for more information on logical sites.

The usual value of "nrunr" is 10, but the local site may add more runner stations at their option.

Default value: 10

3.2.6.3 Keywords: NS - NZ

NSITE: number of physical sites

The value of "nsite" determines the total number of PLATO sites (groups of 32 terminals) in the network.

The value of "nsite" should be one greater than the total number of sites which will be used by real users since runner programs occupy the highest site.

On most systems, where the ASCII (PNI) network is the only communications network in use, "nsite" should be set to one greater than "nnlsi".

There is no usual value for "nsite" since it is dependent on the number of users on the system at the same time.

Default value: 3

N1SIT: first NAM (PNI) site

The value of "n1sit" determines the number of the first PLATO site (group of 32 terminals) assigned to the ASCII network.

On most systems, where the ASCII (PNI) network is the only communications network in use, "n1sit" should be set to 0.

Default value: 0

3.2.7 Keywords: O - Q

PASSW: password required for system lessons

The value of "passw" determines if NOS user name passwords are to be required in jobs submitted by system lessons.

If the value of "passw" is OFF, system lessons may use the -submitm- and -submitx- commands without specifying a password for the NOS user name to be used.

Default value: OFF

PDCHK: master file (pack) directory verify

The value of "pdchk" determines if a verification of each master file directory written to disk is to be done.

When a master file directory is written to disk after a PLATO file is created or destroyed, the directory may be read back into an EM buffer and verified against the copy of the directory which was written to disk. This may detect hardware problems in disks or low-speed port/ DDP. Setting the value of this keyword to OFF will save EM and disk accesses, but will reduce reliability.

The usual value of "pdchk" is ON.

Default value: ON

PNET: network database (lesson "pnet") lockout

The value of "pnet" determines if a check is to be made to insure that the network configuration for each terminal has been defined each time a user signs in.

If the value of "pnet" is ON, when a user attempts to sign in on a terminal whose network configuration has not been defined in the network database maintained through lesson "pnet", access will not be allowed.

The usual value of "pnet" if OFF.

Default value: OFF

PRTUN: NOS user number for print jobs

The value of "prtun" specifies the NOS user number to be used when submitting central system print requests through lesson "prints".

The value of "prtun" must be a one- to seven-character alphanumeric string which is a legal NOS user name.

The usual value of "prtun" is "prints".

Default value: "prints"

PTLIM: password time limit

The value of "ptlim" determines the number of days until a user is prompted to change their PLATO signon password when they attempt to sign on.

There is no usual value for "ptlim" since it is dependent on local site policies.

Default value: 60

QUESZ: backgnd, auto-break queue size

The value of "quesz" determines the length of various queues used by the PLATO executor, such as the queue of users waiting for -backgnd- execution time or of users waiting for another time-slice.

The value of "quesz" must be an integral power of 2.

The usual value of "quesz" is:
 a. 64 for systems with less than 64 users
 b. 128 for systems with 64 - 127 users
 c. 256 for systems with 128 - 255 users
 d. 512 for systems with more than 255 users

When the value of "quesz" is set to zero, the system will automatically set it to the usual values given above.

Default value: 0

3.2.8 Keywords: R

RAX: extended memory reference address (RA)

The value of "rax" determines the PLATO application extended memory reference address. The actual EM RA is set to (value of "rax") * 1000b.

Extended memory is allocated in different sized blocks depending on whether you are using ECS/UEM or ESM. ECS and UEM are allocated in blocks of 1000b words while ESM is allocated in blocks of 10000b words. This determines the minimum value of "rax" which may be used to give the PLATO application the most EM possible.

 a. For systems using ECS or UEM, the minimum value of "rax" will be 11b.

 b. For systems using ESM, the minimum value of "rax" will be 20b.

If the value of "rax" is not a multiple of 10b on a system using ESM, it will be rounded up.

If the value of "rax" is set to 0, the value of "rax" used will be automatically determined by the system. This may cause the PLATO system to have different amounts of EM allocated each time it is loaded. If you wish to leave some EM for other jobs, you will have to set a specific value for "rax". See the "fix" keyword for more information.

Default value: 0

RID: routing identifier

The value of "rid" specifies a system identifier which is used by features such as lesson "authors" and the PLATO inter-system link.

The value of "rid" is a three-character alphanumeric string established for each system by Control Data at installation time. You must use this pre-determined value and must not change it once it has been set. The value of "rid" must be unique across all PLATO systems. Changing the value of "rid" after installation can cause features to work incorrectly.

Default value: 0

3.2.9 Keywords: S

SECUR: application security level

The value of "secur" determines the availability of features which can be used to inspect central memory and extended memory of running jobs.

The value of "secur" is either ON or OFF depending on the desired security level. When the value of "secur" is ON, system lessons are prevented from accessing NOS information. For example, no console displays may be seen via lesson "console".

The usual value of "secur" is OFF.

Default value: OFF

SID: system identifier

The value of "sid" specifies the name of the PLATO system for display purposes. It is used in features such as the "zsystem" reserved word, lesson "authors", the signon display and the PLATO inter-system link.

The value of "sid" must be a one- to seven-character alphanumeric string.

Default value: "cdc"

SUBUN: submit user name

The value of "subun" specifies the NOS user name to be used to submit the PLATO load procedures.

The value of "subun" must be a one- to seven-character alphanumeric string which is a legal NOS user name. The submit file PLATOJ must be in the permanent file catalog of this user name.

The usual value of "suoun" is "sys".

Default value: "sys"

SYOT: system origin permission

The value of "syot" determines who can submit system origin jobs from a PLATO application terminal.

The value of "syot" must be one of the following.

0 = No system origin jobs may be submitted except jobs used to load PLATO control points.

1 = System origin jobs may be submitted by any system lesson.

2 = System origin jobs may be submitted by any system lesson or any user validated for it (CSOJ bit set in NOS validation file for the user name to be used for the submit).

The value of "syot" must be greater than 0 on single mainframe systems so that utilities such as "ldr" will work correctly.

SYSAC: system user access

The value of "sysac" determines if remote system-level users are allowed access to the PLATO system.

The value of "sysac" is either ON or OFF, depending on whether groups "s" and "coserv" will be allowed to sign on to the system. These groups are owned by Control Data and are used to perform system software and courseware maintenance.

The usual value of "sysac" is ON.

SYSDL: system lesson deletion

The value of "sysdl" determines if the system may back authors out of specific system lessons to obtain memory for students.

The value of "sysdl" is ON if the system is allowed to back authors out of system lessons. Author deletion must also be enabled in lesson "site". It is recommended that this keyword be used only on systems which have a high student usage and very little development work by authors. Use of this feature can cause work being done by authors to be lost when they are backed out.

See the section on "Controlling EM deletion" for more information.

The usual value of "sysdl" is OFF.

Default value: OFF

3.2.10 Keywords: T - Z

TIMEF: time format

The value of "timef" controls the format of the data produced by the -ctime- TUTOR command. This command is used in many system lessons to display the current time.

The value of "timef" must be one of the integers 12 or 24. The format produced by each of these values is:

```
a.  timef=12.   display time in 12 hour format
b.  timef=24.   display time in 24 hour format
```

The usual value of "timef" depends on the local site.

Default value: 12

USERS: maximum number of users

The value of "users" determines the maximum number of users who may be signed on at the same time.

If the value of "users" is set to 0, the system will change this to 32 * (value of "nsite"). Sites may want to reduce this number to save memory since not all terminals will be signed in at the same time.

There is no usual value for "users" since it is dependent on local system needs.

Default value: 0

4 System Lesson Parameters

Most system lesson configuration parameters are controlled via lesson "ipedit". The following is a list of parameters which may be changed through options in this lesson.

Prime-Time table
This table defines the value of the "ptime" reserved word. It is intended to show the hours when the system is fully operational as scheduled.

Services-available time table
This table determines when the system is fully staffed and operators and consultants are available. Outside of these hours, a message will appear on the signon display stating that services personnel are unavailable.

Time zone
This option allows changing the time zone identifier which appears along with the current time in several system displays.

"Welcome to PLATO" message
This option allows changing a message line on the signon display.

PNET "Lock Out" message
The option is used to edit the message a user sees when not allowed to sign on if the terminal he is using is not defined in the network database. This will only appear if the "pnet" configuration file keyword has the value ON.

Required Master Files table
This option allows editing the list of master files which should always be attached when the PLATO application is running. If one or of these master files is missing, users will be prevented from creating new files.

