96768290

CONTROL DATA CORPORATION

INTERACTIVE TERMINAL-ORIENTED SYSTEM (ITOS) VERSION 1 REFERENCE MANUAL

CDC® COMPUTER SYSTEMS:
CYBER 18 MODELS 10M AND 20

REVISION RECORD

REVISION	DESCRIPTION
01	Preliminary manual released.
(4/77)	
A	Manual released. Obsoletes all previous editions.
(6/77)	
B	Version 1.1, obsoletes all previous editions.
(10/77)	
C	Manual revised to reflect version 1.2 system update. This revision obsoletes all previous editions.
(3/78)	

Publication No.
96768290

REVISION LETTERS I, O, Q AND X ARE NOT USED

Printed in the United States of America

Address comments concerning this manual to:
Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
Cover	—								
Title page	—								
ii	C								
iii/iv	C								
v thru viii	C								
1-1 thru 1-4	C								
2-1 thru 2-10	C								
3-1 thru 3-24	C								
4-1 thru 4-11	C								
5-1 thru 5-5	C								
6-1 thru 6-4	C								
7-1	C								
7-2	C								
A-1 thru A-5	C								
B-1	C								
B-2	C								
C-1	C								
D-1	C								
E-1 thru E-10	C								
F-1 thru F-7	C								
G-1	C								
G-2	C								
H-1 thru H-3	C								
Index-1 thru Index-4	C								
Comment sheet	C								
Cover	—								

PREFACE

The CDC® Interactive Terminal-Oriented System (ITOS) Version 1.2 for CYBER 18 consists of an ITOS executive and utilities that operate under MSOS Version 5.0 in a CYBER 18-10M or 18-20 hardware environment.

The ITOS executive provides sequencing and control of terminal user programs by interfacing them to the necessary MSOS programs. MSOS 5 includes the monitor, drivers, the batch processing control programs, and MSOS utilities. As a part of ITOS, the user is also supplied with File Manager Version 2.0 and with a sort module. Application programs for ITOS may be written in any of the following languages:

- RPG II Version 2.0
- MS FORTRAN Version 3A/B
- MSOS Macro Assembler

The user has the option of buying one or more of the standard application (manufacturing and distribution system) modules to satisfy his application needs. These modules are independent user programs and are compatible with other application programs that the user may generate to satisfy a particular need.

This manual is written for the reader who has an interest in modifying or expanding application programs operating under ITOS. It is assumed that the reader is familiar with MSOS 5 and with File Manager Version 2.0. The reader should also be familiar with the language intended for use in writing the new application programs: RPG II, MS FORTRAN, or macro assembler.

The following CYBER 18 manuals contain information useful to ITOS users:

Publication	Publication Number
MSOS Version 5 Reference Manual	96769400
File Manager Version 2.0 Reference Manual	96768040
RPG II Version 2.0 Reference Manual	96768710
MS FORTRAN Version 3A/B Reference Manual	60362000
Macro Assembler Reference Manual	60361900
Software Peripheral Drivers Reference Manual	96769390
MSOS 5 Installation Handbook	96769410
MSOS 5 Instant	96769430
MSOS 5 Diagnostic Handbook	96769450
Order Entry/Invoicing 1 Reference Manual	96769160
Accounts Receivable 1 Reference Manual	96769130
Inventory Control 1 Reference Manual	96769120
Accounts Payable 1 Reference Manual	96769140
Payroll 1 Reference Manual	96769180
General Ledger 1 Reference Manual	96769150
Routing 1 Reference Manual	96768760
Bill of Materials Processor 1 Reference Manual	96768770

Publication	Publication Number
Material Requirements Planning 1 Reference Manual	96768780
Purchase Order Processing 1 Reference Manual	96769170
Physical Inventory 1 Reference Manual	96769190
Work in Process 1 Reference Manual	96768790

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

INTRODUCTION 1

This section of the manual provides a general introduction to the ITOS system. Section 2, User Terminal Procedures, and 3, Utilities, contain information on operating the system as well as systems characteristics. The remaining sections of the manual contain information only on the externals of the system.

This reference manual describes ITOS and its utilities. The ITOS system operates under MSOS in a CYBER 18-10M or 18-20 computer hardware environment (see figure 1-1) and provides support for a series of terminals that are used interactively on a timeshared basis.

Typically, to perform a task, the user at his terminal selects one of the standard application modules licensed for his system. A list of these application modules is given in table 1-1. After the operator selects the application module to be run, ITOS displays a list (menu) of the tasks that the module can perform. The operator selects a task and ITOS begins an interactive dialogue with the operator during execution of the task. In general, each task consists of a sequence of operations and programs, known as procedures.

ITOS informs the operator as each portion of the procedure begins execution. In most cases, ITOS then requests the operator to enter data or further instructions by means of messages displayed on the terminal screen. The data displays are formatted to simulate the type of business forms normally used to maintain records. Some of the reports generated are also delivered to the terminal screen. Other reports may be sent to the system printer, and may be formatted to be output on preprinted forms (for instance, payroll data to be printed on check registers).

When a task is completed, the operator is notified. ITOS then requests that the operator enter the name of the next task.

After the operator becomes thoroughly familiar with the system, he can request tasks directly by entering the mnemonic for that task rather than requesting a task from the menu. Provision is also made for the operator to recover from errors that occur during the running of a task.

