Publication Number 29-379R01

OS/32-ST
PROGRAM CONFIGURATION MANUAL
(CUP/ST)

INTERDATA®
2 Crescent Place, Oceanport, New Jersey 07757

© INTERDATA INC., 1974
All Rights Reserved
Printed In U.S.A.
November 1974

OS/32-ST PROGRAM CONFIGURATION MANUAL

TABLE OF CONTENTS

APPENDICES

29-379R01
November 1974

OS/32-ST PROGRAM CONFIGURATION MANUAL

1. INTRODUCTION

The OS/32-ST Configuration Utility Program (CUP/ST; Program Number 03-076) provides to the user the ability to configure an OS/32-ST operating system tailored to meet specific requirements. CUP/ST runs as a utility task within an OS/32-ST operating system environment. Various software configurations of OS/32-ST can be generated from a library of system object modules by modifying the input to CUP/ST; this allows the user a great degree of flexibility in creating OS/32-ST operating systems.

A pre-SYSGENed OS/32-ST STARTER program (Program Number 03-075) is provided to run CUP/ST. The OS/32-ST STARTER program is provided primarily as an environment in which to run CUP/ST. However, it may have other applications, depending upon user requirements. The OS/32-ST STARTER program is described in more detail in Appendices 1 and 2.

The input to CUP/ST consists of a series of Configuration Statements and a DCB/Driver Library. The input medium for the Configuration Statements must support ASCII/Read; the input medium for the DCB/Driver library, Binary/Read; the CUP output medium must support BINARY/Write. The DCB/Driver library can be one of the following: the INTERDATA supplied OS/32 General Purpose Driver Library (03-073); a user-written DCB/Driver library as specified in the OS/32 Series Driver Manual (Publication Number 29-384); or a library containing both INTERDATA supplied DCB's and Drivers and user written DCB's and Drivers.

The purpose of this manual is to give the user a thorough understanding of the requirements and procedures involved in configuring an OS/32-ST operating system. It is in addition a reference manual and operator's guide for the use of the OS/32-ST Configuration Utility Program.

2. CONFIGURATION OF OS/32-ST

OS/32-ST is constructed in a highly modular form. Within each module, numerous System Generation (SYSGEN) parameters have been incorporated into the source code, providing a great degree of flexibility and fine-tuning at the source level. The size of OS/32-ST precludes the source-level SYSGEN as the everyday means of getting a system up and running. Furthermore, the majority of system generations involve the inclusion or deletion of one or more drivers, and perhaps of certain major modules, to meet new or revised system requirements. Here again the source-level SYSGEN is out of place. For this reason, the OS/32-ST configuration procedure is composed of two parts: the source-level SYSGEN, which may be used either for fine-tuning or for system modification, and the object-level configuration procedure, which provides a rapid method of putting together a system out of major packages.

2.1 Source SYSGEN

All OS/32-ST modules contain source SYSGEN parameters which give the user the facility to modify code generated at assembly time. In this way, the user is provided with a module general enough to meet various needs, which can then be tailored by user installations to meet specific requirements.

The source SYSGEN process is performed as follows: First the user assembles the OS/32-ST source library with the appropriate SYSGEN parameters set. The assembly produces object modules which must be linked by the OS/32 Library Loader (03-065) with the output produced by CUP/ST in the object SYSGEN process (see Section 2.2).

Source packages of the OS/32-ST are:

EXECUTIVE (07-062)
COMMAND PROCESSOR (07-066)
FILE MANAGER (07-064)
FLOATING POINT TRAPS (07-065)
CONTROL BLOCKS AND PARAMETERS (07-063)
UBOT (07-067)

Section 5 contains a detailed description of the Source SYSGEN options and describes how to perform an OS/32-ST source system generation.

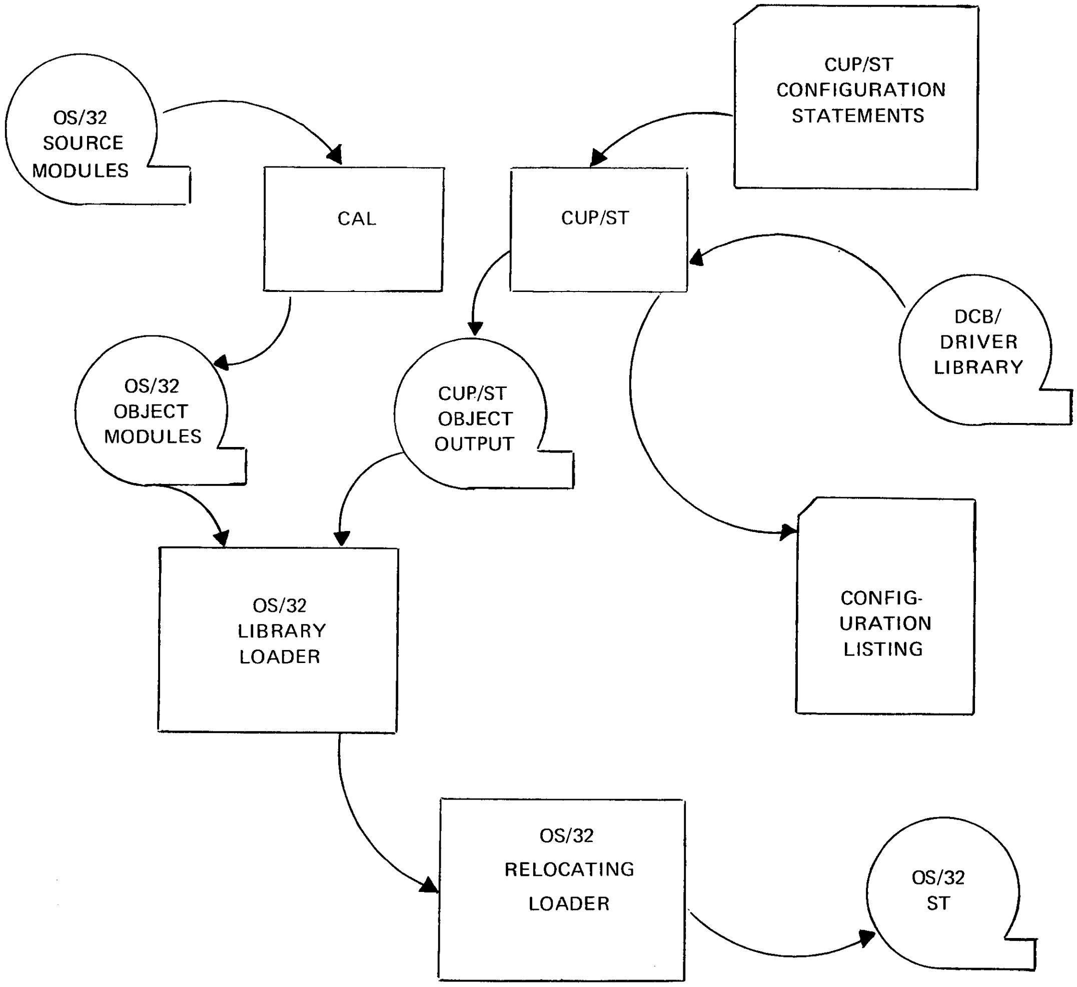


Figure 1. Typical CUP/ST Configuration

2.2 Object SYSGEN

The OS/32-ST Object SYSGEN procedure provides the user with the facility to modify the OS/32-ST without requiring a new assembly. The object-level SYSGEN process is accomplished by running the OS/32-ST Configuration Utility Program (03-076) to produce a file of object-format modules. This file must be linked by the OS/32 Library Loader (03-065) with either the pre-assembled OS/32-ST System Module Library (07-061) or the user's assembled version as described in Paragraph 5. The result of this procedure is a single absolute object-format load module which can be loaded using the 32-bit Relocating Loader (03-067).

The Object System Module library (07-061) consists of the relocatable object load modules:

EXEC.S01	- Executive with no integrity checks, with JOURNAL calls
CMDP.S01	- Command Processor with CSS, without direct access support
CMDP.S02	- Command Processor without CSS, without direct access support
CMDP.S03	- Command Processor with CSS, with direct access support
CMDP.S04	- Command Processor without CSS, with direct access support
FMGR.S01	- File Manager with Contiguous and Chain file support
FMGR.S02	- File Manager with no direct-access support
FMGR.S03	- File Manager with Contiguous file support only
FLTP.S00	- Floating Point Traps
UBOT	- Provide entry for top of Operating System

Paragraph 3 contains a detailed description of the object SYSGEN options and describes how to run CUP/ST.

3. THE OS/32-ST CONFIGURATION UTILITY PROGRAM (CUP/ST)

This section describes the method by which a user may perform an object-level system generation using the OS/32-ST Configuration Utility Program. CUP/ST must be run on an INTERDATA 32-Bit Processor under control of one of the OS/32 operating system family. The input to CUP/ST is a series of command statements, describing the configuration of the target OS/32-ST and a DCB/Driver library containing all the INTERDATA supplied Drivers with their associated Device Control Blocks (DCBs) and any user-written Drivers and DCBs.

Upon completion, CUP/ST produces an object library, to be processed with the OS/32-ST System Module Library by the OS/32 Library Loader (03-065), and a printout describing the configured system.