Special station list
This option allows editing the list of stations where special system functions can be performed by consultants and system software maintenance groups. The major feature allowed is the ability to inspect files when the password is not known.

Restrict system personnel access
This option allows changing the value of the "sysac" configuration file keyword while PLATO is running to temporarily allow system groups to sign on.

Edit/inspect system lesson access lists
This option jumps to lesson "s0calutil" to allow editing access lists to control user access to restricted system features.

Batch submission control

This option is used to control who may submit batch jobs and to edit a list of mainframes which can be used.

Continuous polling

This option allows changing the recall characteristics of the PLATO application which can result in more CPU time being available for batch jobs. This option has a HELP sequence which explains this control.

Network management

This option is the network system table editor. It is used to edit the list of systems and their characteristics.

Archive recycle period

This option is used to change the length of time an archive file may exist.

4.1

Preferred language table

This option allows setting the language some system displays such as the signon display and the Author Mode display will use. The only choices are English and French for these displays, but the "zlang" reserved word is available to user lessons to determine what language is desired.

Update level for new files

This option is an editor for the list of possible PLATO file types and their "update levels". New files which are created will have their update levels set to the value which appears in this table for the specific file type. When one of these values changes, you will be instructed to change the update level value during the upgrade installation. See the section on "File Conversions" in the PLATO Operations Guide for more information.

Multi-CONDENSOR submission

This option is used to determine which mainframes in a multi-mainframe configuration CONDENSORs will be submitted to when additional ones are required.

Change length of terminal location list

This option is used to lengthen the terminal location list which can be viewed in lesson "systemi" when new logical sites are added to the system.

Change length of EM allocation tables

This option can be used to lengthen the EM allocation tables when new sites are added to the system.

5 EM Management

This section describes the management of the EM which has been allocated to the PLATO application. This information concerns memory usage within the PLATO system itself, not how it affects the operating system.

5.1 The Lesson Buffer

Depending on the system configuration, a set amount of EM is reserved for overhead. This overhead represents the "binaries" for the various overlays as well as the many tables and buffers used by the application. Some of the overhead is of fixed size, but other parts of it will vary considerably in size, depending on the configuration of the system (i.e., number of sites, master files, etc).

What remains, after subtracting the overhead from the EM size, is the lesson buffer. The lesson buffer will contain all lessons, commons, storage, etc., which are in use. Once an entity is no longer in use, the "ECS manager" will periodically be activated and remove that entity from EM to make room for something else. If a user tries to use more EM when there is no more available, problems will arise. It is for this reason that EM must be allocated in an attempt to control the number of such problems, and to whom these problems will occur.

5.2 Logical Sites

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".

5.3 Allocateable EM

Not all of the lesson buffer may be allocated to the users. Some of it is reserved to contain student banks for each terminal signed in as well as for some of the system lessons. Most system lessons are in EM only when someone is using them (just as any normal lesson). However, a few of them (lessons "edit" and "sysopts", for example) always remain in EM regardless of whether or not someone is actually using them. Thus, you should not allocate EM represented by this lesson buffer overhead.

Furthermore, when users in other system lessons are not charged for the full length of the lesson, a certain amount of the buffer must be reserved for this uncharged EM.

The result of all this is when allocating EM, the entire EM buffer must not be allocated to the logical sites. A guideline which has worked well for most systems is to leave about 20% of the lesson buffer unallocated.

5.3.1 How Much to Allocate

There is no answer for how much EM can be safely allocated without causing problems for a fixed number of users. Here are two methods to give an estimate as to how much to allocate. Start the allocation with this amount and adjust to conform to actual need.

1. Use lesson "s0config" to determine the size of the lesson buffer for your configuration and do not allocate 20% of the length given.

2. In the "EM management statistics" option in lesson "system1", there is an option entitled "Lesson length/type distribution" which provides the following:

   a. the size of the lesson buffer
   b. the distribution of files by type throughout the lesson buffer
   c. the amount of lesson buffer overhead
   d. an estimate as to how much of the lesson buffer may be allocated

   In order for the estimate to be valid, run the option when the application is running at a peak load.

5.3.2 Adjusting the Allocations

Once EM has been allocated, there are several statistics which may be used to determine how well the allocation scheme is working, and adjustments should be made accordingly.

a. In lesson "stats", option b, you can see the EM free at 15 minute intervals during the day. If the EM free approaches zero, you are out of EM. This means the total amount of EM is about right; no more should be allocated.

b. In the "EM management statistics" section of lesson "system1", you can see the number of times users were totally unable to obtain requested EM. If this number is non-zero, there are times when the entire lesson buffer was full and someone tried to get additional EM. In such a case, the user would get a message such as "NO EM available". A few of these each day is no problem, but if the number gets too big, the entire system will be affected. In such a case, you MUST lower the EM allocated to one or more logical sites via lesson "allocate".

c. If you or other users receive messages like "You have exceeded your EM allocation", this means your site does not have enough EM allocated for all of its desired uses. You may allocate more to this site if other statistics have indicated that more EM may be available.

d. When a site is using all of its allocated EM and a student tries to get more, an author may be deleted to make room for the student, if this option is turned on in lesson "site". These deletions are recorded and may be seen in a PLATO Availability Report, or as part of the "opjobs" daily run. You may decrease the number of deletions by increasing the EM allocated to the site. However, if no more EM is available to allocate, you may have to accept these deletions.

5.3.3 Actions to Correct Memory Shortages

If statistics or messages indicate that there are problems, action should be taken, if possible.

a. Reduction of EM allocated to sites should only be necessary if there are several times when users are totally unable to obtain requested EM. In such a case, reduce the total amount of allocated EM.

b. Allocate more EM to the logical sites if there are problems and statistics indicate more EM is available.

c. If more than one non-system logical site exists, it may be possible to move EM from one site to another.

d. Change the system configuration file to reduce overhead.

e. Make sure that only essential runners are running during the peak hours of the day.

f. Use lesson "enforcer" to disable certain lessons during peak times of the day. Be warned, however, that the enforcer requires EM and may add to the shortage.

g. If management prefers to protect one group of users at the expense of another, set up a separate site for each group. Allocate more EM to the "protected" group than to the "unprotected" group.

h. Watch for users designing extremely large lessons and try to help them to make them smaller. Encourage authors working on lessons to condense only those blocks they are currently working on.

i. Obtaining more EM may be the only alternative.

5.3.4 Controlling EM Deletions

This section describes the algorithm used to back authors out of lessons when a student requests memory and there is not enough available in the student's logical site memory allocation. It also shows how to use the "edel1", "edel2", "edel3" and "sysdl" configuration file keywords to change the parameters of this algorithm.

```
    if author deletion not allowed for this logical site
      through option in lesson "site", then exit (failure).

    set BASE_AUTHOR_EM = 1500.
    set EM_NEEDED = REQUEST + 5000.
    set EM_OBTAINED = 0.

    loop
        if first pass, then set LIMIT = "edel1", else
            if second pass, then set LIMIT = "edel2", else
                if third pass, then set LIMIT = "edel3", else
                  exit (failure).

        loop
            get next author to check.
            set EM_IN_USE = length of (lesson + common + storage).
            if EM_IN_USE less than LIMIT, then reloop.
            set EM_AVAILABLE = EM_IN_USE - BASE_AUTHOR_EM.
            if protected system lesson, then
                if always protected, then reloop, else
                    if "sysdl" equals OFF, then reloop.
            if lesson also in use by students at site, then
              reloop.
            if lesson in use by more than 3 users, then reloop.
            back out author.
            issue NOS account file message (PD entry).
            set EM_OBTAINED = EM_OBTAINED + EM_AVAILABLE.
            if EM_OBTAINED greater than or equal to EM_NEEDED, then
              exit (successful).
        endloop

    endloop
```

5.4 EM Manager Parameters

This section describes the usage of the "emgr1", "emgr2", "emgr3" and "emgr4" configuration file entries and how changing these values will affect the PLATO system. Extreme care must be taken if you decide to change these parameters since drastic changes can severly affect the performance of the PLATO application and any other jobs running in the system.

The "EM Manager" is a program called periodically by the PLATO system to maintain free space in the lesson buffer. It does this in three phases:

1. by maintaining two different pointers to areas of

free space in the lesson buffer (Search Phase).

2. by deleting unused entries in the lesson buffer (Deletion Phase).

3. by relocating entries in the lesson buffer to open up larger areas of free space (Compaction Phase).

The amount of EM sought by the EM manager is set to "emgr4" unless a user request for EM has been refused because of inadequate memory. In this case, the amount sought is set to the greater of 100,000 words or one-eighth of the lesson buffer size. The number 100,000 is the maximum amount of memory which may be requested by a user.