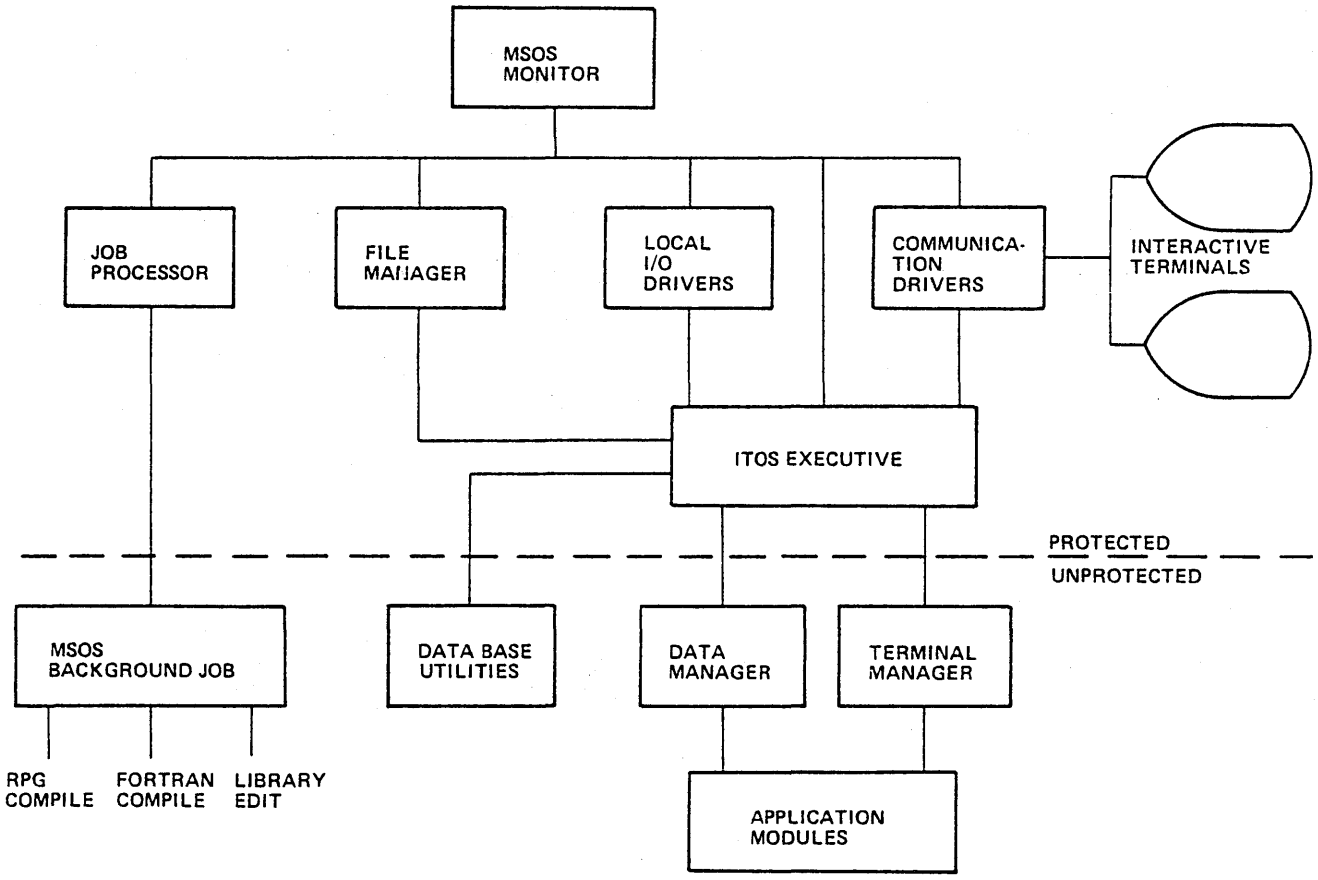


Figure 1-1. ITOS Software Configuration

TABLE 1-1. LIST OF APPLICATION MODULES AVAILABLE

Application Program	Mnemonic †
Order Entry/Invoicing/Sales Analysis † †	OE
Order Entry File Maintenance † †	OF
Accounts Receivable	AR
Inventory Control	IN
Accounts Payable	AP
Payroll	PR
General Ledger	GL
Routing	RT
Bill of Material Processing	BM
Material Requirements Planning	MR
Purchase Order Processing	PO
Physical Inventory	PI
Work in Process	WP

† Mnemonic used to activate the applications module. It is entered from the user terminal.

†† Order Entry/Invoicing/Sales Analysis has two discrete lists of tasks that can be performed. Each has its own mnemonic.

A primary characteristic of the application module technique is that the operator works interactively with programs using simple, easy-to-understand commands and data displays. Very little training is required to become proficient in the use of the system.

ITOS includes a file management system that provides for storage and manipulation of data. Files may be either sequential or indexed. If indexed, a file may be accessed by one to four keywords, allowing the user a variety of ways to retrieve stored data.

The user can develop his own business-oriented programs and enter them into the system as a new application module. The modules can be written in RPG II, FORTRAN or macro assembler language. Further, the user may use RPG II language to modify any of the existing application modules.

A built-in security system provides privacy for each of the files in which the user keeps his information. An optional security system requires that each user know the system password before he can use the system at all.

At least one terminal is required for the system; up to 17 terminals can be supported. One of the terminals is designated the master terminal. The master terminal initiates the system and also starts and stops ITOS.

In addition to executing the full range of tasks listed above, the operator can use the master terminal as the MSOS comment device, and perform tasks (such as job processor tasks) under an MSOS system running independently of ITOS. These capabilities are described in detail in the MSOS reference manual listed in the preface.

The ITOS executive acts as an interface to the MSOS request processors, including those processors that transfer data to and from the terminals. Requests may originate from a terminal or from a user program. ITOS prevents one user from altering another user's private data files, but allows all users to share common data files.

Several user programs may reside in executable memory at one time, and may execute in interleaved segments.

An ITOS system may contain work stations. These word stations consist of clusters of devices that are operated remotely through the communications controller and are coupled with a remote terminal. Each device occupies a communications port and is controlled only from the associated terminal unit.

FEATURES

The principal features provided by the ITOS executive are:

- Program execution throughout the first 256K bytes of main memory. ITOS uses a memory paging technique to map programs in physical main memory into an execution memory area. This feature speeds throughput and program swapping. It also provides compatibility with MSOS and its utilities.
- User program execution in unprotected memory. Protected memory is reserved for executive functions: MSOS, the ITOS executive, and certain file management functions.
- Nonuser task execution in the background by means of the MSOS job processor. In large memory configurations, ITOS may be configured to allow MSOS job processor programs and user programs to execute concurrently on a timeshared basis; otherwise, ITOS must be disabled to execute the MSOS job processor.
- The memory used for background execution is allocated from the same pool used by the ITOS executive for user programs. This allows a larger amount of execution memory for ITOS users when the background is not active.
- Complete support for programs written in RPG II, FORTRAN, or macro assembler language. The ITOS executive intercepts and checks each program request before translating the request into MSOS format for execution.
- Complete support for from 1 to 17 user terminals on a timeshared basis. The ITOS executive intercepts and checks all terminal requests before translating them to MSOS format for execution.
- Overlapped execution of user programs with terminal I/O operations, file management functions, and mass memory swapping to improve user program throughput
- RPG II programs are compiled in such a way that individual programs share one or more common runtime interpreters (known as multiuser programs) in a serially

re-entrant fashion. This reduces main memory space usage and improves program throughput.

- Data files are kept on disk packs, with each disk pack defined as a volume. Every volume is uniquely identified. The system may have up to eight volumes online (the maximum hardware configuration) and an unlimited number of volumes offline. The first volume contains MSOS, the ITOS executive, and all user programs. This volume must be online at all times, and is referred to as either the system volume (SYSVOL) or the MSOS library unit.

- User task processing is sequential and is managed to optimize throughput. To accomplish this, programs may be swapped and exchanged between the main and mass memories.

- Protect processing validates each request, preventing interference with any other user program that is ready to execute or that has started execution. Protection is provided in two ways:

  -The system operator may specify a system level password. If this option is chosen, a user may not log onto the system unless he knows the proper password. Passwords may be changed at any time from the master terminal.

  -User file protection is provided by the file names. The file manager treats the name as having three distinct parts: a file identification (name), an owner name, and a volume identification. If the user logs onto the system using no identification, he is restricted to using the common files that can be accessed (and altered) by any other user who logs onto the system with no user name. If the user logs onto the system with his unique user ID, this ID automatically becomes part of any new file that he creates. This user is then restricted to accessing files containing his user ID as a part of the file name. In addition, if the user chooses to restrict his files to a dedicated volume, he has the option of mounting that volume only when he is using his terminal, thus ensuring that no one else has access to his files.

  Since the user ID is verified against a list of permitted user IDs, it is possible to restrict any user to certain terminals. This terminal usage is specified in a system file and can be changed by the system operator.

- I/O completion processing prepares other programs for execution and can suspend execution of a running program if that program's timeslice has been exceeded. This prevents lengthy user programs from locking out other user programs. The suspended program is then queued for further processing.

- The operator may manually interrupt programs; he may abort program processing at any time.

ITOS UTILITY FEATURES

UTIL

The file manager utility (UTIL) is provided to allow the terminal user interactive file management. Three types of functions are provided.

- Aiding the user while he executes UTIL: listing names of all UTIL commands, listing parameters for a UTIL command, changing I/O devices, listing all the files stored on currently online volumes

- Bringing volumes online and taking them offline; renaming, deleting, or purging files; and compressing or clearing records in files

- Altering file contents on storage media: loading files from another medium (for instance, from magnetic tape where they have been previously saved) or dumping files to another medium.

These utility calls may also be embedded in user programs, procedure streams, or functions. It should be noted that files are managed (stored) only on disk packs (volumes).

EDITOR

A second ITOS utility, text editor (EDITOR), provides line-by-line (record) editing for certain types of files. These must be sequential or direct files with 80-character records, and they must contain ASCII text or be direct (blank-filled). A direct file is a sequential file that initially has all of its records filled with blanks. EDITOR is used only in interactive mode.

SORT

The sort utility (DSORT) rearranges records from one or several files into a specified order. Records can be sorted by one or more keys. It is possible to specify the order of the sortings, and therefore the primary sorting criterion, the secondary sorting criterion, and so forth. The sort utility is not used in interactive mode.