An example of the Configuration Statement input necessary to generate the OS/32-ST STARTER program is given in Appendix 1, a sample printout describing it is contained in Appendix 7.

3.1 General Principles of Operation

CUP/ST is a non-interactive program. A sequence of configuration command statements is input. Each statement is printed as it is read, in order to produce a hard copy of the configuration statement sequence. From time to time, as certain portions of the operating system (e.g., tables) are built, they are output in object format. When all configuration command statements have been read, CUP/ST then reads the DCB/Driver library and copies all required drivers and DCBs to the object output medium.

This DCB/DRIVER library can be one of the following: the INTERDATA supplied OS/32 General Purpose Library (03-073); a user-written DCB/DRIVER library built by the user as specified in the OS/32 Series General Purpose Driver Manual; or a library containing both INTERDATA supplied DCBs and DRIVERs and user written DCBs and DRIVERs.

When CUP/ST has terminated properly, the user has a file of object-format modules output by CUP/ST. This file must then be processed using the OS/32 Library Loader (03-065), together with the OS/32-ST System Module Library (07-061).

After all processing is complete, CUP/ST outputs a description of the OS/32-ST as configured. All defaults assumed by CUP/ST are flagged with an asterisk (*) in the left hand column of the printout.

If an I/O error occurs while processing the Configuration command statements or the DCB/Driver library, an informative message is logged and CUP/ST PAUSEs. At this time the I/O error can be corrected and CUP/ST CONTINUEd. All non I/O errors which occur during the execution of CUP/ST cause an error message to be logged and CUP/ST terminated. CUP/ST must be re-STARTed after the error is corrected.

The DCB/Driver library consists of a series of DCB/CCB and Driver programs. These programs may all be on the same physical medium as in the INTERDATA supplied DCB/Driver library or may be on separate physical mediums; for instance, one paper tape for each program. If an I/O error occurs while processing the DCB/Driver library between programs (as would always occur if each program was on a separate paper tape), the I/O should be continued as explained above. However, if an I/O error occurs while processing within a DCB/CCB or Driver program, the I/O error must be corrected and CUP/ST reSTARTed.

3.2 Environment

3.2.1 Memory. CUP/ST resides in 10 KB of memory, over and above that required by the operating system. In addition, a variable amount of memory is required for working storage. This amount of memory is dependent on the complexity of the I/O configuration being built. CUP/ST does not execute EXPAND calls; therefore, the user should give the program sufficient memory prior to starting it. Typical required workspace might be approximately 1200 bytes; a workspace of 4K bytes should be adequate for most large systems.

3.2.2 Devices. CUP/ST requires an input device capable of supporting read ASCII, an input device supporting read binary and an output device supporting write binary. If a listing is desired, an output device capable of supporting ASCII is necessary.

3.2.3 Logical Units. CUP/ST uses 4 logical units. These are:

LU 1	Configuration statement input
LU 2	Object format output
LU 3	Configuration statement listing and final configuration description
LU 4	DCB/Driver library input

Listing may be suppressed by assigning LU 3 to the NULL device.

Since LU 1 and LU 4 are not used simultaneously, it is possible to assign both to the same device, and to use that device to read first the configuration statements and then the DCB/Driver library.

Upon the completion of the Configuration statement processing phase of CUP/ST, the following message is logged:

```
CONFIGURATION STATEMENT PROCESSING COMPLETE
```

At this time, CUP/ST will issue a read to LU 4 to begin processing the DCB/Driver Library.

3.3 Configuration Statements

Configuration statements are read from Logical Unit 1. A hard copy of these statements is produced on Logical Unit 3.

With the exception that the Assignment statements must come last, no fixed order is placed on the input of statements. Most statements are optional and defaults exist for these. Where there is no default, or where a statement is mandatory, it is noted in the text. Appendix 5 contains a list of all statements and the defaults associated with each.

Each statement may occur only once. Only one statement is accepted per line.

The general format of a configuration statement is:

VERB <operand or operands>

The verb must appear at the beginning (first character position) of a statement line. This verb takes the form of a Mnemonic, as in the OS/32-ST command processor. A verb may be abbreviated by an unambiguous substring of its initial characters.

For example, the verb UNITS may be abbreviated as follows:

U
UN
UNI
UNIT

The minimum required characters for each verb are underlined in the descriptions which follow.

For clarity of notation, operands are shown in this manual enclosed in angle-brackets, e.g.:

<fd>
<address>

These angle-brackets are not a part of the operands, but are a notational convenience. The angle-brackets should not be entered in the actual text.

The notation [ ] is used to indicate an optional operand.

The blank character is shown in the text thus:

△

In general, wherever one blank is permitted, multiple blanks are permitted. In addition, all numeric operands may be optionally preceded by blanks. Leading zeros may be omitted in all numeric operands, whether decimal or hexadecimal.

Characters present on a line following the final operand are ignored. This permits comments to be placed on any configuration statement line.

The DEVICES, MODULE, and ASSIGN statements are followed by one or more statements which do not have the above syntax. The syntax of these sub-statements is given in the text. In every case, however, the statement must begin in the first character position of a line.

The following error messages may appear after any statement is read:

CUP: MNEM-ERR	Statement verb not recognized.
CUP: SEQ-ERR	Statement has already been accepted once; a statement may not be entered twice.

If an I/O error is detected on any read or write operation, the following error message is output:

```
CUP:    I/O-ERR xx fd
```

where xx is the returned SVC 1 error status in hexadecimal, and fd is the device Mnemonic or File Descriptor of the offending device or file. CUP/ST then PAUSEs. If the operator chooses to resume the program by entering the command CONTINUE, the erroneous I/O operation is retried.

3.3.1 Logical Units. The UNITS statement specifies how many Logical Units the user task is able to use. This may be a number anywhere between 1 and 255, inclusive. Note that the maximum LU number is one less than the number of logical units, since LU 0 is a valid Logical Unit. It is not recommended to use values smaller than 8 for UNITS, as certain INTERDATA-supplied programs will not run with fewer than eight Logical Units. The format of this statement is as follows:

```
UNITS△<n>
```

where <n> may be any number from 1 to 255. Errors are:

```
CUP:LU-ERR   <n> is less than one or greater than 255.
```

If no UNITS statement is found, the default is 16.

3.3.2 Memory. The MEMORY statement indicates the number of KB (1 KB = 1024 bytes) of memory on the system. The format of this statement is:

```
MEMORY△<n>
```

where <n> is a number between 32 and 1024 KB inclusive, evenly divisible by 16. Errors are:

```
CUP:MEM-ERR        <n> is greater than 1024, less than 32, or
                     not evenly divisible by 16.
```

Note that if the size of the OS/32-ST system as configured would exceed this memory size, the error is not detected by CUP-ST. Such an error is discovered when the final load module is created.

If this statement is omitted, the default OS/32-ST system size is 48K.

3.3.3 Device Addresses. Although the standard number of device addresses on the 32-bit processors is 255, it is possible to optionally increase this number up to 1023. The DEVADS statement provides the user with the capability of supporting this increased device addressing range. The format of the statement is:

```
DEVADS △ <n>
```

where <n> may be the digit 0, 1, or 3, as follows:

<n>	No. of Dev.	Maximum Address	Space Used for ISP Table
0	255	X'0FF'	512 bytes
1	511	X'1FF'	1024 bytes
3	1023	X'3FF'	2048 bytes

This number should be set identically to that provided in the system's hardware configuration. If the number is greater than that which the system actually supports, substantial space may be wasted in the Interrupt Service Pointer (ISP) table; if the number is less, interrupts from higher-numbered devices cause system failure with possible destruction of data. Errors are:

```
CUP:DN-ERR          <n> not equal to 0, 1 or 3
```

If no DEVADS statement is found, the default is <n> = 3.

3.3.4 Default Volume. The VOLUME statement is used to specify the initial default volume. Its format is as follows:

VOLUME Δ<voln>

where <voln> is any volume identifier of four or fewer characters. Specifying more than four characters causes the string to be truncated after the fourth character. If no VOLUME statement is entered, the default consists of four blanks. (This is not a legitimate volume identifier and must be corrected by the console operator after system initialization.) Note that any ASSIGNs which specify a default volume result in ASSIGN errors at the time the ASSIGNment is attempted.

The following error is generated if the volume name is syntactically incorrect:

CUP:VOL-ERR

3.3.5 Journal. The JOURNAL statement indicates the number of journal entries within the system. The System Journal is a list of historical data, maintained by the operating system, each entry consisting of five fullwords of information. The format of this statement is:

JOURNAL Δ<n>

where <n> is the number of entries required, a positive number ≤ 13,000.

CUP/ST will reserve 20*n+8 bytes of storage for the JOURNAL.

If this statement is omitted, the default Journal size is 6 entries.

Refer to the OS/32-ST Program Logic Manual (29-381) for an explanation of the System Journal.