Setting "emgr4" lower than recommended values will cause the EM Manager to spend most of its time in the Search Phase and to not advance to the other phases because it will be more likely to find a free space of the amount sought. This will make it use less CPU time, but unused lessons will not be deleted as often and compaction of the lesson buffer will be rare. This will cause users to get more errors due to EM shortages.

Setting "emgr4" higher than recommended values will cause the EM Manager to advance to the other phases more often and it will use more CPU time, reducing CPU time which could be given to users and to batch jobs. When the EM Manager is using too much CPU time, users in lessons with -backgnd- commands will freeze up because the EM Manager has priority over -backgnd- users, and condenses will take much longer because the PLATO executor is using most of the available CPU time.

5.4.1 Search Phase

If the number of lesson buffer entries is near the maximum, the EM Manager immediately advances to the Deletion Phase. This maximum is the value of the "lesns" configuration file keyword.

If there is more free EM available than that specified by the "emgr3" keyword, the EM Manager will immediately exit.

> Setting "emgr3" lower than recommended values will cause the EM Manager to search the lesson buffer less often and will lead to user errors due to memory shortages.
>
> Setting "emgr3" higher than recommended values will cause the EM Manager to search the lesson buffer more often, using up CPU time which could be given to PLATO users and batch jobs.

The Search Phase searches the lesson buffer for adequate free EM and sets two pointers called the "short request pointer" and the "long request pointer".

The short request pointer is used when filling requests for memory which are shorter than "emgr1". It points to the last free area of EM smaller than "emgr4" but greater than "emgrl" found during the lesson buffer search.

> Setting "emgrl" lower than recommended values will cause more user memory requests to be considered as long requests, which take more time to process and will cause the EM Manager to be called more often. This will also make the short request pointer less useful because the area of free EM it points to will be smaller and it is more likely that a user request will occupy the entire free area, leaving none for the next user short memory request. When this happens, the next user must do a full search of the lesson buffer for a free area, which is very expensive in CPU time.
>
> Setting "emgrl" higher than recommended values will cause the EM Manager to ignore areas of free memory which would otherwise qualify to be pointed to by the short request pointer. This will lead to many small areas of free EM appearing throughout the lesson buffer which cannot be used until they are collected by the Compaction Phase.

The long request pointer is used when filling requests for memory which are greater than "emgrl". It points to the first area of EM larger than the amount sought during the lesson buffer search. If no area of free EM is found which is larger than the amount sought, the EM Manager advances to the Deletion Phase.

5.4.2 Delete Phase

The Deletion Phase searches the lesson buffer for a predetermined amount of time, deleting unused entries. It begins the search at the point where it left off the last time is was called. If the EM Manager must exit before it has finished a complete search of the lesson buffer due to running out of time, it saves the point it reached and sets a flag so it will begin with the Compaction Phase when it is called again.

If the search reaches the end of the lesson buffer before the time limit has been reached, the EM Manager will reset the short request pointer and the long request pointer.

If an area of free EM larger than the amount sought could not be found and if there is at least 20,000 words of free EM overall which could be used if it were in one large area, the EM Manager will advance to the Compaction Phase. Otherwise, it exits.

5.4.3 Compaction Phase

If there less than twice the value of "emgr2" words of EM available throughout the lesson buffer, the EM Manager will return to the Deletion Phase to attempt to delete more lessons. This is done because the Compaction Phase is extremely expensive in CPU time and this amount is too small to be worth the effort it would take to compact it. The value of "emgr2" is also used to reject user requests for large amounts of EM when there is less than "emgr2" words free.

> Setting "emgr2" lower than recommended values will allow users making requests for large amounts of EM to use up EM which should be reserved for the shorter requests. Priority should normally be given to users requesting small amounts of memory.
>
> Setting "emgr2" higher than recommended values will cause user requests for large amounts of EM to be rejected more often, which leads to the EM Manager searching for larger amounts of free EM more often. Since it must advance to the higher Phases in order to obtain the larger amounts of free EM sought, it will consume more CPU time.

If there is enough memory available to make the Compaction Phase worth the CPU time it will take, the EM Manager searches the lesson buffer for entries which can be moved to open up larger free areas of EM. When it completes this search, it resets a flag so the EM Manager will return to the Search Phase when it is called again.

6 Disk System Management

6.1 Setting Up Your Disk System

Optimization of the PLATO disk system is done by proper use of the following.

1. Sharing use of mass storage devices between PLATO master files and NOS.

   Each copy of the PLATO disk driver (PMS) can process disk requests on several different mass storage devices. The processing of a disk request consists of two distinct steps, physically moving the read/write head on the device to the proper position and transferring the data from the disk to PLATOs extended memory. Each copy of PMS can direct the positioning of the read-write head on several different mass storage devices at the same time, but can transfer data from only one mass storage device at a time.

   The positioning process is much more time-consuming so it is best to avoid sharing the use of a mass storage device between NOS and PLATO files since this could result in conflicts between different PPUs wishing to position the device at the same time. PMS avoids deadlock conflicts by dedicating itself to positioning a device which is shared by the NOS system instead of attempting to position other devices at the same time it is positioning a shared device. This will slow down processing of all following disk requests.

   PMS dedicates itself to positioning a device when it has any of the following characteristics.

   equipment not of type DD, DG, DI, DJ, DK, DL, DM, DQ
   copy of system on device
   device shared between mainframes
   dayfile, errorlog, accountlog or mainlog on device
   files other than master files attached to jobs
   temporary, rollout or output files allowed on device
   multi-spindle device
   independent-shared device

2. Assigning Equipment Status Table (EST) entries for mass storage devices which contain PLATO master files.

   This factor is not important for systems using only one copy of the PLATO disk driver. When two copies are used, one PPU processes all requests for devices with even numbered EST entries and the other processes all requests for devices with odd numbered EST entries. When four copies are used, the distribution of PPUs to devices is:

PPU	Last Digit of EST ordinal
0	0,4

1	1,5
2	2,6
3	3,7

3. Creating PLATO master files with the MFCREAT utility.

Master files should be created individually so that the tracks allocated to the file will be contiguous. This will reduce the amount of work required in the disk system to find the proper track for a disk access. If more than one job is creating a master file at the same time, the tracks allocated to each file will be randomly intermixed. This also applies to copying master files from one disk or tape to another by using NOS utilities.

6.1.1 Setting Up Your Disk (continued)

4. Setting the "npms" PLATO configuration file entry.

This parameter determines the number of copies of the PLATO disk driver (PMS) to be loaded when the MASTOR job is initialized. This disk driver transfers data between the PLATO master files and PLATO's extended memory (EM) where it is accessed by users. The number of copies of PMS must be a power of 2 (i.e., 1, 2, or 4). Each copy of PMS used occupies a dedicated PPU. PMS uses a disk channel when processing a disk request and, on 700-series mainframes only, uses a distributive-data path (DDP) port access to ECS or a low-speed port access to ESM or a CM-data-transfer path, which is defined by the "ncmb" configuration file entry. PMS uses the EM port defined by the "D1" EQPDECK entry. This equipment is shared with other PPU programs, such as MRQ and MAS. On 800-series mainframes, PMS uses direct access to UEM to transfer data and does not need a CM-data-transfer path.

The number of copies of PMS to be used depends on the following factors.

1. Maximum number of users active on the system at one time.

2. Number of disk controllers and disk channels configured for the system.

3. Number of DDPs, ESM low-speed ports or CM-data-transfer paths configured for the system.

5. Setting the "ncmb" PLATO configuration file entry.

This configuration file entry determines the number of CM-data-transfer paths available for use by PMS when a DDP access to ECS or a low-speed port access to ESM is not available (either none are defined for the system, or all are reserved by other PPs). Users of 800-series machines must set this parameter to zero or omit it from the configuration file. For 700-series machines, this parameter should never be set higher than the "npas" configuration file entry since this would just waste CM and not improve performance. A general guideline to follow is that "ncmb" should be set to "npas" minus the number of DDP or low-speed ports available for use by PMS. In practice, if CM is short on a lightly-loaded system, this parameter may be set lower since the copies of PMS are very unlikely to all need a CM data transfer path at the same time.

6.2 User File Space Management

6.2.1 File Space Monitoring

File space should be monitored on a regular basis. The amount of space in use and that still available may be seen by looking at the options in lesson "operator". While looking at "operator", there are 3 things that should be looked for:

1. If another master file is necessary, because the current ones are almost used up.

2. If another master file slot needs to be added. For a smooth production environment, there must always be one slot for each required master file, plus one additional master file slot for special operations (e.g., loading a backup master file to get a backup copy of a lesson). To add another slot, simply change the value of the "ndsus" keyword in the configuration file.