HARDWARE REQUIREMENTS

ITOS exists in a CYBER 18-10M or 18-20 hardware environment. The typical CYBER 18-20 equipment for the system is:

Device	Comments
CPU: CYBER 18-20	Required; this is the processor without main memory.
1882-32 (4)	Storage increments, each 64K bytes long. One is required; four are maximum.
Terminal: 1811-2	One is required; 17 can be supported. One terminal is designated the master terminal; all peripheral devices are located in the vicinity of the master terminal.
Mass memory (disk): 1833-1/1833-3 1867-20	Required; disk drive controller with up to eight SMD disk drives. One drive is the minimum required.
Input device:	One of the following can be used:
1828-1/1829-30 or 1829-60	Card reader
1832-4	Magnetic tape controller with a mix of up to four 1860-72 or -92 Magnetic Tape Transports

Device	Comments
Line printer: 1828-1/1827-30 1828-1/1827-60 or 1827-7	Required
Communications line adapter: 1843-1	One is required for one to nine terminals. Two are required for 10 to 17 terminals.
Punch device: 501-12 TAB Products Card Punch	Optional

SOFTWARE REQUIREMENTS

The software environment of the ITOS requires:

- MSOS, the operating system for the CYBER 18-10M and 18-20. It is partially main-memory-resident and partially mass-memory-resident. In this version, the ITOS executive adjoins the main-memory-resident MSOS. MSOS includes the background job processor. MSOS is described in detail in the MSOS reference manual listed in the preface.

- Standard MSOS utility routines

  -On-line debugging program that contains main and mass memory change and dump capability, memory search, allocation and release, and magnetic tape control commands

  -Breakpoint and recovery programs for job processor programs

  -System checkout program for inspecting the main memory image after failure

  -Library editing for both the system and program libraries. The former includes MSOS system-level programs and utilities, and the latter includes all ITOS user programs and job processor programs.

  -System initializing aids

  -Library preparation and library editing aids

  -Listing and sorting aids

  -I/O utilities

  All of these are described in the MSOS reference manual listed in the preface.

- File Manager Version 2.0. This module manages sequential and indexed sequential files in a volume- (disk pack-) oriented system. After records are stored, they may be accessed sequentially, by key words (indexes), or by relative record position in the file. Initial record storage may be accomplished sequentially, or sequentially according to a selected primary key value. The file manager provides general file support, and is specifically designed for efficient support of RPG II-type files. A complete description of the file manager is found in the file manager reference manual listed in the preface.

- Batch file drivers. These modules provide an efficient interface between the File Manager Version 2.0 and the MSOS batch processor. Both batch input and batch output interfaces are included.

In addition, ITOS supports the following:

NOTE

All assembler, FORTRAN, and RPG II programs that output to a device other than the terminal must use the actual logical unit of that device, so that there is no conflict with the batch output driver.

- RPG II, a symbolic programming language compiler. The programmer describes his program requirements on RPG II specification sheets, which serve as input to the compiler programs. The compiler also accepts additional data arranged in tables and arrays. Compiler output includes an executable object program (stored by the compiler on disk) and/or a listing of the specifications plus any error messages. The RPG II program may then be cataloged into the MSOS program library for subsequent ITOS terminal execution. RPG II is described in detail in the RPG II reference manual listed in the preface.

- MS FORTRAN Version 3A/B, a symbolic programming language compiler that is ANSI FORTRAN-compatible. Linking of programs in FORTRAN is automatic. The operator may select compiled output in any of the following formats:

  -Source program listing on a list device

  -Object code listing on a list device

  -Condensed object code listing on a list device

  -Run-anywhere object code listing

  -Relocatable binary object code on a binary output device

  MS FORTRAN is described in detail in the MS FORTRAN 3A/B Reference Manual listed in the preface.

- CYBER 18 Macro Assembler, a three-pass assembler that converts source language input (including macro instructions) to relocatable output. It also generates a listed output. Source programs are written with symbolic machine, pseudo, and macro instructions. Macros may be defined by the user within the source program, or they may be placed on a separate macro library. Input is from the standard input device, binary output is to the standard output device, and list output is to the standard list device. A complete description of the macro assembler is found in the macro assembler reference manual listed in the preface.

- COMM 18 Version 2, a software remote batch communications subsystem that allows information transfer between an ITOS system and remote host processors. It provides a remote batch facility to Control Data 6000 or CYBER 70/170 host processors via the 200 user terminal (200UT) transmission mode (mode 4A). It can also communicate with an IBM Model 360/370 host processor by utilizing the HASP multileving protocol.

The software environment includes several optional software modules called the manufacturing and distribution system. These 12 optional user application modules are listed in table 1-1. Any or all of them may be ordered to accommodate the user's data processing requirements.

This section, together with section 3, describes ITOS as it is viewed from a user terminal. One of the user terminals is designated as the master terminal, and is located adjacent to the CYBER 18 cabinet. It is the only terminal that can be used to install the system, to start or stop ITOS, to run batch programs and MSOS utilities, or to execute user application programs or ITOS utilities that require use of a system peripheral device such as a line printer, card reader, or magnetic tape. Any terminal including the master terminal can be used to execute application programs or to run an ITOS utility.

Before executing an application program or ITOS utility, the user must log the terminal onto the system. After the desired tasks are completed, the user customarily logs the terminal off the system.

To run a user application program, the operator enters the mnemonic for that program as described under User Terminal Procedures below. Note that the CDC-supported application modules require considerable operator interaction from the terminal. In most cases the application module is designed to display messages that instruct the operator exactly how to find data, to alter data, or to receive reports. If the operator wishes more information than is supplied by the messages displayed at his terminal, he should consult the appropriate application reference manual listed in the preface.

Data is entered in the same manner from all terminals, and must always follow the display of the input prompting character, >. The output of this character indicates that the system is expecting some type of data entry whose form depends on the context of the request. The blinking underline (cursor) always indicates the position of data entry on the screen. Within limits, erroneous data entries can be corrected by using the backspace (←) key to reposition the cursor and then entering the data correctly.

Termination of data entry depends on the type of system request. In general, there are five types of input termination within MSOS, ITOS, and the standard applications programs as indicated in table 2-1.

NOTE

To operate properly, the terminal must be in page mode. This is accomplished by pressing the PAGE MODE switch on the terminal.

TABLE 2-1. INPUT TERMINATIONS

Key/Other	Effect		
	MSOS	ITOS	Standard Applications Programs
CARRIAGE RETURN	Normal terminator	Normal terminator	Ends the current field. If in the first position in the field, the field is filled with blanks (for letters) or zeros (for numerals).
ENTER+	Normal terminator	Normal terminator	Ends the current field. If in the first position in the field, the field is filled with blanks (for letters) or zeros (for numerals). This is the ENTER+ key on the arithmetic part of the keyboard.
ENTER-	Normal terminator	Normal terminator	Ends the current field. The entered value is made negative. If in the first position in the field, the field is filled with blanks (for letters) or zeros (for numerals). This is the ENTER- key on the arithmetic part of the keyboard.
RUBOUT	Data entry is ignored; request is repeated	Normal terminator	1. If the cursor is beyond the first character of the field, it is repositioned to the beginning of the field. 2. If the cursor is at the first character of the field, it is repositioned to the beginning of the previous field. 3. If the cursor is in the first field on a screen, it is repositioned at the first field on the control screen.
LINE FEED	Normal terminator	Normal terminator	Duplicates the current field.
RESET	Normal terminator	Tab function	Not used

MASTER TERMINAL PROCEDURES

INSTALLATION

The ITOS system allows two types of installation media: magnetic tape and disk pack. If magnetic tape installation is ordered, the MSOS-ITOS system is delivered on tape and must be installed using the MSOS magnetic tape procedures described in detail in the MSOS installation handbook. If disk pack installation is ordered, the entire system is installed on the install disk pack and verified by Control Data prior to shipment. To initiate the system, the operator places the pack on disk unit 0 and autoloads the system.

All MSOS and ITOS programs are installed in the first 32,767 sectors of disk unit 0. This is referred to as the system library unit in MSOS and as the system volume in ITOS. Disk autoloads always occur from unit 0. Note that this disk pack is labeled, as are all other disk packs.

AUTOLOADING THE SYSTEM

Before autoloading the system, the operator should be sure that an MSOS-ITOS system pack is ready on disk unit 0; that is, the disk pack is on the drive and the READY light is illuminated.

The CYBER 18 computer cabinet contains a control panel from which system control functions are performed. The PROCESSOR section of this panel contains the switches necessary to autoload the system.

The system is autoloaded by pressing successively on the control panel:

```
STOP
MASTER CLEAR
AUTOLOAD
RUN
```

All remaining entries are made from the master terminal keyboard.