3.3.6 Maximum Disc Buffer Size. This parameter specifies the maximum physical block size allocatable for any Chained disc file on the target system. Its format is:

DISCBLOCK Δ <n>

where <n> is any non-zero decimal number less than 256. This number represents the size, in 256-byte sectors, of the maximum physical block size that can be specified for any Chained file at Allocate time. For example:

DISCBLOCK 4

specifies that a Chained file may be allocated with a physical block size of 1, 2, 3, or 4 sectors on the target system. This storage is not required until such time as the file is actually assigned.

The following error may be detected:

CUP:BLK-ERR <n> is zero or greater than 249.

This statement may be omitted if there are no direct-access devices in the system, or if the user has removed chain-file support. If this statement is omitted, and Chained file support is configured, the default DISCBLOCK maximum size is 4.

This statement is ignored if there is no chain file support in the system.

3.3.7 Command Substitution System (CSS). The CSS statement provides the user with the capability to set the maximum nesting depth for Command Substitution System (CSS) calls. Its format is:

CSS Δ<n>

where <n> is any number from 1 to 249. The following error may be detected:

CSS-ERR

If <n> is omitted, the default nesting depth is set to 3. If n=0 or this statement is omitted, CSS is excluded from the system.

"Maximum nesting depth" in this context refers to the number of CSS files that can be active at any one time. If the statement

```
CSS 1
```

is entered, then the target system gives the user the capability of calling CSS files from the console, but these files may not call other CSS files; if the statement

```
CSS 2
```

had been entered, then files called from the console could in turn call other files, but these in their turn could not make further calls, and so forth.

3.3.8 Command Buffer Length. The CMDLEN statement sets the length of the system's command buffer (or command buffers, if CSS is present). Its format is:

```
CMDLEN △<n>
```

where <n> is a decimal number from 32 to 1024 inclusive. The following error may occur:

```
LGTH-ERR          <n> less than 32 or greater than 1024
```

If this statement is omitted, the default value is 80.

Note that although the maximum line length of the system console device may be 72 or 80 bytes, a greater buffer length may be desirable if CSS is present. This is for two reasons:

Commands may be read from devices or files with a greater record length;

Argument replacement may cause lines, once read in, to be expanded, as in the following example:

CSS call:

```
CREATE PROGRAM1.SRC
```

In CSS file CREATE.CSS:

```
IFNX @1; ALLO @1; ENDC
```

which is expanded to:

```
IFNX PROGRAM1.SRC; ALLO PROGRAM1.SRC; ENDC
```

3.3.9 Log Message Buffer Length. The LOGLEN statement sets the maximum size of the Log Message buffer. Any user task running under the system being configured that attempts to log a message of length greater than this loses trailing bytes of data. The statement format is:

```
LOGLEN △<n>
```

where <n> is a decimal number between 32 and 128 inclusive. The following error message may occur:

```
LGTH-ERR          <n> less than 32 or greater than 128
```

If this statement is omitted, the default value is 72.

3.3.10 Device Description. These statements specify not only the individual devices within the system to be configured but also define to CUP/ST the exact physical structure of the I/O system. The DEVICES group statements may not be omitted. Its format is as follows:

```
DEVICES                                         Start of device statements.
<level>  [<dm> : <dn> , <dcod> [,<flags>]]      As many as required.
ENDD                                            End of device statements.
```

The individual fields within the device statements have the following formats and meanings:

<level> must contain either an asterisk (*) or a positive non-zero number. The asterisk is used only for resolution of shared-busy conflicts, described below. The number is used in most cases, and relates to the number of devices, controllers, or channels through which requests from the Processor to this device must pass. For an independent device (one interfacing directly to the Processor) the number in this field should be 1. Appropriate level numbers are given for each individual device or controller in Paragraph 4 where device-dependent configuration requirements are specified.

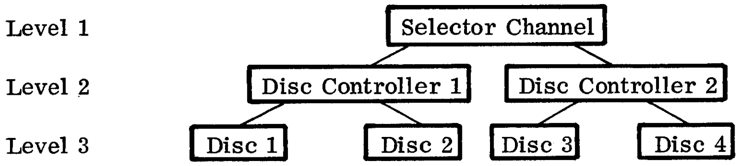
<dm> is the device Mnemonic, the name by which the device is to be known to the system. It must be composed of from one to four characters. The first character must be alphabetic; the remaining characters, if any, must be alphanumeric. Each device Mnemonic must be unique. The <dm> field may be omitted for all channels and controllers; it is only required for actual devices.

<dn> is the device number (physical address), in hexadecimal, of the device, controller or selector channel, which may contain leading blanks. It may not be greater than the maximum device number reflected by the DEVADS statement. For devices or controllers which have no actual physical device number, this field should contain zero.

<dcod> is the device code, in decimal, of the device, optionally containing leading blanks. A device code must be a number greater than 15 and less than 255. Device codes are given in Appendix 3; also see the OS/32 Series General Purpose Driver Manual.

<flags> is a field that should contain the letter C, for the Console device, or the letter D, for a Directory device (e.g., moving-head disc). The presence of a Directory flag indicates to CUP/ST that direct access support is to be provided. If the device in question is neither of these, the flags field must be omitted. The <flags> field must contain only a 'C' or 'D' with no leading blanks.

Device statements must be placed in an order that, for any device at level n, the previous device at level n-1 is its controller. Thus, given this structure:



The following is a valid order for the device statements:

1	(Selector Channel)
2	(Disc Controller 1)
3	(Disc 1)
3	(Disc 2)
2	(Disc Controller 2)
3	(Disc 3)
3	(Disc 4)

The following is an invalid order:

```
1      (Selector Channel)
2      (Disc Controller 1)
2      (Disc Controller 2)
3      (Disc 1)
3      (Disc 2)
3      (Disc 3)
3      (Disc 4)
```

because this order implies that Disc 1-4 are all controlled by Disc Controller 2.

Certain devices, because of hardware considerations, must share devicc-busy logic with certain other devices at their own level. These cases are pointed out in Paragraph 4; an example is the keyboard/printer and reader/punch of an ASR Teletype. In such a case, the device specification statements for all devices in the interrelated group must be consecutive; the first device specification statement in the group contains the actual level in the level field, but the others contain an asterisk (*) in this field.

Examples of Device Configurations:

1. The user's hardware environment consists of a card reader (400 CPM), a line printer (200 LPM), a Model 33 KSR Teletype (which will be used as the Console device), and an INTERTAPE cassette transport, containing two cassette drives.

There are no Selector Channels or controllers; all levels are level 1. There is a shared-busy conflict between the cassettes. The configuration statements are:

```
DEVICES

1 TTY:2,016,C        Console TTY, dn = X'002'
1 CARD:4,096         Card reader, dn = X'004'
1 PRIN:62,112        200 LPM printer, dn = X'062'
1 CAS1:45,066        Cassette #1, dn = X'045'
* CAS2:55,066        Cassette #2, dn = X'055'
ENDD
```

2. The user has a hardware environment consisting of the following:

One nonediting CRT, device number 010, to be used as console.
Two M33 ASR Teletypes, device numbers 002 and 012.
One 400 CPM card reader, device number 004.
One 200 LPM printer, device number 062.
One selector channel, device number 0F0, on which are:

One magnetic tape controller, controlling
Two magnetic tapes (800 BPI), device numbers 085 and 095.

One disc controller, device number 0B6, controlling
Two 2.5 Mbyte removable discs, 0C6 and 0D6.

All devices not on the Selector Channel are at level 1. Since the Selector Channel is at level 1, the tape and disc controllers are at level 2; this forces the tapes and discs to level 3. No shared-busy conflicts exist other than between the ASR keyboard/printer and reader/punch units. The configuration statements are:

```
DEVICES

1 CRT:010,018,C        Console CRT
1 TTY1:002,016         M33 ASR TTY
* TRP1:002,081           Reader/Punch
1 TTY2:012,016         M33 ASR TTY #2
* TRP2:012,081           Reader/Punch
```

```
1 CARD:004,096          400 CPM Card Reader
1 PRIN:062,112          200 LPM printer
1:0F0,0                 Selector Channel
2:0,0                     Magtape controller
3 MAG1:085,064              Magtape #1
3 MAG2:095,064              Magtape #2
2:0B6,0                   Disc Controller
3 DSC1:0C6,049,D            2.5 Mbyte disc #1
3 DSC2:0D6,049,D            2.5 Mbyte disc #2
```

Errors detected by CUP/ST while reading the device specification statements are as follows:

CUP: NODV-ERR — The ENDD statement directly follows the DEVICES statement or no device statements were found at all.

CUP: NFUL-ERR — CUP/ST ran out of memory while processing device statements.

CUP: LEVL-ERR — <level> is zero, or greater than the previous level plus one (this may indicate shuffled statements).

CUP: SHAR-ERR — <level> is contains an asterisk, but either there is no previous statement, or the previous statement refers to a controller or channel rather than a device. (This error may have been caused by the previous statement rather than by the one currently being processed).

CUP: DM-ERR — The previous statement did not refer to a controller or channel, yet it had no <dm> field.

CUP: DMSN-ERR — The <dm> field is syntactically incorrect.

CUP: SEPR-ERR — Missing colon (:) before <pa> , or missing comma (,) after <pa> .

CUP: DN-ERR — <dn> contains a number greater than the maximum physical device address for the system.

CUP: DCOD-ERR — <dcod> contains other than a decimal number between 16 and 255.