3. If another disk unit is necessary. The initiative for obtaining an additional disk unit should be started well in advance of its actual need. When you start to use the last unit, it is time to think about getting the next unit.

6.2.2 Adding a Required Master File

The following steps must be followed whenever a new required master file is added.

a. Change EQPDECKS if a new disk is to be added.

b. Create the master file with MFCREAT.

c. Change procedure MFNX on the deadstart tape. Both the RESOURC and ATTACH commands may have to be changed.

d. You may need to increase the value of the "ndsus" keyword in your PLATO configuration file.

e. Make changes to dump and load procedures so that the new master file will be properly dumped. This includes changing MFDX and the master files to be dumped table through BACKMOD.

f. Build a new deadstart file with the modified copies of the EQPDECKs, MFNX, MFDX and your configuration file.

g. Reload the operating system on the new deadstart file.

h. Reload the PLATO application. The new master file should be attached by MFNX during initialization.

i. Change the required master files table through lesson "ipedit".

Refer to the PLATO Operations Guide for more information.

6.2.3 Changing a Required Master File

The following steps must be followed whenever an existing master file is changed, such as when changing the NOS name of a master file or changing the disk pack on which a master file is located. This procedure is also using during the initial installation of published courseware.

a. Change EQPDECKs if a new disk is to be added.

b. Make the required master file name change or move the master file to its new location.

c. Change procedure MFNX on the deadstart tape. Both the RESOURC and ATTACH commands may have to be changed.

d. You may need to increase the value of the "ndsus" keyword in your PLATO configuration file when doing the initial installation of courseware.

e. Make changes to dump and load procedures so that the new master file will be properly dumped. This includes changing MFDX and the master files to be dumped table through BACKMOD.

f. Build a new deadstart file with the modified copies of the EQPDECKs, MFNX, MFDX and your configuration file.

g. Reload the operating system on the new deadstart file.

h. Reload the PLATO application. The changed master file should be attached by MFNX during initialization.

i. Change the required master files table through lesson "ipedit".

Refer to the PLATO Operations Guide for more information.

6.3 Binary File Space Management

Lesson "binary" is a runner job which periodically goes through all of the binary master files and purges old binaries (thus making room for new ones). The criteria for deletion may be changed by systems personnel by executing the lesson itself. The controlling parameters are:

a. Minimum age for deletion: This is the minimum number of hours a binary must exist before it is normally a candidate for deletion. This is normally set to 24.

b. Disk space low threshold: Binary deletion will be initiated whenever the number of file parts remaining on the master file is less than this number. This is normally set to 500.

c. Disk space high threshold: Once binary deletion has begun on a master file, it will continue until either this number of file parts are free or until the entire master file has been scanned. This is normally set to 900.

d. Number files threshold: Binary deletion will be initiated whenever the number of files on a master file is greater than this number.

7 CPU Usage Management

7.1 Statistics Collection

CPU usage may be examined via the "CPU consumption" option in lesson "system1".

If you wish to collect statistics over a long period of time, you should do the following:

a. Create file "s0cpudata".

   NOTE: Since this file name is longer than 8 characters, use the "create a file" option under the "File Options" in lesson "operator" to create it.

b. Add "s0cpustat" as a runner.

These statistics may also be viewed via the option in "system1".

7.2 Adjusting "cpspd"

Application lessons should work the same on all systems, regardless of the machine speed. In order to do this, the PLATO application must give more CPU time to users on slow machines and less to users on fast machines. This is done by setting the "cpspd" configuration file keyword to the correct value for your machine.

Initially, "cpspd" should be set to the value given in the section on configuration file keyword definitions.

Lesson "s0cpspd" may then be executed to adjust the value of "cpspd" for the specific machine configuration in use.

8 Network Management

8.1 Physical Sites

Physical sites may be numbered from 0 to 62. The equipment connected to each site is determined by the configuration file.

8.1.1 Adding a New Physical Site

A new physical site should be added to your network as follows:

a. Change the configuration file network parameters to reflect the additional site(s).

b. If the site being added is a NAM site (keyword "nlsit"), make sure that the 2550 can drive as many ports as NAM sites defined. The number of ports available on the 2550 can be changed by doing a partial build of CCP, which runs in the 2550. A description of that partial build process can be found in the PLATO Operations Guide and the NOS Installation Handbook.

c. Bring up the application using the new configuration file.

d. Lengthen the terminal location list as follows:

   1. Execute lesson "ipedit".

   2. Choose option to "Change length of terminal location list".

   3. When asked the number of sites, enter the desired number and press NEXT.

   4. Edit file "sites" and set terminal locations.

e. Lengthen the ECS allotment tables as follows:

   1. Execute lesson "ipedit".

   2. Choose the option to "Change length of EM allotment tables".

   3. When asked the number of sites, enter the desired number and press NEXT.

   4. If desired, use the option in lesson "allocate" to assign the additional stations to any of the existing logical sites.

f. Change file "s(netwk" so that it has a minimum of 18 * "nsite" records and "nsite" names.

g. Lengthen "narfile" so that it contains at least "nsite" records.

h. Update the network database as follows:

   1. Execute lesson "pnet".

   2. Press LAB for more editing options.

   3. Choose the option to verify the database.

   4. Press NEXT to start the verification. The routine will update the database as required.

8.3 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. See the PLATO Operations Guide for more information about PNI.

Key echo response time on PNI can be improved in terms of average and variability by making the frequently called NAM overlays CM/EM resident through LIBDECK entries. The appropriate overlays can be identified in the output produced from the NAM STATS package.

9 System Network Management

Some PLATO systems are linked together with communications links which allow the use of features such as inter-system general notes, personal notes and file transfers. Even without a link, a system may have an "authors" database for another system. In either case, a network system table entry must be maintained for all systems that are intended to be in the network.

9.1 Adding a System

A system only needs to be added if the current system, and the new system, are both part of the linked system network, or if the "authors" database for a particular system is to be accessed.

Use the following procedure to add a new system to the network system table.

9.1.1 Check for enough room in table

First, you must check that your system is configured to allow adding another system to the network system table.

a. Make sure that there is room in the network system table for another entry by checking the current number of systems in the table against the value of the "netms" configuration file keyword. Increase the value of the "netms" configuration file keyword if the limit has been reached.

b. Use the following procedure to determine if you will need to lengthen common "link" of file "sysfile", which contains the network system table.

   1. Edit file "sysfile".

   2. Press "+" until you find the block named "link".

   3. Press the letter which appears next to block "link" to edit it.

   4. Near the top of the next display, you will see the current length of the common displayed.

   5. If this length is greater than (1 + 10 * the value of the "netms" configuration file entry), you will not need to lengthen this common.

c. If the current length of the common is too short, you should use the following procedure to lengthen it.

   1. At a convenient time, back out all users. This is necessary to prevent a user from writing a common into file "sysfile" while the file is reorganized.

2. Edit file "sysfile".

3. Press "+" until you find the block named "link".

4. Press the letter which appears next to block "link" to edit it.

5. Press SHIFT-LAB for "other options".

6. Choose the "change length of common" option.

7. Choose a new length which is the lowest multiple of 320 greater than (1 + 10 * the value of the "netms" configuration file entry).

9.1.2 Modify network configuration file

If the new system is to be linked to your system through the PLATO Inter-system Link, you now need to modify your NOS communication network. If the new system is not to be linked to your system, continue with modifying the PLATO network system table.