The master terminal displays:

```
MSOS 5.0 — PSR LEVEL xxx mm/dd/yy
SET PROGRAM PROTECT (ESC J28 @ )
```

The operator presses ESCAPE and types:

```
J28@
```

The master terminal displays:

```
'system name'
ENTER DATE/TIME MMDDYYHHMM
>
```

The operator enters the correct time in the requested format (for example, 0815770900 for August 15, 1977, at 9:00 a.m.).

The master terminal displays:

```
DATE: 15 AUG 77  TIME:  0900:00
```

At this point the system is ready to accept operator instructions from the master terminal.

MASTER TERMINAL OPERATIONS

The following operations are possible from the master terminal:

- All supported MSOS operations including running job processor programs (note that for large memory versions of ITOS, these job processor tasks may also be run after ITOS is enabled)
- Enabling ITOS

After ITOS has been started (enabled), other system operations are permitted from the master terminal only:

- Disabling ITOS
- Purging system files
- Changing the system password

All of these operations must be initiated by manually interrupting the system.

The other operations that are only possible from the master terminal include:

- Certain file manager utility operations such as purging files, dumping and reloading files, and all volume-related tasks. These are described in section 3, Utilities.
- Altering system files. This is described in section 5, System Files.

Operations possible at all terminals (that is, logging on, logging off, and interrupting or aborting ITOS programs) are discussed later in this section under User Terminal Procedures.

Manual Interrupt

All MSOS and ITOS system requests must be initiated from the master terminal by use of a manual interrupt. This is performed either by pressing the MANUAL INTERRUPT switch on the computer control panel or by simultaneously pressing the CONTROL and G keys on the master terminal. The system responds by clearing the master terminal screen and displaying:

```
MI
>
```

At this point the operator may request an MSOS system function by following the procedures described in the MSOS reference manual, or he may perform an ITOS system request.

Enabling ITOS

ITOS is not automatically enabled when the MSOS system is autoloaded. It must be started by the system operator.

Several initialization functions may be performed as a result of a manual interrupt, including building system files. If the system contains no files when the request is made to enable ITOS, all necessary system files are defined and initialized from data contained in the MSOS program library. This

process requires several minutes, but should not be necessary once ITOS has been executed and file backup procedures have been performed. Section 5 contains a description of the ITOS system files.

Every time the ITOS system is enabled, a program directory file ($$PGMNAM) may be built from information contained in the MSOS program library. This process requires approximately 30 seconds and ensures that any user programs recently loaded into the system may be rapidly and correctly found by the ITOS executive. The ITOS system may be started without building the program name directory file. In this case, the ITOS executive takes slightly longer to locate user programs that are not contained in the directory file.

CAUTION

A user program loaded in the MSOS program library that replaces an existing program (that is, the programs have the same name) may not be correctly located by the ITOS executive unless the start operation is performed. If the start operation is abnormally terminated an INIT must be performed prior to attempting START again.

The following procedure is used to start ITOS:

Display/Keyboard	Comments
CONTROL G	The operator performs a manual interrupt by pressing CONTROL and G simultaneously.
MI >	The system indicates that a manual interrupt is active.
START (cr) or START,N (cr)	The operator enters a start request. The optional parameter (N) specifies that the program directory file will not be rebuilt.
BUILDING SYSTEM FILES	The message indicates that the system files are being built.
ITOS ACTIVE AT hhmm	The system confirms that ITOS is active.

At this point, any terminal can log out ITOS. A sample initialization/startup procedure is shown in table 2-2.

Password Entry

The master terminal operator may add, delete, or change the system password. The password consists of from one to eight alphanumeric or special characters. If the password option is active, the current password must be entered by the terminal user during log-on. Passwords are deleted by entering a single blank character as the new password value.

The following procedure is used for password entry.

Display/Keyboard	Comments
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
PASSWD (cr)	The operator enters the password request.
PASSWORD = >	The message requests the new password.
newpass (cr) or (cr)	The operator enters a new password. The entry of just a carriage return aborts the request without changing the password.
PASSWORD ENTERED	The system confirms that the password was entered.
REQUEST ABORTED	This message appears if a carriage return only was entered.

Purging System Files

The system operator may purge all files on the system volume. This is an extremely severe operation that is required only if the system volume files contain errors and no backup is available. (Refer to Save in section 3.) Following this request, the space allocation directory is intialized, removing all files from the system volume. A subsequent start request then builds the necessary system files to an initial condition. Purging system files should be done only when no other recourse is available. ITOS must be disabled during system file purging.

The following procedure is used to purge system files:

Display/Keyboard	Comments
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
INIT (cr)	The operator enters the system file purge request.
WARNING: ALL SYSTEM VOLUME FILES WILL BE PURGED	The message indicates the result of the request and requests operator verification.
VERIFY >	
OK (cr)	The operator confirms the request. Any other entry aborts the request.
REQUEST COMPLETE	The system indicates that the files have been purged.

NOTE

A start operation should always be performed immediately following a purge operation. If the system is autoloaded before a start is performed, file errors will occur during the next start request. The INIT operation must be repeated to correct this situation.

TABLE 2-2. SAMPLE ITOS PROCEDURES

Display/Keyboard	Comments
INITIALIZATION	
STOP MASTER CLEAR AUTOLOAD RUN	With SYSVOL mounted and ready, the operator presses the CYBER 18 panel switches to autoload the system.
MSOS 5.0 – PSR LEVEL 118 05/04/77 SET PROGRAM PROTECT (ESC J28@)	The terminal display indicates the current MSOS support level.
ESCAPE J28	The operator protects the system by pressing the ESCAPE key followed by typing in J28 .
'system name' ENTER DATE/TIME MMDDYYHHMM >	The terminal displays the system name and requests the operator to enter the system time base.
0504770900	The operator enters the date and time.
DATE: 4 MAY 77 TIME 0900:00	The system displays the date and time.
STARTUP	
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
START (cr)	The operator enters the start request.
BUILDING SYSTEM FILES	The message indicates that system files are being built.
ITOS ACTIVE AT 0902	The system confirms that ITOS is active. At this point, any terminal can log onto the system.
DISABLING ITOS	
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
STOP (cr)	The operator enters the stop request.
VERIFY	The message requests operator verification.
OK (cr)	The operator confirms that the system is to be stopped. Any other entry aborts the request.
UNTIL HHMM >	The message requests the expected restart time.
1145	The operator replies with the expected return to operation.
ITOS OFFLINE AT 0905	The system confirms that ITOS has been disabled. A message to this effect is displayed at every active terminal, together with the expected restart time.

Disabling ITOS

Whenever it is necessary to disable ITOS for maintenance or other reasons, the following procedure is used:

Display/Keyboard	Comments
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
STOP (cr)	The operator enters the stop request.
VERIFY >	The message requests the operator to verify that the system is to be stopped.
OK (cr)	The operator confirms that the system is to be stopped. Any other entry aborts the request.
UNTIL HHMM	The message requests the expected restart time.
hhmm (cr) or (cr)	The operator replies with a value between 0000 and 2400. If the expected restart time is not known, the operator replies with a carriage return only.
ITOS OFFLINE AT hhmm	The system confirms that ITOS has been disabled. A message to this effect is displayed at every active terminal, together with the expected restart time, if any.

If a user attempts to log onto the system while it is disabled, the following message appears at the terminal:

OFF UNTIL hhmm

If the system was stopped without an expected restart time, the message is:

OFF UNTIL ????

The operator at the master terminal may repeat the stop procedure at any time so that the expected restart time can be updated.

A sample disabling operation is shown in table 2-2.

Changing Card Reader Format

The system operator may specify the format of the cards being read by the system card reader. Two formats are allowed: 026 (ASCII-63) or 029 (ASCII-68).

The following procedure is used to specify the card reader format:

Display/Keyboard	Comments
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active
CARD26 (cr) or CARD29 (cr)	CARD26 for 026 format, or CARD29 for 029 format.

Local Batch Processing Control

Local batch processing is controlled from the master terminal. It is initiated by means of the MSOS *BATCH command. There are three forms of this command:

Command	Comments
*BATCH	Causes the batch processor to begin receiving input from the MSOS standard input device (logical unit 10)
*BATCH,lu	Causes the batch processor to begin receiving input from the specified logical unit. (If the operator wishes to control the job from the master terminal, *BATCH,4 should be used.)
*BATCH,F	Causes the batch processor to begin processing deferred batch jobs from the batch queue file ($$BATCH)

Once this command is invoked, the batch processor executes until all jobs are received from the specified device, or until the batch queue file is empty.