CUP: FLAG-ERR — <flags> contains something other than a C or a D.

CUP: CONS-ERR — More than one device was flagged as the Console device.

3.3.11 Initial Options. The OPTIONS statement sets up the default options for tasks running under the system being configured. Its format is:

OPTIONS Δ <option> [,<option> [,<option>]]

where <option> may be any one of the following:

FULL
HALF
UT
ET
AFPAUSE
AFCONT

The action of this statement is to establish initial options in the same manner as the OPTIONS command of OS/32-ST. See the OS/32-ST Program Reference Manual, Publication Number 29-380 for details on the operation of the OPTIONS command. The following error may be detected:

CUP: OPTI-ERR — Invalid or misspelled <option>

This statement may be omitted; the default is UT, FULL, AFPAUSE.

Note that the HALF option should not be specified unless the target system is configured for Halfword Mode support. This support requires a source-level SYSGEN.

3.3.12 Initial Memory Expansion. The EXPAND statement establishes the initial value for memory expansion of a newly loaded task. Its format is:

EXPAND <n>

where <n> is a decimal number ranging from zero to 4096. Whenever any task is loaded, an implicit EXPAND operator command is performed. The size of this expansion is in <n> 256-byte increments. For example, if <n> = 4, an additional 1 KB is added to the allocation of any task being loaded.

The following error may be detected:

CUP: EXP-ERR	<n> is greater than 4096.

Example:

```
EXPAND 4
```

If this statement is omitted, the default value is 4.

NOTE:

All Fortran programs require an EXPAND size of at least 1 KB (i.e.,<n>= 4).

3.3.13 Memory Access Controller. The MAC statement causes the target system to allow for the existence of a Memory Access Controller on the processor on which it is to run. The OS/32-ST system does not attempt to utilize the MAC, whether or not it is configured; however, if a MAC is configured on the target system, an area of memory X'100' bytes long, starting at the first multiple of X'100' bytes above the Interrupt Service Pointer table, can not be used. The MAC statement causes internal locations to be adjusted so that this memory is not used. The format of the statement is:

MAC

3.3.14 Floating-point Traps. The TRAPS statement causes the floating-point trap routines to be included in the system being configured. Its format is:

TRAPS

This statement should be used if the target system is to run on a processor without hardware floating-point instructions, but it is desired to run programs that use these instructions under the target system. The instructions emulated by the software trap routines are:

LE
LER
STE
CE
CER
AE
AER
SE
SER
ME
MER
DE
DER
LME
STME
FXR
FLR

3.3.15 MODULE Statement. The MODULE Statements are used to generate alternate object module names to be called by the OS Library Loader in place of those module names generated by CUP/ST. The format of these statements is:

MODULE - begin scope of MODULE statement
<module-name> - one for each alternate module name
ENDM - terminate scope of MODULE statements

where <module name> is an 8-character name starting with one of the following:

EXEC.	-	for modifications to the EXECUTIVE
CMDP.	-	for modifications to the COMMAND PROCESSOR
FMGR.	-	for modifications to the FILE MANAGER
FLTP.	-	for modifications to the FLOATING POINT TRAPS

The syntax of the <module-name> is a five character field as described above, ending in a period followed by 3 alphanumeric characters.

3.3.15.1 Source SYSGEN Use of Module Statements

Each module in the Source System Module Library is provided with a PROG, ENTRY and EQU statement with a label of the form XXXX.U00 where XXXX is the appropriate module name as specified above. If the user has made modifications to any of the OS/32-ST source modules, a module statement must be included in order to inform CUP/ST that a user modified routine, rather than an INTERDATA supplied system module is to be included in the user's system.

For example, any source modifications to the EXECUTIVE module require the following sequence of statements input to CUP/ST:

```
MODULE
EXEC.U00
ENDM
```

If the user has built a library containing more than one version of these modules, the PROG, ENTRY and EQU statements in the source must be modified at assembly time to contain a unique name of the form

XXXX.YYY

where YYY can be any three alphanumeric characters. Then for source modifications to the EXECUTIVE, the following sequence of statements must be input to CUP/ST:

```
MODULE
EXEC.YYY
ENDM
```

The MODULE statements may also be used if the user is writing additional routines to be included in the user's OS/32-ST. In this case, the MODULE statements are of the form:

USER.XXX

where XXX are any three alphanumeric characters.

A corresponding PROG, ENTRY and EQU statement with a label of the form USER.XXX must be included in the user's source code.

3.3.15.2 Object SYSGEN Use of the MODULE Statement

The MODULE Statements can also be used to obtain alternate object modules from the INTERDATA supplied system module library. In this case, the module-name field should contain the 8-character name of the object module desired. At the present time, this procedure is only required if the user wishes to obtain the FILE MANAGER module,

FMGR.S03

which contains support for Contiguous Files **only.** **If** this is omitted, the FMGR.S01 module will be included if there are directory devices in the **system;** otherwise, the FMGR.S02 module is included. Use of the MODULE statement in this way allows **the user** to incorporate this version of the FILE MANAGER without requiring a reassembly of the FILE **MANAGER.** The format is:

```
MODULE
FMGR.S03
ENDM
```

A MODL-ERR will be generated if the <module-name> fields are syntactically incorrect.

3.3.16 Queue. The Queue **statement is used** to define a size for the OS/32-ST System Queue, an internal list maintained by the system to **schedule** driver termination. The maximum size necessary for the System Queue is equal to the total **number of** devices in the system.

The format of this statement is

```
QUEUE <n>
```

where <n> is a positive number greater than or equal to 10 and ≤65,000. The following error may occur:

```
CUP: QUEU-ERR      <n> is less than 10
```

If this statement is not present, the queue size defaults to the total number of devices in the system or 10, whichever is greater.

Extreme care should be taken when establishing the size of the system queue. The driver termination routines do not check if there is sufficient room in the list when adding entries; therefore, the user should make sure the list generated can handle the number of devices generated in the system. Refer to the OS/32-ST Series General Purpose Driver Manual, Publication Number 29-384 for additional information about the system queue.

3.3.17 Assignments. If assignment statements are used, they must appear as the last statements in the configuration statement sequence. These statements establish the initial assignments that are set up at system initialization time or on the execution of a RESET command within OS/32-ST. Their format is as follows:

```
ASSIGN                          this starts the initial assignments
<lu>,<fd> [/<ap>] [,<keys>]     as many of these as are required
ENDC                            end of assignments and of configuration
```

If no initial assignments are desired, the configuration statement sequence must be terminated as follows:

```
ENDC
```

In either case the ENDC statement terminates the configuration statement sequence phase of CUP/ST.

The meaning of each element in the assignment statements follow:

<lu> Logical Unit to be **assigned,** must be less than number specified in UNITS statement.

<fd> File Descriptor or device Mnemonic to which <lu> is to be assigned; must be syntactically correct, but is not checked for reasonability.

<ap> Access privileges desired at time of assignment; default = SRW.

<keys> Keys to be specified at assignment time.

Valid access privileges are:

SRO	-	Sharable read-only
ERO	-	Exclusive read-only
SWO	-	Sharable write-only
EWO	-	Exclusive write-only
SRW	-	Sharable Read-Write
SREW	-	Sharable Read-Exclusive Write
ERSW	-	Exclusive Read-Sharable Write
ERW	-	Exclusive read-write

The keys field contains 4 hexadecimal digits XXYY where XX is the write key and YY the read key associated with the file.

The following errors may occur:

CUP:	LU-ERR	<lu> is greater than or equal to UNITS specification.
CUP:	SEPR-ERR	Missing comma following <lu>.
CUP:	FD-ERR	Syntax error in file descriptor.
CUP:	AP-ERR	Access privilege Mnemonic not recognized.

NOTE

> The fact that an assignment statement does not generate an error during the execution of CUP/ST does not ensure that the assignment will succeed at system initialization time. For example, if no default volume name is specified via a VOLUME statement, but the user is defaulting all the volume names in the assignment statements, CUP/ST does not generate assignment errors but the assignments themselves fail at system initialization time. Also at initial system load time, all discs are marked off-line; therefore, assignment of a file will fail at initial system load time. If the appropriate disc is then marked online, subsequent RESET commands will cause the proper assignment to be made.

3.4 Processing CUP/ST Output

The output of CUP/ST is a series of object-format modules which must be linked together with the desired system modules in order to produce a single absolute object-format load module. The procedure to accomplish this follows:

Load the OS/32 Library Loader (03-065).

Assign LU 5 to the console device (for the Library Loader) and one Logical Unit to each of the following: CUP/ST output; User Assembled Module Library (if any); OS/32-ST Object System Module Library, 07-061; OS/32 Library Loader output; a device supporting ASCII/Write for Library Loader MAP.

Start the Library Loader, and issue the following sequence of commands:

RW	n	(Rewind Library Loader output)
WF	n	(Write Filemark to output logical unit)
OU	n	(Indicate output LU)
BI	iiii	(Impure Bias computed by CUP/ST)
PB	pppp	(Pure Bias computed by CUP/ST)
LO	m	(Load from CUP/ST logical unit)
ED	m	(Edit from CUP/ST logical unit)
ED	u	(Edit from User System Object Module Library)*
ED	s	(Edit System Object Module Library (07-061))
XO		(End of Sequence)
MA	pr	Map of System Built

* Only if source SYSGEN has been performed.