To establish a connection to another system, you need to define a path through NAM's network configuration file and RHP's logical identifier (LID) table. This procedure and other details about the installation and operation of these applications can be found in the following references:

NOS Version 2 Feature Notes	
NOS Version 2 Installation Handbook	(60459320)
NOS Version 2 Analysis Handbook	(60459300)
Network Definition Language Reference Manual	(60480000)

Follow these steps:

a. Update the LID configuration file. Refer to the the NOS Version 2 Analysis Handbook for examples. You will need to share this information with the administrators of the other system.

b. Update your NDL file. Here are some examples of NDL entries you may need to make:

* LINE definitions

```
line:   LINE,PORT=port,LTYPE=ltype,TIPTYPE=tiptype,
        PSN=psn,NSVC=svcrlc,DFL=dfl,FRAME=frame,
        RTIME=timer,RCOUNT=count,DCE=yn2.
device: TERMDEV,STIP=stiptyp,NCIR=numcir,NEN=encir,
        DT=devtyp.
```

* INCALL and OUTCALL statements for X.25

```
INCALL    ANAME=ptfs,FAM=famname,UNAME=usernam,
          SNODE=srcnode,PORT=portnum,DNODE=dstnode,
                 DBZ=dwnlsiz,UBZ=upbsize,DPLS=dpls.
        OUTCALL  NAME1=ptfs,PID=pidname,SNODE=srcnode,
                 DNODE=dstnode,PORT=portnum,DBZ=dwnlsiz,
                 UBZ=upbsize,DPLS=dpls,SHOST=srchost,
                 DHOST=dsthost,DTEA=dtea.

        * INCALL and OUTCALL statements for shared 2550

        INCALL   ANAME=ptfs,FAM=famname,UNAME=usernam,
                 DBL=dwnblim,ABL=abl.
        OUTCALL  NAME1=ptfs,PID=pidname,SNODE=srcnode,
                 DNODE=dstnode,DBL=dwnblim,ABL=abl.

        * INCALL and OUTCALL statements for direct line
        *    or TRUNK

        INCALL   ANAME=ptfs,FAM=famname,UNAME=usernam,
                 SNODE=srcnode,PORT=portnum,DNODE=dstnode,
                 DBZ=dwnlsiz,UBZ=upbsize,DPLS=dpls.
        OUTCALL  NAME1=ptfs,PID=pidname,SNODE=srcnode,
                 DNODE=dstnode,PORT=portnum,DBZ=dwnlsiz,
                 UBZ=upbsize,DPLS=dpls,SHOST=srchost,
                 DHOST=dsthost.
```

Refer to the NOS Version 2 Feature Notes for more examples.

c. Build your new network configuration file and corresponding local configuration file using the NDLP system command. Refer to the Network Definition Language Reference Manual for examples.

d. Bring down NAM, then reload NAM using the new configuration files and test your network.

9.1.3 Modify PLATO network system table

You now need to update the PLATO Network System Table, which includes descriptions of the links between your system and other systems, and the options available to each link. Follow these steps:

a. Sign on to PLATO with your "p" signon.

b. Execute lesson "ipedit".

c. Choose the "Network Management" option.

d. Choose the "System Table" option. This takes you to the "Network System Table Management" display.

e. Choose the "Add a new system to the table" option.

f. Enter the name of the new system. This should be the same as the name specified by the "sid" PLATO configuration file keyword on that system.

The next steps to be taken depend on if the system is not connected to your system through a link, the system is directly connected to your system through a link or the system is indirectly connected to your system through a link. A "directly connected" system is one which is linked to your system with no intermediate systems while an "indirectly connected" system is linked to your system through one or more intermediate systems.

9.1.3.1 Unconnected Systems

If the system you are adding is not connected to your system through a link, follow these steps. Otherwise, continue with the next section.

a. Choose the "Not linked to this system" option.

b. On the next display, enter the routing identifier for the new system. This must be the same as the "rid" PLATO configuration file keyword on the new system.

c. If there is to be an authors database file for the new system on your system, set the "authors database availability" option to on.

d. Continue with the "Final steps" section which follows.

9.1.3.2 Directly Connected Systems

If the new system is directly connected through a 2550 link, use the following procedure. Most of the information you need to enter in the table will have to be provided to you by the administrators of the remote system. If the new system is not directly connected by a 2550 link, continue with the next section.

a. Choose the "Directly connected by 2550 link" option.

b. On the next display, choose the "Routing Identifier" option. The RID you enter must be the same as the "rid" PLATO configuration file keyword on the new system.

c. Choose the "Link identifier:" option. Enter the logical identifier (LID) of the remote system (from your LID configuration file) and press NEXT.

d. Choose the "Family name:" option. Enter the NOS family name to be used on the remote system and press NEXT.

e. Choose the "User name password:" option. Enter the the user name password to be used on the remote system and press NEXT.

f. Choose the "Charge number:" option. If the remote site will not be using NOS charge numbers, or, if they plan to use the default charge number they specified for the PLARECV/PLASEND user numbers, press NEXT. If they want to account for each remote system's usage, enter the charge number that was agreed upon by you and the remote sites administrator here. Refer to the "Link Accounting" section of this document for details.

g. Choose the "Pack name:" option. Enter the pack name to be used on the remote system and press NEXT.

h. Choose the "Pack type:" option. If you specified a pack name in the previous option, specify the pack type and press NEXT. The default is "al".

i. Choose the "Encryption key:" option. You and the administator of the remote system should decide upon a alphanumeric key of up to 7 characters that is used as the seed for encrypting files sent between your systems. This key is used only between your system and this particular remote system. You should have a different key for each remote system in your table. When you have decided upon a key, enter it and press NEXT. You should remind the remote system administrator to do the same for your system in his network table.

j. Choose the "Encryption method:" option. Select "a. header only". (The option to encrypt the data being transfered is not available at this time.)

k. Turn on the appropriate data transfer options. For instance, if you want to be able to send and receive pnotes, turn both these options on. You must also set the "Status of link with this system" option to on to allow any data transfers to take place. You can inhibit all data transfers by setting this one option to off.

l. If there is to be an authors database file for the new system on your system, set the "authors database availability" option to on.

m. Continue with the "Final steps" section which follows.

9.1.3.3 Indirectly Connected Systems

If the new system is indirectly connected through a 2550 link, use the following procedure. Most of the information you need to enter in the table will have to be provided to you by the administrators of the remote system.

a. Choose the "Indirectly connected through a 2550 link" option.

b. On the next display, choose the "Routing Identifier" option. The RID you enter must be the same as the "rid" PLATO configuration file keyword on the new system.

c. Choose the "First intermediate system:" option. Enter the name of the system that is the first node between your system and the remote system, and press NEXT. The name you enter should be the same as that specified by the "sid" PLATO configuration file keyword on the intermediate system.

d. Choose the "Encryption key:" option. You and the administator of the remote system should decide upon a alphanumeric key of up to 7 characters that is used as the seed for encrypting files sent between your systems. This key is used only between your system and this particular remote system. You should have a different key for each remote system in your table. When you have decided upon a key, enter it and press NEXT. You should remind the remote system administrator to do the same for your system in his network table.

e. Choose the "Encryption method:" option. Select "a. header only". (The option to encrypt the data being transfered is not available at this time.)

f. Turn on the appropriate data transfer options. For instance, if you want to be able to send and receive pnotes, turn both these options on. You must also set the "Status of link with this system" option to on to allow any data transfers to take place. You can inhibit all data transfers by setting this one option to off.

g. If there is to be an authors database file for the new system on your system, set the "authors database availability" option to on.

h. Continue with the "Final steps" section which follows.

9.1.3.4 Final steps

Once you have completed entering information required for the type of system being added, follow these steps.

a. Press BACK to return to the "Network System Table Management" index.

b. Choose the "Update the EM copy of the system table" option. Press SHIFT-HELP when requested.

c. If the system is directly or indirectly connected via the PLATO inter-system link, run the "queue verification" system option in lesson "s0rhp". If errors occur, files "3netinq" and "3netoutq" may have to be lengthened.

d. If there is to be an authors database for the new system, use the system options of lesson "authors" to add the new database.

9.2 Deleting a System

A system may be removed from the network as follows:

a. Use the "network management" option in lesson "ipedit" to delete the system from the network table.

b. Use the option to "update the EM copy of the system table".

c. Run the "queue verification" system option in lesson "sорhp". When it pauses on an entry for a non-existent system, press SHIFT-HELP to delete it.

d. If the system is directly or indirectly linked by a 2550 link, you must also remove it from your LID configuration file and NDL file.

9.3 Renaming a System

Rename a system as follows:

a. Use the "network management" options in "ipedit" to inspect the entry for the system to be renamed. Write down the current parameter settings.

b. Delete the system from the table, but DO NOT run the verification options.

c. Add a system with the new name and set all parameters to those you wrote down.

d. Use the option to "update the EM copy of the system table".

9.4 Link Accounting

The PLATO Inter-System Link has been designed to accomodate the standard NOS accounting mechanisms via charge and project numbers. At this time, there is no accounting within PLATO, so charges cannot be billed to a specific user, group or account.

You have the option to use:

1) No charge and project number. In other words, no accounting at all.

2) Default charge and project numbers for user names PLASEND and PLARECV. This will give accounting for link traffic on your system as a whole. This

will not show you from which remote systems the traffic is coming.