Once initiated, deferred batch processing (*BATCH,F) continues automatically, even after the batch queue file is empty. This allows remote users to continually submit deferred jobs for processing without requiring computer operator intervention. Automatic deferred batch processing may be disabled in two ways:

- Local batch may be disabled by means of the SET,LOCL,0 command under the ITOS utilities (refer to the section on batch utilities). This command terminates local batch processing at the end of the current job and automatic batch until the next *BATCH,F is entered.
- If the automatic batch feature is not desired the indicator AUTON located in SYSDAT must be set negative at system installation time. This permanently disables the feature.

It is always possible to abnormally terminate batch processing by means of the *Z command. However, this aborts the current job and the batch processor slews to the next job in sequence (or to the next entry in the batch queue).

In some ITOS configurations it may not be possible to initiate certain system requests while batch processing is active. This is evidenced by a lack of response to the manual interrupt request. If this occurs, the operator may disable automatic deferred batch and/or terminate the currently-executing job as described above.

MSOS DEVICE ERRORS

MSOS peripheral devices are often used during the execution of ITOS programs from the master terminal. Occasionally device errors may occur (for example, the line printer is not ready). This message is displayed:

```
L, lu FAILED ec
ACTION
>
```

Where: lu is the MSOS device logical unit number.

ec is the device error code.

One of two legal responses can be made to this message:

- RP – Allows the MSOS I/O request to be repeated after the error condition has been resolved.
- CU – Allows the operator to abort the program that is performing the I/O request.

The standard logical unit assignments are given in table 2-3.

USER TERMINAL PROCEDURES

To execute terminal programs under ITOS, it is necessary to log on at a terminal. Logging-on may not be performed unless ITOS is enabled (that is, a start operation has been performed).

One of the initialization operations of START is to activate all terminals so that entered data is echoed on the screen. If data entered on the keyboard does not appear on the screen, either the system is not enabled or a terminal malfunction exists.

LOGGING-ON

Once ITOS is enabled, all inactive terminals may accept a request to log on, so the input prompter is not required. The log-on request consists of a single plus character (+) followed by a carriage return. If no response is forthcoming, the entry should be repeated, since data may have been inadvertently entered prior to the plus sign.

The logging-on operation is identical on all terminals and is initiated by entering:

```
+ (cr)
```

The terminal displays:

```
date time
CDC CYBER 18 ITOS SYSTEM - VER 1.2
system name
TERMINAL = nn
```

The system name is a parameter in the system data base (SYSDAT). The master terminal is terminal 00; other terminal numbers coincide with port numbers in the range 01 through 16.

TABLE 2-3. STANDARD LOGICAL UNIT ASSIGNMENTS

Logical Unit	Device or Function	Notes
01-03	Reserved for MSOS	
04	Master terminal	
05	Remote terminal controller	
06	Magnetic tape unit 0	1
07	Magnetic tape simulation unit 0	
08	Disk unit 0	
09	Line Printer	
10	Card reader	2
11	Card punch	3
12	FORTRAN line printer	
13	Batch input	
14	Batch output	
15	Disk unit 1	
16	Magnetic tape unit 1	
17	Magnetic tape unit 2	
18	Magnetic tape unit 3	
19	Disk unit 2	
20	Disk unit 3	4
21	Disk unit 4	
22	Disk unit 5	
23	Disk unit 6	
24	Disk unit 7	5

NOTES:

1. If the system contains no tape, a dummy is substituted.
2. Magnetic tape unit 0 or the master terminal may be substituted, depending on the configuration.
3. Magnetic tape unit 1, magnetic tape unit 0, or the magnetic tape simulation unit 0 may be substituted, depending on the configuration.
4. CYBER 18-10M may only contain disk units 0 through 3. CYBER 18-20 may contain disk units 0 through 7.
5. Work station logical units are always assigned as the last group of units in the system.

If the ITOS system has been disabled by the system operator, the terminal displays:

```
OFF UNTIL hhmm
```

which indicates the time the system is expected to return to service. If this time is not known, hhmm is replaced by ????.

If a system password is in use, the next display is

```
PASSWORD = >
```

to which the operator enters the current password. As a security measure, the password is not echoed at user terminals.

The terminal next displays:

```
USER ID = >
```

to which the operator enters the necessary identification. The user identification is not echoed at user terminals.

Common File Option

To access common files, the operator replies to USER ID = > with a carriage return. The system now uses a blank as the owner portion of the file name, making common files available to this operator. Any new files created by the operator are common files. The operator may not access any files that have a nonblank user ID. If the user wishes to access his private files, he must log off the system and then log on using his user ID.

User File Option

To use only his own private files, the user replies with his own ID. The message

USER ID = >

is answered with the user ID, consisting of one to eight alphanumeric characters; the blank, comma, semicolon, and equal sign are prohibited characters.

If this ID matches one of the set of IDs in the system $$USERID file for this terminal, the user can enter requests; otherwise, the user is logged off.

NOTE

Each terminal has a list of users who can legally perform operations at that terminal. Techniques for changing terminal access are discussed in section 5, System Files.

As in the case of a password error, entry of an invalid user ID is not immediately conveyed to the operator. Instead, the system proceeds as if a valid user ID were entered.

System File Option

The operator at the master terminal can use the system files to perform maintenance on them. To do this, the operator replies to

USER ID = >

with the entry

$$ (cr)

A directory to all the system files (with file names such as $$DAYFIL, $$USERID, and $$OEMENU) may be obtained at the master terminal by use of the UTIL command STATUS. These files are described in section 5, System Files.

After the user identification is entered, the system responds with:

REQUEST = >

At this point the user has logged on, and may select any of the ITOS or application modules for execution.

If an invalid password or user identification was supplied during the logging-on process, the system does not accept a request entry. Instead it displays the message:

ILLEGAL LOG-ON

and automatically logs the terminal off. If this occurs, the logging-on procedure should be repeated after verifying that the password and user identification are correct.

Certain ITOS utility functions (such as purge and save) cannot be performed unless ITOS is inactive. To provide this capability, the operator can log on from the master terminal even though the system has been disabled by the stop operation. These utility functions are described in detail in section 3. ITOS should be restarted when the utility task is completed to allow normal user terminal operation.

In response to REQUEST = >, the user can request one of four types of operations:

- Program request – Every program that can be executed under ITOS is contained in the MSOS program library as a program library file. Individual programs may be executed by entering the name of the desired program in response to REQUEST = >. These names are one to six characters in length and may include ITOS utilities such as UTIL, EDITOR, and SWITCH.
- Device control request – Commands are provided to allow input/output data normally associated with the user terminal to be directed to other system devices or files. These commands are of the form:

  INPUT = name

  and

  OUTPUT = name

  where name represents a system device or file as defined in table 2-4. Note that many ITOS systems may not contain all of the devices listed in the table.

TABLE 2-4. DEVICE NAMES

Device/File Name	Description
'filename†	Any sequential file that has been properly defined
TERMINAL†	User terminal
LPRINTER	System print device
PRINTER	System print device - FORTRAN mode
READER	System card reader
TAPE0	Magnetic tape unit 0
TAPE1	Magnetic tape unit 1
TAPE2	Magnetic tape unit 2
TAPE3	Magnetic tape unit 3

† May be specified from a user terminal; all other devices must be specified only from the master terminal.

When the input command is used, data must be arranged on the input device or file exactly as it would be received from the user terminal, with blank records representing null entries (that is, carriage return only). An end-of-file condition terminates the input or output command and returns control to the normal interactive mode of the terminal.

Files used with the output request must be defined as sequential with non-sector-aligned records. The maximum record length is 136 bytes.

- Procedure stream request – The execution of a procedure stream may be initiated from the user terminal. A procedure stream consists of a sequence of program names, device control requests, and program parameters that are contained in a file or system device. The procedure stream name may consist of from one to eight characters and is only requested individually as a part of an error recovery operation.

  ITOS contains two device procedure names: CARDPRO, which allows procedure streams to be entered from the system card reader, and TAPEPRO, which allows procedures to be entered from magnetic tape unit 0.