An example of this object SYSGEN process is given in Appendix 7.

This output consists of one absolute object format load module which can be loaded using the 32-bit Relocating Loader (03-067).

The sequence of commands to the Library Loader must be executed exactly as described above; otherwise, the results are unpredictable. All user-assembled modules must be EDited before the INTERDATA supplied Object System Module Library. This is because the final object module in the library, UBOT, must be the last module linked. This provides an absolute address for the top of the operating system, to be placed in the System Pointer Table as UBOT, the bottom of the user's available memory.

4. DEVICE-DEPENDENT CONFIGURATION CHARACTERISTICS

This section gives details of how to prepare the device specification statements for all devices currently supported by OS/32 drivers. The facilities of CUP/ST may also be used to configure user-written drivers into an OS/32-ST system. See the OS/32 Series General-Purpose Driver Manual, Publication Number 29-384, for details on how such a driver should be written.

The device specification statement used to include a user-written driver is prepared in an identical fashion to that used to include a standard driver. CUP/ST includes drivers in the following manner. The DCB is included via the device code specified on the device specification statement, then the driver is included via an EXTRN in the DCB. For this reason, the user must choose a device code for each user-written driver that does not conflict with any device code that might appear on the Driver Library, otherwise the call may be satisfied by some driver on that library. See Appendix 3 for a list of all device codes currently allocated by INTERDATA; any additional device codes used in the future by INTERDATA will appear in the OS/32 Series General Purpose Driver Manual. Device codes 240-254 will not be used by INTERDATA and are reserved for user-written drivers. Device code 255 always refers to the NULL device.

CUP/ST processes the DCB/Driver library to resolve requests for drivers, based upon the device specification statements. If an End of Medium (EOM) is detected before all requests have been resolved, CUP/ST will issue an 'ENDVOL' message and PAUSE. If there are additional INTERDATA drivers on another file, the operator must assign LU4 to the file containing the remainder of the INTERDATA drivers and CONTINUE the processing of CUP/ST.

When CUP/ST has processed all of the INTERDATA drivers and there are additional outstanding requests, CUP/ST issues an 'ENDVOL' message again and PAUSEs. At this time, the user can assign LU4 to the file containing the user written DCB/Driver library and then CONTINUE. When all DCB and driver references are satisfied, CUP/ST ceases reading the file and proceeds to completion.

4.1 Teletypes and Nonediting CRT

Teletypes should not be configured on a Selector Channel. The Reader/Punch and Keyboard/Printer of ASR teletypes are considered to be separate devices; in this way KSR teletypes may be supported. Any device which is fully teletype-compatible and is connected to the normal teletype interface (including the nonediting CRT) may use the teletype driver.

Device Codes:	K/P	R/P
Model 33 TTY	016	081
Model 35 TTY	017	082
Nonediting CRT	018	---

If both keyboard/printer and reader/punch facilities are desired in the same device, a "shared-busy" condition exists. The keyboard/printer and reader/punch specifications must be consecutive, with the second specification containing an asterisk (*) in the <level> field. Failure to abide by this restriction may cause interference (garbling) between the keyboard/printer and reader/punch.

Model 33 and 35 TTY Keyboard/Printer and Nonediting CRT are the only devices permitted as console devices. Therefore, the <flags> field may contain a C for the console device.

Examples:

```
1 TTY1:2,016,C        Model 33, console device, dn = X'002'
* TRP1:2,081          Reader-punch, same device
```

NOTE

The reader/punch facility of an ASR TTY configured as the console device, may not be assigned to a user task Logical Unit.

4.2 Intertape Cassettes

Cassettes should not be configured on a Selector Channel. The two cassettes in a transport are interlocked in hardware; consequently, a shared-busy condition exists. The two cassettes in a transport must be specified in consecutive statements; the second statement should contain an asterisk (*) in the level field. Failure to abide by this restriction causes interference between the two cassettes, resulting in the possible loss of data and incurring the risk of system failure.

Device Code:	CASSETTE
	066

The <flags> field may not be used for these devices.

Examples:

```
1 CAS1:45,066         The two cassettes in the same transport,
* CAS2:55,066         having addresses of X'045' and X'055'.
```

4.3 Card Readers

Card readers should not be configured on the Selector Channel. The level field should always contain 1. The <flags> field should not be used.

Device Codes:	CARD READER
	096 - Card Reader without automatic Hollerith/ASCII conversion 097 - Card Reader with automatic Hollerith/ASCII conversion

Example:

```
1 CARD:4,96       Card Reader without Hollerith/ASCII conversion
1 CARD:4,97       Card Reader with Automatic Hollerith/ASCII conversion
```

4.4 Line Printers

Line printers should not be configured on the Selector Channel. The <level> field should always contain 1. The <flags> field should not be used.

Device Codes:	LINE PRINTER
	112 200 LPM 114 600 LPM

Examples:

```
1 LPR1: 62, 112          Low speed (200 LPM) printer
1 LPR2: 72, 114          High speed (600 LPM) printer
```

4.5 High Speed Paper Tape Reader/Punch

High speed paper tape reader/punches should not be configured on the Selector Channel. The <level> field should contain 1. The <flags> field should not be used.

Device Code:	HSATA/P
	080

Example:

```
1 PTRP: 13, 80
```

4.6 Selector Channel

The Selector Channel is not a device, but a channel. As such, it must always be on level 1. All devices or controllers directly controlled by the Selector Channel must be on level 2. Specification statements for these devices or controllers must directly follow the Selector Channel specification statements; the next specification statement found with a level of 1 signals to CUP/ST the end of the Selector Channel's control.

A Selector Channel should not have a dm in its specification; there is no device code, so dcod should be set to zero. Selector Channels do have device addresses; this must be entered in the dn field.

Example:

```
1:0F0,0          selector channel,     dn = X'F0'
```

4.7 Magnetic Tapes

Magnetic tapes are always configured on a Selector Channel. Between the channel and the tape transports is a Magnetic Tape Controller, supporting up to four transports. Although this controller has no device address, it is not fully transparent to the driver and must therefore be accounted for by a device specification statement.

The controller is always at level 2; the tapes controlled by it are therefore at level 3. The specification statement for each tape controller must directly precede the statements for all tapes controlled by that controller.

The <dcod> and <dn> fields of the controller statement should contain 0. The <dm> field of the controller statement should be omitted. The <flags> field of both controller and tape statements should be omitted.

Device codes:	MAG TAPE
064	800 BPI
065	1600 BPI (phase-encoded)

Examples: (assume Selector Channel is on level 1)

```
2: 0, 0             Controller #1
3 MTP1: 85, 64        Tape #1  (800 BPI)
3 MTP2: 95, 64        Tape #2  (800 BPI)
2: 0, 0             Controller #2
3 MTP3: C5, 65        Tape #3  (1600 BPI)
3 MTP4: D5, 65        Tape #4  (1600 BPI)
```

4.8 Moving-Head Discs

Moving-head discs are always configured on a Selector Channel. Between the Selector Channel and the disc itself is the Disc Controller. Each controller may control up to four disc drives. All INTERDATA-supported discs except the M46-416 (10-MB disc) are organized with one volume per disc drive. However, the 10-MB disc is in fact composed of two 5-MB volumes. Since these volumes share the same drive mechanism, a shared-busy condition exists. Therefore, for each 10-MB disc, two volumes must be specified consecutively, with the second specification containing an asterisk (*) in the <level> field. Failure to abide by this restriction will cause positioning errors on either volume, leading to loss of data and to the probability of eventual system failure.

The <flags> field must contain a D for each volume. This notifies the system to create control structures for each volume's directory and bit-map, and to provide direct-access support.

The specification statement for each disc controller must directly precede the statements for all discs that the controller controls. The <dcod> field of this statement should contain zero. The <dm> field should be left out.

Examples:

```
2:B6,0                 Controller on level 2 (SELCH is on level 1)
3 DSC1:C6,049,D          2.5 MB removable disc
3 DSC2:D6,049,D          2.5 MB removable disc
2:B8,0                 New controller
3 DSC3:C8,051,D          5 MB removable disc
* SYS1:C9,050,D          5 MB fixed disc
3 DSC4:D8,051,D          5 MB removable disc
* SYS2:D9,050,D          5 MB fixed disc
```

Note that the <dm> field of the disc specifications establishes the name of a disc device rather than of a disc volume. This name is only used in OS/32-ST under conditions when there is no volume on-line; if a volume is on-line on any particular disc device, the volume name should be used.

Device Codes:	DISC
049	2.5 MB removable platter
050	5 MB fixed platter
051	5 MB removable platter
052	41 MB removable platter

5. SOURCE LEVEL SYSTEM GENERATION

This section describes to the user the available source-level options and the procedure for performing a source level system generation.

The Source SYSGEN is necessary for one of the following two reasons:

> The user is taking advantage of the source SYSGEN option to obtain an alternate object module to the ones provided in the object library supplied by INTERDATA, or
>
> The user has made modifications to the OS/32-ST code in either the EXECUTIVE, COMMAND PROCESSOR or FILE MANAGER.