3) A charge number specified by you, with a project number for each remote system. This will give accounting for link traffic based on the originating system.

Whatever option you choose, remote systems which connect to your system must have the corresponding information in their PLATO network system table. An error in charge information will inhibit link traffic until corrected.

9.4.1 No Charge/Project Numbers

If you do not specify a charge number in either the PLATO network system table or as defaults for PLASEND/PLARECV, there will be no accounting for link traffic. The "charge number" option for your system in the PLATO network system table should be set to the default value.

9.4.2 Default Charge/Project Numbers

When you specify default charge and project numbers for user names PLASEND and PLARECV, you will have a mechanism for accounting for link traffic on your system as a whole. This will show how much link traffic you have, but will not show from which remote system the traffic comes.

1) Set the default charge and project numbers for both PLASEND and PLARECV. This is done by using the MODVAL CN and PN parameters. We recommend that you use "PLATOLINK" as the default charge number for both user names. The default project numbers that you specify for the two user names may be different if you wish.

2) Set up the charge and project number you chose as defaults, using the NOS PROFILE command. Refer to the NOS Version 2 Administration Handbook for more information.

3) The "charge number" option for your system in the PLATO network system table should be set to the default value (*).

9.4.3 Charge Initiating System

When you specify a charge number in your PLATO network system table and have remote systems specify a charge number for your system in their PLATO network system table, you will be able to account for traffic based on the originating system.

1) Use PROFILE to create a charge number for link traffic. We recommend the name "PLATOLINK". Then define the following project numbers:

- one named "OVERHEAD" for link management jobs,
- one named "UNKNOWN" for unidentified connections,
- one for each remote system, using the system routing identifier (RID) as the name.

2) Use MODVAL to set the default charge and project numbers for user names PLASEND and PLARECV. Set the charge number to your specified charge number, and set the project number to "UNKNOWN". Connections using the default charge and project numbers will be accounted under this project number.

   The user numbers must also have the CNRD (charge not restricted to default) access word bit set.

3) Set the charge number for your system in the PLATO network system table.

It is possible for remote systems to use more than one charge number, but link jobs submitted by your system will use the charge number specified in your system's entry in the PLATO network system table.

Wherever possible, the RID of the system which initiated the traffic will be used as the project number for charges. The link management jobs which periodically check for incoming requests will be charged to the "OVERHEAD" project number. Connections which do not specify a charge number will be charged to the "UNKNOWN" project number.

APPENDIX A

EXTENDED SEMI-CONDUCTOR MEMORY MANAGEMENT

20.1 ESM Management

This section describes the management of Extended Semiconductor Memory. Documentation of memory management within the PLATO application is described in the section on EM Management.

The PLATO application, on 17x, or 170-700 series mainframes uses either ECS (Extended Core Storage) or ESM (Extended Semiconductor Memory) to store lesson material. The 800 series mainframes use UEM (Unified Extended Memory). This section is only applicable to systems which use ESM. The section on the "Low-speed port / DOP test program" is applicable to systems which use ESM or ECS.

ESM may either run in ECS mode or in ESM mode. In either mode the side-door port is available and the error log maintained by the hardware is available. In ESM mode the following additional features are available:

1. addressing up to 16 million words
2. accessing the 16,384 extended flag registers
3. use of the relocation memory to flaw banks

20.2 Configuration file keywords

The following PLATO configuration file keywords are used only on systems which use Extended Semiconductor Memory running in ESM mode.

EFRB: extended flag register base

> The value of "efrb" is the base or first ESM flag register to use. This allows several jobs (possibly even multiple systems) to share the extended flag registers.
>
> The PLATO application uses 64 flag registers, starting with the base register.
>
> On systems without ESM, "efrb" should be omitted. On systems with ESM, the usual value of "efrb" is 64.
>
> Default value: 64

20.3 Program ESM

This program is used to load ESM relocation memory, and to monitor and log errors. It is only used on systems which are using Extended Semi-conductor Memory. Many of the options of this program are used only if the memory is used in ESM mode.

The control statement format is:

```
    ESM(nk)
```

where:

nk = NK, if you do not want to use the K-display. The program will roll out and periodicially roll in to update the disk copy of the error log.

The K-display commands are listed below. All of the commands, except for "ERRORS", require that there be an ESM controller present, which is the case when running more than two million words of ESM.

Command — Description

CLEAR. — Clear the error logs. All single bit and double bit errors are removed from the ESM error log.

CONFIG. — Build a relocation memory that matches the ESM configuration. No programs should be using ESM when this option is used.

END. — Save the relocation memory and terminate job. The relocation memory is not loaded into ESM.

ERRORS. — Display the current copy of the ESM error logs.

FLAW,bsu,bank.
FLAW,bank. — Toggle the flaw status of a bank. If no BSU is specified, BSU 0 is assumed. The relocation memory is sorted into bank order, placing all flawed banks at the end.

GO. — Save and load the relocation memory. Terminate program.

HELP. — Display the list of commands on the right screen.

INIT. — Initialize the relocation memory to 16 million words. When the relocation memory is over 8 million words, both the left and right screens are used to display the relocation memory. No programs should be using ESM when this option is used and no shared packs should be loaded in a multi-mainframe configuration.

LOAD. — Load relocation memory from file ESMRM under user SYSTEMX.

MA,addr. — Set "addr" as the highest logical address in use. No programs will be able to access past "addr". This extra memory may be used by Customer Engineering or may contain reserve or or flawed banks.

PA,addr. Set "addr" as the highest physical entry in use in relocation memory. "addr" is the number of banks available - 1.

RELOC. Display relocation memory.

SAVE. Write relocation memory to file and save it for later loading. This file is called ESMRM and is defined under user SYSTEMX.

STOP. Terminate program.

SET,LA=addr,PA=addr.
Set relocation entry "LA" to "PA". This option should only be used when specific relocation memory layout is required.

ZERO. Write zeroes to all of ESM. This option can be useful in clearing errors. No programs should be accessing ESM when this option is used.

When the relocation memory is displayed, the following symbols denote special status of a bank:

* - not addressable
f - flawed

20.3.2 Automatic Loading and Monitoring

NOTE
THE FOLLOWING SHOULD ONLY BE DONE IF THE ESM
IS RUNNING IN ESM MODE.

Prior to NOS 2.1, ESM error monitoring was only performed on PLATO systems by program ESM. Now, error monitoring can be performed by program ESM by using the SP EST entry to identify the ESM maintenance channel or by the system by identifying the ESM maintenance channel by using the "MC" parameter on the ESM EST entry ("DE"). See the NOS V2 Analysis Handbook for more information.

The following is applicable only if ESM errorlog monitoring is done by program ESM.

You can automatically load the relocation memory and monitor the error log by entering "X.ESM(NK)" at the console. To insure that this is run at all times, this entry is normally added as a DSD entry in the system IPRDECK.

When ESM is first brought up with the NK option, it will load the relocation memory which has been previously saved and roll itself out. Every five minutes the program is rolled in again. It checks the error log for new entries and logs each message in the NOS error log and in dataset "s0esmerr" if it exists and is formatted correctly.

NOS error log messages issued by ESM have the following format:

```
ESM SNGL ERR,BSU1,BANK22,SCAN3,CS4,BIT55 CAB66,MODULE77-88,CHIPf99.
ESM DBLE ERR,BSU1,BANK22,SCAN3,CS4,xxxxx CAB66,MODULE77-88.
```

where:

1	= BSU number
22	= bank number
3	= scan number
4	= chip select number
55	= bit number (or MB if multiple bit error)
66	= cabinet name containing the board in error
77-88	= module name containing the board in error
f99	= chip location on board
xxxxx	= 13 bits of address excluding BSU, bank, scan, and chip select

Dataset "s0esmerr" also contains a log of the error messages. The dataset can be of any size. It is treated as a circular file. When an entry is written to the last slot in the file, the next entry is written to the first slot. The record size must be 320 words. The first record contains pointers to the next slot:

Word 1 - 'esmerrs' (single quotes indicate right-justified, zero filled)
Word 2 - record number to which next message is written. Initially, the record number should be 2.
Word 3 - word number within record. Initially, the word number should be 1.

Records 2 through the end of the file contain error log message prefaced by the date and time of entry. If a message does not fit in a record, the next record is used; messages do not span record boundaries.

20.3.3 Reloading the Memory

NOTE
THE FOLLOWING SHOULD ONLY BE DONE IF THE ESM IS RUNNING IN ESM MODE.