- Function request – The normal mode of system operation uses the function request. Functions comprise the set of standard application programs available under ITOS. These function names are contained in the system menu.

  The system menu is displayed when the operator responds to:

  REQUEST = >

  with

  ? (cr)

  The system displays a menu similar to the one shown below.

```
OE - ORDER ENTRY/INVOICING/SALES ANALYSIS
OF - ORDER ENTRY FILE MAINTENANCE
AR - ACCOUNTS RECEIVABLE
IN - INVENTORY CONTROL
AP - ACCOUNTS PAYABLE
PR - PAYROLL
GL - GENERAL LEDGER
BM - BILL OF MATERIALS
MR - MATERIAL REQUIREMENTS PLANNING
PO - PURCHASING
RT - ROUTING
WP - WORK IN PROCESS
PI - PHYSICAL INVENTORY
UT - SYSTEM UTILITIES
ED - SOURCE PROGRAM EDITOR
EX - EXIT
REQUEST = >
```

  The operator enters the desired two-character mnemonic, followed by a carriage return. Once the menu mnemonic selection is made, the requested function is initiated. If the function is one of the standard application modules, a function menu is displayed on the screen. A typical example is shown below.

```
ORDER ENTRY FILE MAINTENANCE

A - FILE MAINTENANCE: TERMS
B - FILE MAINTENANCE: AREA
C - FILE MAINTENANCE: SALESMAN
D - FILE MAINTENANCE: INVENTORY
E - FILE MAINTENANCE: CUSTOMER
F - FILE MAINTENANCE: SALES ANALYSIS
G - FILE MAINTENANCE: CONTROL
H - FILE DISPLAY: TERMS
I - FILE DISPLAY: AREA.
J - FILE DISPLAY: SALESMAN
K - FILE DISPLAY: INVENTORY
L - FILE DISPLAY: CUSTOMER
M - FILE DISPLAY: SALES ANALYSIS
N - FILE PURGE: TERMS
O - FILE PURGE: AREA
P - FILE PURGE: SALESMAN
Q - FILE PURGE: INVENTORY
R - FILE PURGE: CUSTOMER
S - FILE PURGE: SALES ANALYSIS
Z - EXIT
    SELECTION = [>]
```

  The operator selects the desired item by entering the corresponding single-letter mnemonic. Each of the function menu items represents a procedure stream contained in a procedure file. A message is displayed indicating the beginning and the conclusion of every program as it is executed. When the procedure stream execution is finished, the same function menu is again displayed, and the operator makes another selection. If the letter Z is entered, the system returns to REQUEST = >, which allows another system function to be executed.

Whenever the operator enters a program, procedure, or function mnemonic, the ITOS executive searches the following directories in the order indicated:

1. Program name file
2. Procedure directory file
3. MSOS program library directory

If the request cannot be located in any of these directories, a message to this effect is displayed, and the system returns to REQUEST = >.

LOGGING OFF

Whenever the

REQUEST = >

message is displayed, the operator may log off the system by entering

EXIT or EX (cr)

The system replies with

ITOS LOG OFF hh:mm:ss

where time is given in hours, minutes, and seconds according to the system clock.

At this time, a record is added to the day file, which identifies the terminal number, user, and terminal active time (log on to log off).

A sample terminal use sequence is shown in table 2-5.

PROGRAM INTERRUPT

The ITOS utilities and text editor both allow the terminal operator to manually interrupt certain of their operations. Two methods are used to do this, depending on the type of terminal activity.

If messages are being displayed on the screen, this output must be terminated by pressing the CONTROL and G keys simultaneously. When this is accepted, the input prompter (>) is displayed. It may be necessary to repeat the CONTROL G operation several times in order to obtain the prompter. Once the prompter is displayed, the operator must enter CONTROL D (pressing the CONTROL and D keys simultaneously) to interrupt the program. If output is not in process, or if an input request has been made from the program, entering CONTROL D is all that is necessary. The individual actions taken by the ITOS utilities and text editor after the interrupt has been processed are described in section 3.

PROGRAM ABORT

It is sometimes necessary to terminate program execution prior to normal completion. The terminal operator may do this at any time by entering CONTROL A (pressing the CONTROL and A keys simultaneously). As with program interrupt, screen messages must be terminated with CONTROL G before the abort is entered.

Once the abort is accepted, the message

PROGRAM ABORTED

is displayed, and the system returns to REQUEST = >.

TABLE 2-5. SAMPLE TERMINAL USE SEQUENCE

Display/Keyboard	Comments
+ (cr) AUG 15 77 15:15:35 CDC CYBER 18 ITOS SYSTEM-VER 1.2 ITOS 1.2 SYSTEM CHECKOUT TERMINAL = 10	The user logs onto the system. The system replies with the current time, the system name, and the terminal ID, then asks for the password if the password option is in the system.
PASSWORD = >	The password is entered at the keyboard but is not echoed on the display except at the master terminal.
USER ID = >	The system requests the user identification. The user enters his identification, but it is not echoed on the display except at the master terminal.
REQUEST = > ? (cr)	The operator requests the system menu.
OE - ORDER ENTRY/INVOICING/SALES ANALYSIS OF - ORDER ENTRY FILE MAINTENANCE AR - ACCOUNTS RECEIVABLE IN - INVENTORY CONTROL AP - ACCOUNTS PAYABLE PR - PAYROLL GL - GENERAL LEDGER BM - BILL OF MATERIALS MR - MATERIAL REQUIREMENTS PLANNING PO - PURCHASING RT - ROUTING WP - WORK IN PROCESS PI - PHYSICAL INVENTORY UT - SYSTEM UTILITIES ED - SOURCE PROGRAM EDITOR EX - EXIT REQUEST = [>]	The system menu is displayed together with a request prompting statement.
AR (cr)	The operator replies by selecting the accounts receivable module.
ACCOUNTS RECEIVABLE A - CUSTOMER FILE MAINTENANCE B - CUSTOMER FILE LISTING C - CUSTOMER FILE PURGE	The system displays the list of procedure streams available in the accounts receivable module.

TABLE 2-5. SAMPLE TERMINAL USE SEQUENCE (Contd)

Display/Keyboard	Comments
D - TRANSACTION ENTRY E - TRANSACTION EDIT LISTING F - TRANSACTION CORRECTIONS G - CASH RECEIPTS/INVOICE REGISTER H - DETAILED AGING I - SUMMARY AGING J - PRINT STATEMENTS K - MONTH-END ACTIVITY L - YEAR-END ACTIVITY Z - EXIT SELECTION = [>]	
B	The operator selects the customer file listing procedure stream.
*BEGIN CUSTOMER FILE LISTING . . .	Comments are displayed indicating the beginning and end of each program in the procedure. Application programs customarily use the interactive mode to receive and display data.
*CUSTOMER FILE LISTING COMPLETE	
ACCOUNTS RECEIVABLE A - CUSTOMER FILE MAINTENANCE B - CUSTOMER FILE LISTING C - CUSTOMER FILE PURGE D - TRANSACTION ENTRY E - TRANSACTION EDIT LISTING F - TRANSACTION CORRECTIONS G - CASH RECEIPTS/INVOICE REGISTER H - DETAILED AGING I - SUMMARY AGING J - PRINT STATEMENTS K - MONTH-END ACTIVITY L - YEAR-END ACTIVITY Z - EXIT SELECTION = [>]	Accounts receivable procedure stream menu is again displayed.
Z	Exit is requested.
REQUEST = >	
EX (cr)	The operator requests log-off.
ITOS LOG OFF 15:20:08	The terminal logs off.

This section, together with section 2, System Procedures, describes ITOS as it is viewed from a user terminal.

As explained in section 2, the operator has the choice of executing either an application program or a utility each time the terminal indicates that it is ready for another operation by displaying REQUEST = >. The operator replies with a system mnemonic to execute a user application program or with the name of one of the utilities described in this section.

It is also possible to call a utility from a procedure stream. In this case, the name of the utility is included in the procedure stream at the point where that utility's operation is required.

ITOS provides three utilities:

- UTIL is a file oriented utility that controls equipment to be used, lists files, and sets up, changes, compresses, or releases files. UTIL also brings disk packs online and loads, copies, or dumps files.
- EDITOR is a line-by-line text editor for files that have 80-character text records. It allows entry and modification of data in existing files. It also lists files, reformats file data, and locates data in files.
- Sort/merge is a file-oriented utility for sorting and merging file data by one or more key values.