5.1 Source SYSGEN Options

The following is a list of the SYSGEN parameters available at the source level and the modules affected by each:

The SYSGEN options for the starter system are underlined:

SGN.HWRD — 0 - Eliminate code supporting halfword mode
1 - Include code supporting halfword
modules affected: EXECUTIVE, COMMAND PROCESSOR

SGN.SAFE — 0 - Eliminate code which performs system integrity checks
1 - Include code to perform system integrity checks
module affected: EXECUTIVE

SGN.JRNL — 0 - Replace JOURNAL subroutine with BR R8
1 - Include JOURNAL subroutine
module affected: EXECUTIVE

SGN.BCMD — 0 - Exclude Bulk file command support
1 - Include Bulk file command support
module affected: COMMAND PROCESSOR

SGN.DA — 0 - No direct-access support
1 - Support for direct-access devices included
modules affected: COMMAND PROCESSOR, FILE MANAGER

SGN.CO — 0 - Contiguous file support excluded
1 - Contiguous file support included
module affected: FILE MANAGER

SGN.CHN — 0 - Chain file support excluded
1 - Chain file support included
module affected: FILE MANAGER

SGN.CSS — 0 - no CSS support
1 - CSS support included
module affected: COMMAND PROCESSOR

5.2 Reassembling the OS/32-ST

The SYSGEN options listed in Paragraph 5.1 are all contained in the Parameters and Control Blocks Module (07-063). The procedure to modify OS/32-ST at the source level is as follows:

1. Edit the library containing the Parameters and Control Blocks Module and make the necessary changes.

2. Assign the Library containing the Parameters and Control Blocks Module to Logical Unit 7 and reassemble the affected modules using the CAL Assembler, 03-066.

3. Run CUP/ST including MODULE statements for the modules modified.

4. Link the output of CUP/ST, using the OS Library Loader, with the reassembled system modules and the Object System Module Library (07-061) as described in Section 3.4.

5. Load this final absolute object-load module with the 32-Bit Relocating Loader.

For example, suppose a user wishes to generate an OS/32-ST with all source SYSGEN options as in the STARTER system except that the target system should exclude Bulk file command support from the COMMAND PROCESSOR. The procedure would be to:

1. Set SGN.BCMD to 0 in the Parameters and Control Blocks Module.

2. Assign the Parameters and Control Blocks Module Library to LU 7 and reassemble the COMMAND PROCESSOR.

3. Include the following statements in the Configuration Statement Sequence Input to CUP/ST:

   ```
   MODULE
   CMDP.U00
   ENDM
   ```

4. Process the CUP/ST output as previously defined.

APPENDIX 1

CONFIGURATION OF OS/32 ST STARTER

The size of the OS/32-ST STARTER system is 37KB, based upon the following:

Modules included:

EXECUTIVE
COMMAND PROCESSOR
FILE MANAGER
UBOT

Options included in the STARTER system and the modules affected:

Direct Access Support, Contiguous Files only (FILE MANAGER, COMMAND PROCESSOR)
Bulk file command Support (COMMAND PROCESSOR)
CSS Support (COMMAND PROCESSOR)
Journal Support (EXECUTIVE)

Options excluded:

FLOATING POINT TRAPS
SYSTEM INTEGRITY
HALFWORD MODE

Devices included are:

Model 33 TTY Keyboard/Printer
High Speed Paper Tape Reader/Punch
Card Reader without conversion
Line Printer
800 BPI Magnetic Tape
2.5 MB removable disc
2 Cassette Tapes

APPENDIX 1 (Continued)

CONFIGURATION OF OS/32 ST STARTER

CUP Configuration Statements for Starter System

```
QUEUE 10                         (same as default)
EXPAND 4                         (same as default)
MAC
UNITS 16                         number of logical units = 16
                                 (same as default)
MEMORY 48                        OS/32-ST system size   48KB
                                 (same as default)
DEVADS 3                         1024 device addresses
                                 (same as default)
JOURNAL 6                        6 journal entries (same as default)
CSS 3                            maximum nesting depth = 3 (same as
                                 default)
CMDLEN 80                        System Command Buffer Length = 80 bytes
                                 (same as default)
LOGLEN 72                        Maximum size of Log Message Buffer =
                                 72 bytes (same as default)

MODULE
FMGR.S03                         File Manager with Contiguous only
ENDM

DEVICES
1 TTY:2,16,C                     Teletype (M33) (Console)
1:F0,0                           Selector Channel
2:B6,0                           Disc Controller
3 DISC:C6,49,D                   2.5 Mb removable
2:0,0                            Magnetic Tape controller
3 MAG1:85,64                     800 BPI tape
1 CR:4,96                        Card reader with automatic Hollerith/
                                 ASCII conversion
1 PR:62,112                      Centronics printer
1 PTRP:13,80                     High Speed Paper Tape Reader/Punch
1 CAS1:45,66                     Cassette #1
* CAS2:55,66                     Cassette #2
ENDD
OPTIONS UT, FULL, AFPAUSE        (same as default)
ASSIGN
0,TTY:
1,CR:
2,PTRP:
3,PR:
5,TTY:
ENDC
```

APPENDIX 2

RE-CONFIGURING THE STARTER SYSTEM

If the peripheral device addresses assigned in the STARTER system differ from the user's configurations, it will be necessary to manually change these addresses, either from the display panel (if the console address must be changed) or from the console for the remainder of the devices. The procedure to modify these addresses applies only to the STARTER System.

The procedure to modify the console address is as follows:

1. Obtain the address of the Device Mnemonic Table from the Library Loader MAP of the STARTER system (Entry-point, DMT; refer to the example found in Appendix 7).
2. Add X'C' to this address.
3. Read from Memory the fullword contents specified by this address to obtain the DCB address.
4. Add X'2E' to this address.
5. Modify the halfword at this address to contain the new physical address of console device.

For the remainder of the devices, the console can be used for the modifications. In this case the procedure is:

1. Obtain the DMT address as above.
2. Add n to this address (n is defined in the table below for all devices in STARTER).
3. Read from Memory the fullword contents specified by this address to obtain the DCB address.
4. Add X'2E' to this address.
5. Modify the halfword at this address to contain the new physical address of the device.

Structure of the DMT for the STARTER System:

Device	DCB Address Displacement (Hexadecimal)
TTY (M33) (console)	X'C'
2.5 MB Disc	X'14'
800 BPI Tape	X'1C'
CARD Reader	X'24'
LINE Printer	X'2C'
Paper Tape Reader/Punch	X'34'
Cassette #1	X'3C'
Cassette #2	X'44'

STARTER is configured for 48 KB. If more memory is present, it may be utilized by modifying MTOP. Find the location of MTOP by referring to the label SPT.MTOP on the Library Loader Map of the STARTER system (refer to the example found in Appendix 7). Modify the MTOP fullword to contain the size of memory and restart the system.

APPENDIX 3

DEVICE CODES

Device codes 000 to 015 are reserved for direct-access file structures and as such do not refer to any specific type of device.

016	M33 Teletype keyboard/printer, TTY interface
017	M35 Teletype keyboard/printer, TTY interface
018	Nonediting CRT, TTY interface
019	Editing CRT, TTY interface
020	Tektronix 4010, TTY interface
032	M33 Teletype keyboard/printer, PALS or PASLA interface
033	M35 Teletype keyboard/printer, PALS or PASLA interface
034	Nonediting CRT, PALS or PASLA interface
035	Editing CRT, PALS or PASLA interface
036	Graphics CRT, PALS or PASLA interface
048	2.5 MB disc, fixed platter
049	2.5 MB disc, removable platter
050	5 MB disc, fixed platter
051	5 MB disc, removable platter
053	41 MB disc
064	800 BPI magnetic tape
065	1600 BPI magnetic tape
066	INTERTAPE cassette
080	High speed paper tape reader/punch
081	M33 Teletype reader/punch, TTY interface
082	M35 Teletype reader/punch, TTY interface
096	Card Reader without automatic Hollerith/ASCII conversion
097	Card Reader with automatic Hollerith/ASCII conversion
112	200 LPM line printer
114	600 LPM line printer
255	NULL device

Device codes 240-254 are reserved for user-written drivers.