The relocation memory must be loaded before NOS or the PLATO application may access ESM. Once the relocation memory is loaded, it only needs to be reloaded after:

- power off
- banks need to be flawed or unflawed
- unusual hardware problem

Normally, program ESM will automatically load the relocation memory when it first comes up. However, under certain conditions when the relocation memory has been destroyed or because the part NOS uses has been reconfigured, it will be necessary to load the relocation memory in the following manner:

- Deadstart WITHOUT an ECS equipment (DE or DP) entry in the EQPDECK.

- Enter "X.ESM(NK)." at the console (unless this is automatically done via an IPRDECK entry).

- Re-deadstart in the normal manner.

20.3.4 Error Log Monitoring

ESM hardware uses SECDED to detect and correct for memory errors whenever a single, double, or multiple-bit error occurs, and an entry is made in an error log maintained by the hardware. Program ESM monitors this error log and displays NOS error log message whenever a new entry is made. You can display these errors as follows:

- Enter "X.ESM." at the console.

- Assign the K display.

- Enter "K.ERRORS." if the error display is not present.

- enter "K.CLEAR." to clear the error log, if desired.

- Enter "K.GO." when finished.

20.3.5 Flawing banks

NOTE
THE FOLLOWING SHOULD ONLY BE
DONE IF THE ESM IS RUNNING IN
ESM MODE.

ESM banks may be logically flawed as follows:

- Roll the ESM job in or initiate via "X.ESM." if it is not active.

- Assign the K display.

- Enter "K.RELOC." if the relocation display is not present.

- Use "K.FLAW,bsu,bank." to toggle the flaw status of the appropriate banks.

- Enter "K.GO." when all changes are made.

20.3.6 Changing ESM size

Whenever you change ESM size, alter the configuration table as follows:

- Redefine the ESM entry in the EQPDECK to use the new size.

- After deadstarting, enter "X.ESM." at the console. Do NOT load the PLATO application software at this time.

- Assign the K-display to the control point on request.

- Use the entry "K.PA,rm" to set the maximum physical ESM memory address to use.

- Use the entry "K.MA,rm" to set the maximum logical ESM memory address to use.

- Enter "K.GO." at the console.

- Change the "rax" and "fix" PLATO configuration file entries, if desired, and load the PLATO application.

20.4 Low-speed port / DDP test program

The program DDPT is an on-line diagnostic program used to test a DDP or low-speed port concurrently with normal PLATO operations. See the PLATO Operations Guide for documentation of this program.

APPENDIX B

COMPUTER INTERFACE UNIT NETWORK PARAMETERS

21.1 EQPDECK entries

The following describes EQPDECK entries needed to run the Computer Interface Unit Network.

Note: CYBER 170-800 series mainframes do not support the CIU network.

These parameters are used.

```
ord = One- to three-digit octal Equipment Status Table
      (EST) ordinal of equipment
st  = (status)  ON or OFF
eq  = (equipment)  Controller number, which may vary
      with each system.  The most commonly used number
      is shown.
un  = Unit number
ch  = Channel number to which the equipment is connected
      (ch0 or ch1 is used if two channel numbers are
      required)
ic  = Input channel
oc  = Output channel
```

1. CIU

```
EQord=CI,ST=st,EQ=0,UN=un,CH=ic/oc.
```

If the system has one CIU, set "un" to 0. If the system has two CIUs, set "un" to 0 for the first CIU and to 1 for the second CIU.

2. PIU low-speed port / DDP

If the system has one CIU, use the following entry.

```
EQord=D2,ST=st,EQ=5,UN=un,CH=ch.
```

If the system has two CIUs, you must also have two low-speed port or DDP channels and use the following entry.

```
EQord=D2,ST=st,EQ=5,UN=un,CH=ch0/ch1.
```

21.2 Configuration file keywords

21.2.1 Unique keywords

The following keywords are only used on systems which have a Computer Interface Unit (CIU).

C0SIT: first site on CIU 0

The value of "c0sit" is the number of the first site for which the CIU communication network is used.

The usual value of "c0sit" is 0.

Default value: 0

CISIT: first site on CIU 1

The value of "c1sit" is the number of the first site to be used by a second CIU.

The usual value of "c1sit" is 0 since most systems do not have a second CIU.

Default value: 0

ETED: enhanced terminal error detection

The value of "eted" is either ON or OFF depending on whether or not enhanced terminal error detection is enabled. When this is ON, a block checksum is sent as part of data sent to the terminal, and this checksum is used to detect errors and request re-transmission of bad data. When this parameter is OFF, only the single parity bit sent with each terminal word is used for error detection.

The usual value of "eted" is ON.

Default value: OFF

FDLAY: frame delay for error correction

For purposes of error correction, terminal output must be retained in EM for a long enough time to be sure that the output was received by the terminal without error. The parameter "fdlay" determines the minimum time that such output is retained.

At 1200 baud, the standard "fdlay" of 45 is equivalent to retaining the output for 45/57 (approximately 3/4) of a second. If a system experiences longer echo times, then a larger value for "fdlay" must be used.

The usual value for "fdlay" is 45.

Default value: 45

FRAM1: CIU 0 frame length

The value of "fram1" determines the size of the "frame" used to send output to CIU terminals on the first CIU. A frame contains one output word for each terminal for which output is pending.

The usual value of "fram1" is:
a. 50 for systems with less than 64 CIU users
b. 100 for system with 64 - 128 CIU users
c. 200 for systems with more than 128 CIU users

Default value: 0

FRAM2: CIU1 frame length

The value of "fram2" determines the size of the "frame" used to send output to CIU terminals on the second CIU.

The usual value of "fram2" is 0, since most systems have only one CIU. If two CIUs are used, use the same values as given for "fram1".

Default value: 0

IST3A: IST-III ASCII resident availability

The value of "ist3a" is either ON or OFF depending on whetnar a user of an IST3 terminal may download tne ASCII resident while on the CIU. Tne reason for having tnis parameter is to allow a user on the CIU to load tne ASCII resident and then use some other ASCII feature such as the Interactive Access Facility (IAF).

The usual value of "ist3a" is OFF.

Default value: OFF

IST31: IST-III resident 1 availability

The value of "ist31" is eitner ON or OFF depending on whether resident 1 for the IST-III is available. This resident is a development version. The normal resident used is resident 0.

The usual value of "ist31" is OFF.

Default value: OFF

NC0SI: number of sites on CIU0

Tne value of "nc0si" determines tne number of sites connected to tne first CIU.

Default value: 0

NC1SI: number of sites on CIU 1

Tne value of "nc1si" determines the number of sites connected to tne second CIU. Systems without a second

CIU should set "nclsi" to 0.

Default value: 0

21.2.2 Changes to other keywords

NAMPD: PLATO drop time

The value of "nampd" is the time in seconds between the time the last user of PLATO through the ASCII network signs out and the time PNI drops the PLATO application.

This feature should be used ONLY on PNI-only systems. On a system with both CIU and ASCII networks, if "nampd" is set to a non-zero number, users of PLATO through the CIU network could still be signed on when PLATO is dropped by PNI after the last ASCII network user signs out.

Default value: 0

NSITE: number of physical sites

On systems which use both the CIU and ASCII networks there are special considerations when setting the value of "nsite".

All terminals will belong to a set of sites which are serviced by either the CIU network driver, PIO, or by the ASCII network driver, PNI.

These sets of sites must include all sites (except possibly the runner site), and must not overlap. For example, the following is an acceptable configuration:

```
nsite=13.   thirteen physical sites on this system
c0sit=0.    CIU 0 network begins at site 0
nc0si=8.    eight sites on CIU 0
nc1si=0.    CIU 1 is not available
n1sit=8.    ASCII network starts at site 8
nn1si=4.    four sites on ASCII network

*  note that site 12 is undefined so it may only
*  be used for runner terminals.
```

21.3 Runner programs

These runner programs are used only on systems which have a CIU network and are optional.

netmon: CIU network monitor

```
cycle = 1
restart = 10
priority = 30
```

s0ciurus: collect CIU network performance statistics viewed through lesson "ciudiag"

```
        cycle = 20
        restart = 5
        priority = 30
```

APPENDIX C

MULTI-MAINFRAME CONFIGURATIONS

22.1 Multi-mainframe Considerations

(Multi-mainframe features are not available for CYBER 170-800 series machines.)