Sort/merge is specifically oriented toward operation in procedure streams, and is not available for interactive use. EDITOR is specifically oriented toward interactive use, and is not available for procedure streams.

In addition to the ITOS utilities, there are several utilities that run under MSOS. These utilities are available only at the master terminal by using an MSOS manual interrupt or by calling the utility through the job processor in batch mode. A summary of the MSOS utility capabilities not available under ITOS is contained in appendix D.

UTIL

The file manager utilities operate under the direction of the UTIL executive, which is called through the ITOS executive. UTIL reads the individual request processor into main memory to process each UTIL command. UTIL operates in one of two modes: interactive or procedure stream.

INTERACTIVE MODE

After UTIL is activated by replying to REQUEST = > with

UTIL or UT (cr)

the utility indicates it is ready to execute utility commands by displaying

```
UTIL IN
READY>
```

on the terminal screen. When the operator enters the name of a specific utility command, UTIL individually requests each parameter that is necessary for the command. Two levels of prompting, normal and full, are available to help the operator enter these commands.

- Normal prompting – When the operator specifies a command, UTIL lists the mnemonic for each parameter associated with that command on a separate line. The operator must know the name and the form of each parameter. In this mode, the operator has the option of requesting the full name of a parameter.
- Full prompting – When the operator specifies a command, UTIL lists the full name of each parameter on a separate line.

The prompting level may be specified at any time by use of one of the help commands.

Any UTIL operation can be terminated by a program interrupt (CONTROL D) as described in section 2. This results in the utility requesting another command by displaying

```
READY>
```

at the user terminal.

PROCEDURE STREAM MODE

Procedure stream mode does not use prompting. Rather, parameters are specified in free-field form on the record that contains the mnemonic. This mode requires that the device control command (INPUT =) be previously executed in order to specify the procedure stream as the input for UTIL.

In procedure stream mode the file manager utility is requested by a record containing UTIL as a part of the procedure stream.

UTIL contains 22 commands that provide four types of operations:

- Operations concerned with the UTIL executive – Specifying the input and output devices, listing the UTIL command set, determining the amount of command prompting the utility displays to the operator in interactive mode, and exiting from the utility.
- Operations on a file level – Defining and deleting files, releasing file space, listing file directory information or the file itself, renaming the file, copying a file to another file or to magnetic tape, compressing files to eliminate records previously marked to be deleted from the files, purging expired files from the system, and reloading files that were previously saved on magnetic tape.
- Operations on a record level – Allowing file records to be loaded from a user terminal or from an external device.

- Operations on a volume level – Providing a means for initializing a volume (disk pack), making a volume accessible (online) or inaccessible (offline), and copying one volume to another.

A summary of these operations is provided in table 3-1, with the minimum request mnemonic underlined. Note that several of the commands can be initiated only from the master terminal. This restriction is necessary because these commands involve peripheral devices (magnetic tape, card reader, or printer) or involve a physical action on a disk drive. All of this equipment is located at the master terminal and may require the attention of the operator at that terminal. Note also that certain commands require that ITOS be disabled prior to their use. These commands must be performed on a static file system.

NOTE

Prompting is not used in procedure stream mode, for commands a e contained in records within the stream. These records have the form:

```
command,p1=xxxxxxxx,p2=xxxxxxxx,
. . . pn=xxxxxxxx
```

where p1 through pn are two-character parameter mnemonics. Each parameter is delimited by a comma.

TABLE 3-1. SUMMARY OF UTIL COMMANDS

Command Mnemonic†	Parameters††	Function	Restrictions
		EXECUTIVE OPERATIONS	
HELP	x	Determines the amount of prompting that is given to the operator each time a new command mnemonic is entered.	None
COMMAND	x	Lists every command available in UTIL.	None
INPUT=	name	Changes the device/file used to enter input commands and parameters.	System peripheral devices may be specified only from the master terminal.
OUTPUT=	name	Changes the device/file that receives the output from the command operation.	System peripheral devices may be specified only from the master terminal.
EXIT	–	Returns control to the ITOS executive.	None
		FILE MANIPULATIONS	
DEFINE	FN VL ED TY,LR,NR Kx,Px,SA	Creates a file on mass storage. Enters the name in file directory; places an expiration date on file.	Direct file records must be less than 512 bytes.
DELETE	FN VL	Deletes a file from mass storage; removes the name from file directory.	None
CLEAR	FN VL	Releases space held by a file; retains the name and entry in the file directory but clears all records from the file.	Direct files may not be cleared.
LIST	FN VL M L F	Lists the contents of a specified file on a specified device.	Only the master terminal may list on a system peripheral device.
STATUS	FN OW VL	Prints file status information; status can be obtained from all the files on a single volume, all files for one owner, or all files on all mounted and ready volumes.	Only the files available to the terminal user are printed.

† Minimum calling mnemonic is underlined

†† See table 3-2 for parameter definitions.

TABLE 3-1. SUMMARY OF UTIL COMMANDS (Contd)

Command Mnemonic†	Parameters††	Function	Restrictions
RENAME	FN VL F2 ED	Renames the file and/or changes the expiration date of the file.	None
COPY	FN VL F2 OW V2	Copies a file to a new file; name and/or owner and/or volume of the new file must be different.	Both files must have identical key definitions. Maximum record length is 8000 bytes, and both record lengths must be equal.
COMPRES	FN VL	Copies a file into the same location, rebuilding keys as necessary and deleting all records marked to be deleted; the file name remains unchanged.	None
PURGE	OW VL	File space is released and the entry is removed from the file directory for all expired files.	Available only from the master terminal. ITOS must be disabled.
DUMP	FN OW VL P	Dumps file(s) on the selected output tape; by file name, owner or volumes; used to files on magnetic tape. File remains defined until purged or deleted. Deleted records are omitted.	Available only from the master terminal
RELOAD	FN OW VL I	Reloads file(s) from the selected input device by file name, owner, or entire tape contents onto specified volume. Used to re-enter dumped files.	Available only from the master terminal.
		RECORD MANIPULATION	
LOAD	FN VL I M	Loads records into an existing file.	Maximum record length is 512 bytes. If the file contains records already, new records are appended at the end.
		VOLUME MANIPULATION	
INIT	VL NF DK	Initializes a volume (disk pack) for use by the file manager. Names the volume and writes the volume label. Allows renaming of a volume without destroying existing file contents.	Available only from the master terminal. The volume must be dismounted.
MOUNT	VL DK	Places a volume (that is physically ready on a disk drive) online with respect to ITOS.	Available only from the master terminal
DISMOUNT	DK	Marks as offline to ITOS a volume that is physically ready and currently mounted.	Available only from the master terminal
SAVE	DK D2	Copies the entire contents of one volume onto another.	Available only from the master terminal. ITOS must be disabled.
HOST	HO OP PT	Adds or deletes entries in the host file	Available only from the master terminal

† Minimum calling mnemonicsis underlined

†† See table 3-2 for parameter definitions.

TABLE 3-1. SUMMARY OF UTIL COMMANDS (Contd)

Command Mnemonic†	Parameters††	Function	Restrictions
SET	HO LU	Assigns logical unit numbers to entries in the host file	Available only from the master terminal
BATCH	FN OW VL HO TY PN M	Creates entries in the batch file for processing by the job processor or a specified host	None
BATS	JN HO L	Lists or displays status of an individual active job, status of all active jobs by host, or status of all active jobs for all hosts. Also lists or displays a tabular summary for a particular host or for all hosts	Summary available only from the master terminal. User terminal. may only get the status of those jobs associated with that user ID.
DISCARD	JN	Deletes entries in the host file	The user terminal may only discard jobs associated with that user ID.
FLUSH	HO DO	Purges all jobs in the host file that are in a SENT status and that are n days old. If n is a negative sign, all jobs are purged for that host.	Available only from the master terminal. To purge all jobs from a host, the host must be inactive.
DISPOS	JN OP NC SC V2 FN	Selects the print option for output files or moves an output file to another file. May also select move and print	The user terminal may only dispose of jobs associated with that user ID.
PRINT	OP L	Prints the output files from batch processing	Available only from the master terminal

† Minimum calling mnemonic is underlined

†† See table 3-2 for parameter definitions.