APPENDIX 4

MEMORY REQUIREMENTS

System Modules:

EXECUTIVE: without JOURNAL, INTEGRITY CHECKS HALFWORD support	8.25 KB
JOURNAL support adds	100 bytes
+ for each entry	20 bytes
SYSTEM INTEGRITY CHECKS add	.5 KB
HALFWORD support adds	1.2 KB
COMMAND PROCESSOR: without CSS, HALFWORD, Bulk file commands and Direct-Access support	6.5 KB
CSS support adds	1.5 KB
+ number levels * (buffer length +10)	
HALFWORD support adds	300 bytes
Bulk Command support adds	250 bytes
Direct-Access support adds	2.25 KB
FILE MANAGER: without any Direct-Access support	1.7 KB
Contiguous-file support adds	4.5 KB
Chain file support adds	4 KB
Contiguous and chain file support adds	8.5 KB
FLOATING-POINT TRAPS	1.8 KB

Drivers (1 for each type of device; 1 DCB for each device):

TTY Keyboard/Printer	1050 bytes
DCB/CCB	104 bytes
TTY Reader/Punch	750 bytes
DCB/CCB	104 bytes
Card Reader	600 bytes
DCB/CCB (with Hollerith/ASCII translation)	104 bytes
DCB/CCB (without Hollerith/ASCII translation)	262 bytes
Line Printer	360 bytes
DCB/CCB	104 bytes
INTERTAPE Cassette	900 bytes
DCB/CCB	104 bytes
9-Track Magnetic Tape	800 bytes
DCB/CCB	108 bytes
Moving Head Disc	1200 bytes
DCB/CCB	776 bytes
High Speed PTRP	1250 bytes
DCB/CCB	104 bytes

Control Blocks for Assigned files:

Contiguous File Control Block (FCB)	208 bytes
Chain File Control Block (FCB) plus twice physical block size	268 bytes

APPENDIX 5

LIST OF CONFIGURATION STATEMENTS

Statement		Format	Meaning	Defaults
3.3.17	ASSIGN	ASSIGN <lu>, <fd> [/<ap>] [,<keys>] • • • ENDC	Set initial assignments	none
3.3.8	CMDLEN	CMDLENΔ <n>	Set length of command and CSS buffers	80 bytes
3.3.7	CSS	CSS Δ <n>	Set maximum CSS nesting depth	3 levels
3.3.3	DEVADS	DEVADSΔ <n>	Specify maximum device addressing range	n=3 (max. addr. X'3FF')
3.3.10	DEVICES	DEVICES <level> [<dm>] :<dn>, <dcod>[, flags] • • • ENDD	Establish device descriptions	none
3.3.6	DISCBLOCK	DISCBLOCKΔ <n>	Set maximum Chained Physical Block Size	4 sectors
3.3.16	ENDC	ENDC	End Configuration Statement Input	none
3.3.10	ENDD	ENDD	End Devices Statements	none
3.3.15	ENDM	ENDM	End Module Statements	none
3.3.12	EXPAND	EXPANDΔ n	Value for Initial Memory Expansion	n=4 (1 KB)
3.3.5	JOURNAL	JOURNALΔ <n>	Specify number of Journal entries	6 entries

APPENDIX 5 (Continued)

LIST OF CONFIGURATION STATEMENTS

Statement		Format	Meaning	Defaults
3.3.9	LOGLEN	LOGLEN△ <n>	Specify maximum size of Log Message Buffer	72 bytes
3.3.13	MAC	MAC	Allow for Memory Access Controller	no MAC
3.3.2	MEMORY	MEMORY△ <n>	Specify available memory	48K
3.3.15	MODULE	MODULE module-name • • • ENDM	Establish alternate module names	none
3.3.11	OPTIONS	OPTIONS[△ options]	Set initial task options	UT FULL AFPAUSE
3.3.16	QUEUE	QUEUE <n>	Establish system queue size	Number of devices in system
3.3.14	TRAPS	TRAPS	Include floating point traps	no Traps
3.3.1	UNITS	UNITS△ <n>	Specify number logical units	16 logical units
3.3.4	VOLUME	VOLUME △<voln>	Specify initial system volume	none

APPENDIX 6

LIST OF ERROR MESSAGES

Message	Meaning	Section (s)
CUP:MNEM-ERR	Mnemonic error	3.3.1-3.3.17
CUP:SEQ-ERR	Statement entered more than once	3.3.1-3.3.17
CUP:LU-ERR	LU - specified less than 1 or greater than 255 on UNITS statement;	3.3.1
	LU in ASSIGN greater than or equal to UNITS specification	3.3.16
CUP:MEM-ERR	Memory size specified greater than 1024, less than 32 or not divisible by 16	3.3.2
CUP:DN-ERR	DEVADS statement specified illegal addressing range; DEVICES statement specified illegal device number	3.3.3 3.3.10
CUP:VOL-ERR	Volume syntax incorrect	3.3.4
CUP:JRNL-ERR	Number journal entries not positive integer	3.3.5
CUP:BLK-ERR	Disc Block size greater than 255 or equal to 0	3.3.6
CUP:CSS-ERR	Number of CSS levels greater than 249	3.3.7
CUP:LGTH-ERR	Command Buffer Length less than 32 or greater than 1024;	3.3.8
	Log Message Buffer Length less than 32 or greater than 132	3.3.9
CUP:NODV-ERR	The ENDD statement directly follows the DEVICES statement or no device statements were found at all	3.3.10
CUP:MFUL-ERR	CUP-ST ran out of memory while processing device statements	3.3.10
CUP:LEVL-ERR	Device level is zero, or greater than the previous level plus one	3.3.10
CUP:SHAR-ERR	DEVICES field <level> contains an asterisk, but either there is no previous statement, or the previous statement refers to a controller or channel rather than a device.	3.3.10
CUP:DM-ERR	The previous DEVICES statement did not refer to a controller or channel, yet it had no <dm> field.	3.3.10
CUP:SEPR-ERR	Separator incorrect in DEVICES or ASSIGN statement.	3.3.10 3.3.17
CUP:DCOD-ERR	Device <dcod> field contains other than a decimal number between 16 and 255.	3.3.10
CUP:FLAG-ERR	Device flags field contains something other than a C or a D.	3.3.10

APPENDIX 6 (Continued)

LIST OF ERROR MESSAGES

Message	Meaning	Section (s)
CUP:CONS-ERR	More than one device or none was flagged as the Console device on DEVICES statements.	3.3.10
CUP:OPTI-ERR	Syntax error on Options Statement	3.3.11
CUP:EXP-ERR	Expand Memory size greater than 4096	3.3.12
CUP:MOD-ERR	Module sequence syntactically incorrect	3.3.15
CUP:QUEU-ERR	System Queue size less than 10	3.3.16
CUP:FD-ERR	Syntax error in file descriptor field of ASSIGN statement	3.3.17
CUP:NMER-ERR	Direct-Access flags on DEVICES statements incorrect	3.3.10
CUP:TRNC-ERR	DCB/Driver library built incorrectly	4.
CUP:FRMT-ERR	DCB/Driver library contains illegal loader items	4.
CUP:NTRY-ERR	DCB/Driver library contains incorrect linkages	4.
CUP:CKSM-ERR	Checksum error on DCB/Driver library. Probable cause of error: DCB/Driver library requires rewinding.	4.
CUP:LSEQ-ERR	Sequence error on DCB/Driver library. Probable cause of error: DCB/Driver library requires rewinding.	4.