Several things should be kept in mind when configuring a multi-mainframe PLATO application.

a. Multiple executors and formatters should be spread out over as many mainframes as possible. Since these programs are Central Processing Unit (CPU) bound, much more will be lost than gained if two of them end up contending for the same processor. This is especially true of formatters; two heavily loaded formatters will simply take over a machine, and leave little or no CPU time for anything else.

b. CONDENSORs are disk bound. When the PLATO application is first loaded, and there is little for the executors and formatters to do, one machine can support up to three CONDENSORs reasonably well. Once the initial peak loads are past, extra CONDENSORs will be dropped and loaded automatically as needed.

c. Essential equipment (disk controllers, CIUs, network devices, Extended Semiconductor Memory (ESM) side door ports, etc.) should be switchable between the usual primary mainframe and one other, in order to provide a backup in case the usual primary mainframe fails, or must be taken out of service for some reason. If everything is connected identically (same channels, etc.), it should be possible to switch primary mainframes simply by deadstarting with a different Machine Identifier (MID) and switching non-shared devices over to the alternate mainframe.

   The "mford" configuration file entry would also have to change and file LCONFIG would have to be removed to identify the backup system as the primary mainframe.

22.2 The "config" File

Each mainframe in the configuration must have its own copy of the configuration file. It is likely there will be slight differences in the configuration file for each mainframe. In some cases, such differences are mandatory.

The basic configuration file can still be shared between all mainframes and the keywords which must be different can be placed in the file LCONFIG to override the matching ones in the basic configuration file. LCONFIG serves two purposes. It allows this replacing of configuration parameters and it identifies the mainframe as a secondary one by its presence. LCONFIG is placed in the the permanent file catalog of NOS user name PLATOMF (user index 377773b).

22.2.1 Changes for Mainframe 0

When adding a mainframe, the following parameters MUST be changed on the primary mainframe (mainframe 0).

```
    nmf     must be set to the new number of mainframes
```

Other parameters which are likely to change when adding additional executors, CONDENSORs, etc.:

```
    ncond
    nexec
    nfmtr
```

If you are expanding your system size (EM size, disk, network, etc.) at the same time, review all configuration file entries to see if they need to be changed.

22.2.2 Keywords required for all mainframes

The keywords in this section must be included in the configuration file for every mainframe in a multi-mainframe environment. They are the only parameters required for any mainframe which will not be running an executor.

The parameters below must be the same on all mainframes:

```
    fix
    njob
    rax
    subun
```

The parameters below are required on all mainframes, but each mainframe may use its own setting:

```
    famly
    mford
    mmf
    passw
    secur
    syot
```

22.2.3 Keywords required for executors

On any mainframe which is running an executor, all keywords in the configuration file must be the same with the exception of those listed in the previous section and those listed below:

```
    cpspd
```

22.3 Configuration file keywords

22.3.1 Unique keywords

The following configuration file keywords are used only on systems which are running the PLATO application in a multi-mainframe configuration.

MFORD: mainframe ordinal

The value of "mford" determines the mainframe ordinal.

The primary mainframe uses the value 0 and secondary mainframes use a value greater than 0 but less than "nmf".

Default value: 0

MMF: jobs on this mainframe

The value of "mmf" is either ON or OFF depending on whether the PLATO application may use this mainframe for application related jobs (executors, CONDENSORs, etc.). If it is set to OFF, the second mainframe is only available for batch jobs.

The usual value for "amf" is ON.

Default value: ON

NEXEC: number of executors

The value of "nexec" determines the number of PLATO executors to be run on all mainframes. The distribution of these executors over the different mainframes is up to the site.

If more than one executor is available, users will be automatically distributed to the different executors in a manner that attempts to even out the load.

The value of "nexec" must be less than or equal to 3.

Default value: 1

NFMTP: number of formatters

The value of "nfmtr" determines the number of formatters to be run on all mainframes. The distribution of the formatters is up to the site.

If the value of "nfmtr" is 2, a FORMAT will be run in addition to FRAMAT. The formatters always remain at control points. This option should not be needed except on heavily-loaded multi-mainframe systems.
systems.

The value of "nfmtr" must be either 1 or 2.

Default value: 1

NMF: number of mainframes

The value of "nmf" determines the number of mainframes available for the submission of batch jobs or for the PLATO application jobs.

Most systems have only a single mainframe, so "nmf" is usually set to 1. If "nmf" is greater than 1, the parameter "bpecs" may have to be increased because of the extra extended memory used by MASTORN.

Default value: 1

22.3.2 Changes to other keywords

The following configuration file keywords may have different meanings in a multi-mainframe configuration. See the main section on the "PLATO Configuration File" for more information on each of these keywords.

NAM: PLATO-NAM Interface (PNI) availability

The value of "nam" is set to the number of copies of NAM and PNI running on a system when the ASCII network is in use for PLATO.

On systems which do not use the ASCII network for PLATO, the value of "nam" must be set to C. On single-mainframe systems which use the ASCII network for PLATO, the value of "nam" must be 1. On multi-mainframe systems, the maximum value of "nam" must be less than or equal to 3.

Default value: 1

NN1SI: number of NAM (PNI) sites

The value of "nn1si" determines the number of PLATO sites (groups of 32 terminals) for use with the ASCII network. If running in a multi-mainframe configuration with more than one copy of NAM and PNI, this keyword determines the number of PLATO sites assigned to the PNI with the ordinal 1, while "nn2si" determines the number of PLATO sites assigned to the PNI with ordinal 2, etc.

Default value: 2

NSITE: number of physical sites

When running with more than one copy of NAM and PNI in a multi-mainframe configuration with sites assigned to each copy of PNI, the sets of sites must include all sites defined by the value of "nsite" with the exception of the highest site which is used by the runner programs and all sets of sites must be contiguous with no overlap.

The following is a valid configuration:

```
nsite=7.
n1sit=0.     PNI 1 sites begin at 0
nn1si=4.     for four sites
n2sit=4.     PNI 2 sites begin at 4
nn2si=2.     for two sites

*  note that site 6 is undefined so it may only
*  be used for runner terminals.
```

N1SIT: first NAM (PNI) site

The value of "nn1si" determines the number of PLATO sites allocated. Each site may be used by up to 32 terminals. If running in a multi-mainframe configuration with more than one copy of NAM and PNI, this keyword determines the number of PLATO sites assigned to the PNI with the ordinal 1, while "nn2si" determines the number of PLATO sites assigned to the PNI with ordinal 2, etc.

SECUR: application security level

The value of "secur" is either ON or OFF depending on the desired security level. It is mainframe dependent, so each mainframe in a multi-mainframe system may use a different value. When set to ON, system lessons are prevented from accessing information for a mainframe. For example, if the value of "secur" is ON, no console displays may be seen via lesson "console".

SYOT: system origin permission

The value of "syot" can be different on each mainframe.

The value of "syot" must be greater than 0 on any mainframe which is to run a PLATO executor so that utilities such as "ldr" will work correctly.

22.4 System lesson parameters

An option in lesson "ipedit" can be used to determine on which mainframes additional CONDENSORs will be submitted when they are required.

Another option in lesson "ipedit" is used to edit a list of mainframe to which jobs may be submitted.

See the section on "System Lesson Parameters" in this Handbook.

22.5 LIBDECK Changes

When additional PLATO executors are to be run, the following entry must be added to the LIBDECK:

```
*proc     exec
```

RELEASE CHANGES

70.1 Release Changes

The following changes have been made to the PLATO Configuration Handbook for the current release.

TOPIC	SECTION	CHANGE	DELETE	ADD
PLATO Inter-system Link	8.2		X	
	9			X
	30		X	
Add ISD II (836) disk support	6.1	X		
Rename "coufl" to "coltn"	3.1	X		
	3.2.2	X		
	80.2	X		
Rename "nterm" to "users"	3.1	X		
	3.2.5	X		
	3.2.5.3	X		
	3.2.10	X		
	80.6.1	X		
	80.9	X		
Rename "syslk" to "sysac"	3.1	X		
	3.2.9	X		
	4	X		
	80.9	X		
Change "cospd" estimates	3.2.2	X		
Change "cpspd" default	3.2.2	X		
Change "coltn" default	3.2.2	X		
Change "emgr3" default	3.2.3	X		
Change "emgr4" default	3.2.3	X		
Change "fastl" default	3.2.4	X		
Change "forml" default	3.2.4	X		
Change "lesns" default	3.2.5	X		
Change "nlob" default	3.2.6.1	X		
Change "nparc" default	3.2.6.2	X		
Change "quesz" default	3.2.7	X		
Rewrite procedure for adding to network system table	8.2.1	X		

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.

80.1 Index: A - B

80.2 Index: C - D

80.4   Index: F - L

80.7 Index: O - P

80.9 Index: S - Z