APPENDIX 7

SYSTEM GENERATION EXAMPLE

```
*
*  EXAMPLE OF SYSTEM GENERATION  OS/32 ST STARTER
*
LOAD PTRP:             ;*   LOAD CUP/ST FROM PAPER TAPE
AS 1,CR:               ;*   ASSIGN INPUT FROM CARD READER
AS 2,CUP.OUT           ;*   ASSIGN CUP OUTPUT TO CONTIG FILE
AS 3,TTY:              ;*   ASSIGN LIST OUTPUT TO CONSOLE TTY
AS 4,MAG1:             ;*   ASSIGN DRIVER LIBRARY TO MAG TAPE
START                  ;*   START CUP/ST


CONFIGURATION UTILITY PROGRAM - OS/32 ST
CONFIGURATION STATEMENT INPUT:
MAC                                   IN CASE MAC IS PRESENT
UNITS 16                              (SAME AS DEFAULT)
VOLU PACK
MEM 48
DEVADS 3                          255 DEVICE ADDRESSES (SAME AS DEFAULT)
JOURNAL  6                       (SAME AS DEFAULT)
CSS 3
CMDLEN 8Ø                             (SAME AS DEFAULT)
LOGLEN 72                             (SAME AS DEFAULT)
MODULE
FMGR.SØ3
ENDM
DEVICES
1 TTY: 2,16,C                         TELETYPE (M33) (CONSOLE)
1: FØ,Ø                               SELECTOR CHANNEL
2: B6,Ø                               DISC CONTROLLER
3 DISC:C6,49,D                        2.5 MB REMOVABLE
2:Ø,Ø                                 MAGNETIC TAPE CONTROLLER
3 MAG1:85,64                          8ØØ BPI TAPE
1 CR:4,96
1 PR:62,112                          LOW SPEED LINE PRINTER
1 PTRP:13,8Ø                          HIGH SPEED PAPER TAPE READER/PUNCH
1 CAS1:45,66                          CASSETTE #1
* CAS2:55,66                          CASSETTE #2
ENDD
OPTIONS UT,FULL,AFPAUSE               (SAME AS DEFAULT)
ASSIGN
3,PR:
5,TTY:
ENDC
CONFIGURATION STATEMENT PROCESSING COMPLETE
SET IMPURE BIAS TO      133C
SET PURE BIAS TO      128C
```

```
  OS/32 ST CONFIGURATION
  MAXIMUM LOGICAL UNIT NUMBER =  16
  MEMORY SIZE =     48KB
  MAXIMUM DEVICE ADDRESS = 3FF
  DEFAULT VOLUME NAME IS 'PACK'
 *        MAXIMUM BLOCK SIZE FOR CHAINED FILES =    4 SECTORS
  CSS INCLUDED
          MAXIMUM CSS LEVELS =    3
  COMMAND BUFFER LENGTH =    82
  LOG MESSAGE BUFFER LENGTH =    72
  CONSOLE DEVICE: TTY  AT   2
  OPTIONS:
 *        UT
 *        AFPAUSE
 *        FULLWORD
 *EXPANSION SIZE =        4 BLOCKS (    1024 BYTES)
 SET PURE BIAS TO      128C
 SET IMPURE BIAS TO     133C

 END OF TASK      Ø
 *
 *  CUP/ST FINISHED PROCESSING - LINK FINAL STARTER
 *
 LOAD MAG1:                ;*   LOAD OS/32 LIBRARY LOADER FROM MAG TAPE
 AS 1,SYSLIB               ;*    ASSIGN SYSTEM MODULE LIBRARY (CONTIG FILE)
 AS 2,CUP.OUT              ;*    ASSIGN OUTPUT OF CUP
 AS 3,TTY:                 ;*    ASSIGN LIST DEVICE TO CONSOLE TTY
 AS 4,MAG1:                ;*    ASSIGN MAG TAPE FOR OUTPUT OF LIB LOADER
 AS 5,TTY:                 ;*    ASSIGN COMMAND INPUT TO CONSOLE TTY
 START                     ;*    START THE LIBRARY LOADER
  LIBLDR
 >RW 4
  LIBLDR
 >WF 4                     WRITE A FILE MARK ON OUTPUT DEVICE
  LIBLDR
 >OU 4                     SPECIFY OUTPUT LU
  LIBLDR
 >BI 133C                  SET IMPURE BIAS AS SPECIFIED BY CUP/ST
  LIBLDR
 >PB 128C                  SET PURE BIAS
  LIBLDR
 >LO 2
  LIBLDR
 >ED 2
  EOF
  LIBLDR
 >ED 1
  LIBLDR
 >XO
  LIBLDR
 >MA 3
```

```
PROGRAMS:
  ØØ133C EXTERNS     ØØ133C DMT         ØØ139C EVT         ØØ155Ø IVTBL
  ØØØAØØ DCBØ16ØØ    ØØ18FØ DCBØ49ØØ    ØØØA1C INITTYKP    ØØØØØØ EXEC.SØ1
  ØØØB8Ø INITMHD     ØØØC3C DCBØ64ØØ    ØØ5214 FMGR.SØ3    ØØØC58 INITMAG
  ØØØCF8 DCBØ66ØØ    ØØØD14 DCBØ66Ø1    ØØØF68 DCBØ8ØØØ    ØØØD3Ø INITCASS
  ØØ282Ø DCBØ96ØØ    ØØØF84 INITPTRP    ØØ29ØC INITCARD    ØØ11DØ DCB112ØØ
  ØØ11EC INITLPTR    ØØ2BDØ CLEANUP     ØØ2DCC TCBTAB      ØØ6A3Ø CMDP.SØ3
  ØØ933C UBOT        ØØ973C


ENTRY-POINTS:
  ØØ6A3C INITCMD     ØØ151C SNOD        ØØ2BDØ PTRSTACK    ØØ2DDØ SPT.STCB
  ØØ133C DMT         ØØ1344 DMTØØØØØ    ØØ49E2 IODONE2     ØØ134C DMTØØØØ3
  ØØØØDØ ISPTAB      ØØ1354 DMTØØØØ5    ØØ6A3Ø TERMCMD     ØØ135C DMTØØØØ6
  ØØ4BC2 NSEVREL     ØØ1364 DMTØØØØ7    ØØ3AB6 S21PAUSE    ØØ136C DMTØØØØ8
  ØØ12FC SPT.AFSV    ØØ1374 DMTØØØØ9    ØØ12A8 SPT.CHBK    ØØ137C DMTØØØ1Ø
  ØØ1292 SPT.CRSH    ØØ138C VMT         ØØ139C EVT         ØØ139C NODEØØØØ
  ØØ14DC NODEØØØ1    ØØ14F4 NODEØØØ2    ØØ13BC NODEØØØ3    ØØ15Ø8 NODEØØØ4
  ØØ13DC NODEØØØ5    ØØ13FC NODEØØØ6    ØØ141C NODEØØØ7    ØØ143C NODEØØØ8
  ØØ145C NODEØØØ9    ØØ147C DMLV        ØØ149C NODEØØ12    ØØ14BC NODEØØ13
  ØØ155C IVTBL       ØØ1594 DCBØ16ØØ    ØØ43DØ TMSTART     ØØ15EØ INITTYKP
  ØØ12A4 SPT.CSBF    ØØØA1C ISR1TYKP    ØØ16B8 TERMTYKP    ØØ16EE XLTTYKP
  ØØ1CFØ CMDMHD      ØØ484A EVGCON      ØØ18FØ DCBØ49ØØ    ØØ1CØC $INITMHD
  ØØ4BDC EVRTE       ØØ31ØØ EXEC.SØ1    ØØ8E18 LOADSTAT    ØØ1BE8 INITMHD
  ØØ12AØ SPT.CSLV    ØØ12DØ SPT.CTCB    ØØ12AC SPT.CTOP    ØØ4D56 CANEOJ
  ØØ1D82 TERMMHD     ØØ3234 IIH         ØØ2Ø18 DCBØ64ØØ    ØØ21ØE CMDMAG
  ØØ3238 III         ØØ49DA IODONE      ØØ2Ø68 INITMAG     ØØ129E SPT.CTSP
  ØØ12D4 SPT.DMT     ØØ12B8 SPT.FBOT    ØØ6A3Ø CMDP.SØ3    ØØ21C8 TERMMAG
  ØØ2384 INITCASS    ØØ22EC DCBØ66ØØ    ØØ69E8 DCB255ØØ    ØØ92FC DCBCMD
  ØØ2338 DCBØ66Ø1    ØØ24Ø6 CMDCASS     ØØ2474 TERMCASS    ØØ24A4 DCBØ8ØØØ
  ØØ5A7C DMTLOOK     ØØ24FØ INITPTRP    ØØ1294 SPT.FLV     ØØ6A4E INITCMD$
  ØØ25D2 TERMPTRP    ØØ262Ø XLTPTRP     ØØ282Ø DCBØ96ØØ    ØØ5214 FMGR.SØ3
  ØØ291Ø INITCARD    ØØ123C SPT.INIT    ØØ4A5E EVDIS       ØØ29AØ TERMCARD
  ØØ2AA8 DCB112ØØ    ØØ5652 GETSECTR    ØØ2AF4 INITLPTR    ØØ12C8 SPT.IVT
  ØØ49D4 EVMOD       ØØ2BA2 TERMLPTR    ØØ2BDØ CLEANUP     ØØ2BE4 CMDBUFFS
  ØØ2D7E LGMBUFF     ØØØØ6Ø SPT         ØØ12E4 SPT.JRNL    ØØ1298 SPT.LLV
  ØØ6A8C COMMAND     ØØ57FA RELEB       ØØ2DCC TCBTAB      ØØ2DD4 SPT.UTCB
  ØØ3Ø5Ø JRNLBKS     ØØ3ØDØ SQ          ØØ4AF6 EVREL       ØØ129C SPT.MLBL
  ØØ12BC SPT.MTOP    ØØ12D1 SPT.NTCB    ØØ12CØ SPT.OSID    ØØ12EC SPT.PSV
  ØØ12E8 SPT.RC      ØØ12F4 SPT.RSV     ØØ12EØ SPT.SNOD    ØØ12DC SPT.SVOL
  ØØ12F8 SPT.TSV     ØØ12CC SPT.TTAB    ØØ12B4 SPT.UBOT    ØØ12BØ SPT.UTOP
  ØØ12D8 SPT.VMT     ØØ5AA8 VMTLOOK     ØØ46A4 TMREMW      ØØ449A TMRSAIN
  ØØ45A4 TMRSAOUT    ØØ44B6 TMRSOUT     ØØ5214 SVC7        ØØ8126 CMDLR
  ØØ973C UBOT


 COMMON-BLOCKS:
 NONE


 UNDEFINED:
 NONE
 LIBLDR
>EN

END OF TASK      Ø

*RESET CLOSE                ;* CLOSE ALL FILES BEFORE RELOADING
*
**
```

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From ______________________________ Date ______________________

Title ______________________________ Publication Title ______________________

Company ______________________________ Publication Number ______________________

Address ______________________________
____
____

CUT ALONG LINE

FOLD FOLD

Check the appropriate item.

Error (Page No.___, Drawing No. _________)
Addition (Page No.___, Drawing No. _________)
Other (Page No.___, Drawing No. _________)

Explanation:

FOLD FOLD

Fold and Staple
No postage necessary if mailed in U. S. A.

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT No. 22
OCEANPORT , N.J.

BUSINESS REPLY MAIL

NO POSTAGE NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY:

INTERDATA®

2 Crescent Place, Oceanport, New Jersey 07757

TECH PUBLICATIONS DEPT. MS 53

FOLD FOLD

STAPLE STAPLE