UTIL COMMAND FORMATTING REQUIREMENTS

The basic format for each UTIL request is a command followed by a parameter list (see table 3-2). If interactive mode is used, each parameter is entered on a separate line.

Only the first four letters of the command mnemonic need be used, except for INPUT and OUTPUT.

The logical file names consist of a combination of the file name (FN), file owner (OW), and volume (VL). In general, the file owner is automatically specified as the terminal user identification: blank for common files, USER ID for private files, or $$ for system files.

The file, owner, volume, and device names must adhere to the following rules.

- Each must be eight or fewer characters in length.
- They must not contain a comma (,), equal sign (=), or semicolon (;), since these are used as delimiters in the command set
- They may contain blanks, since they are ignored.

System peripheral devices are available only to the master terminal (terminal 00)

In procedure stream mode, UTIL commands must start at the first character position of the input record.

Parameter strings may occupy more than one record in procedure stream mode. This is accomplished by replacing a comma delimiter with a semicolon (;) and continuing the parameter string on the next record.

Trailing blanks are disregarded.

UTIL COMMANDS

Help

The help commands are used in interactive mode only. They determine the amount of prompting afforded to the operator at his terminal.

TABLE 3-2. PARAMETER DEFINITIONS

Mnemonic	Meaning	Range
D2	Output disk unit	1 to 7
DK	Disk unit to copy from	1 to 7; 0 is SYSVOL
DO	Days old	1 to 999; negative sign indicates all
ED	File expiration date	mmddyy; default is present date
F	Format of list	F = Formatted, with record number U = Unformatted, without record numbers
F2	New file name	1 to 8 characters; F2 ≠ FN
FN	File name	1 to 8 characters; FN ≠ F2; embedded blanks are ignored.
HO	Host name	1 to 4 characters; default is LOCL
I	Input device	1 to 8 characters (see table 2-4); default is user terminal
JN	Job number	J001 - J060 for Local Batch J101 - J160 for host 1, and so forth
Kx	Key size, in bytes (K1 - K4)	1 to 29; default for K1 is 1, for all others is 0.
L	List device name	1 to 8 characters (see table 2-4); default is user terminal.
LR	Record length, in bytes	1 through 65,534; default is 192 bytes (one sector)
LU	Logical unit number	13 to 99
M	Mode for LOAD	A = ASCII (default) E = EBCDIC OA = Index file - ordered index (ASCII) OE = Index file - ordered index (EBCDIC)
M	Mode for LIST	A = ASCII (default) H = Hexadecimal E = EBCDIC
M	Mode for BATCH	R = Produce relocatable binary load A = Produce absolute binary load (default)
name	Input or output device name	1 to 8 characters (see table 2-4)
NC	Record length (number of characters)	1 to 999 characters
NF	Maximum number of files to be written on volume	1 to 2,048; default is 256.
NR	Maximum number of records	1 to 16,777,215; default is 1024.
OP	Option for HOST	ADD = Add DEL = Delete
OP	Option for PRINT	Jmnn = Job number PRxx = Unidentified print file PR00 = All unidentified print files Host Name = All jobs for the host
OP	Option for DISPOS	MOVE = Move file PRINT = Print file MOVEPR = Move and print file
OW	Owner name	1 to 8 characters; default is common files.

TABLE 3-2. PARAMETER DEFINITIONS (Contd)

Mnemonic	Meaning	Range
P	Selected output unit	TAPE0 to TAPE3; default is TAPE0.
PN	MSOS program name	1 to 6 characters
PT	Protocol type	HASP or 200UT
Px	Position of key (P1 - P4)	
SA	Sector alignment	Y = Yes; N = No (default).
SC	Starting character position	1 to 999 characters
TY	File type for DEFINE	S = Sequential file (default) R = Indexed file, random O = Indexed ordered file D = Direct file; max. 512 bytes
TY	Program type for BATCH	R = RPG II (default) F = FORTRAN A = Macro assembler
V2	New volume name	1 to 8 characters; default is automatic search of all mounted and ready volumes.
VL	Volume label	1 to 8 characters; default is automatic search of all mounted and ready volumes.
x	Command expander	Any alphanumeric character HELP: Full prompting COMMAND: Parameter mnemonics are displayed with UTIL command mnemonics.

The prompting level determines the appearance of the display after the operator enters the command mnemonic.

The prompting level remains in effect until EXIT returns control to the ITOS executive or until the alternate form of the help command is entered.

Two prompting levels are possible; normal and full.

Normal prompting is the default condition if operating in interactive mode. Under normal prompting, each parameter is individually requested by a message of the form:

pn = >

The operator is assumed to be familiar with the meaning of the two-character parameter mnemonic (pn). The operator may make one of three possible entries:

- A carriage return representing a null entry that causes UTIL to supply the parameter default value.
- A parameter value followed by a carriage return.
- A question mark (?) followed by a carriage return. If the operator is not familiar with a particular parameter, entering the ? causes the full name of the parameter to be displayed:

  parameter name = >

  Following this, the parameter value or null entry may be made.

Normal prompting is activated by entering

HELP (cr)

in response to READY. Since this is the normal operating mode of UTIL, its entry is only required when disabling the full prompting mode.

Full prompting is activated by entering

HELP, x (cr)

in response to READY. The mode selection (x) may be any nonblank character. Full prompting causes the full name of each parameter to be displayed instead of the two-character mnemonic. Data entry is performed in the same manner as in normal prompting mode, except the question mark has no meaning.

Command

This command displays a list of all UTIL commands on the user terminal. Two forms are available:

- COMMAN displays the names of all UTIL commands:

  HELP
  INIT
  DEFINE
  .
  .
  .
  COMPRE

- COMMAN,x, where x is any nonblank character displays the names of all UTIL commands, together with the corresponding parameter's mnemonics:

```
HELP,    M
INIT,    VL, NF, DK
DEFINE,  FN, VL, ED, TY, LR, NR, K1, P1,
         K2, P2, K3, P3, K4, P4, SA
   .
   .
   .
COMPRE,FN, VL
```

Input

The input command designates the device or file from which commands and parameters will be received by UTIL. All commands and parameters must be arranged on the device or file as they would be in interactive mode. Null entries must be specified by blank records. The format of the command is:

```
INPUT = name
```

where name is a device mnemonic or file name as indicated in table 2-2. This command can designate peripheral devices only if it is entered at the master terminal.

Output

The output command designates the device or file to which the UTIL output will be directed. Files used with the output command must be defined as sequential with 80-character non-sector-aligned records.

The format of the command is:

```
OUTPUT = name
```

where name is a device mnemonic or file name as indicated in table 2-2. This command can designate peripheral devices only if it is entered from the master terminal.

Exit

The exit command returns control of the system to the ITOS executive. All nonstandard device and prompting selections (HELP, INPUT, OUTPUT) are reset by this command. The command format is:

```
EXIT
```

Define

The define command creates a file manager file. The command format is:

```
DEFINE,FN=filename,VL=vlmlabel,ED=mmddyy,
TY=t,LR=reclngth,NR=maxnorec,K1=n,P1=n,K2=n,
P2=n,K3=n,P3=n,K4=n,P4=n,SA=s
```

Where: filename is the one- to eight-character file name; embedded blanks are ignored.

vlmlabel is the one- to eight-character volume name; the default value is disk 0, which is the system volume (SYSVOL).

mmddyy is the expiration date (month, day, and year). The default value is the present date.

t is the file type

- S Sequential file; records must be presented in sequence. This is the default value.
- R Indexed file; records can be presented randomly with respect to the primary key.
- O Indexed file; records must be presented in ordered fashion with respect to the primary key.
- D Direct file; a sequential file in which all records are blank-filled by UTIL. Direct file record lengths may not exceed 512 bytes.

reclngth is the record length in bytes. The range is 1 through 65,534; the default value is 192 bytes (one sector).

maxnorec is the maximum number of records. The range is 1 to 16,777,215; the default value is 1024 records.

Kx is the length of key x in bytes. The range is 1 to 29. For K1 the default is 1; for other keys, the default is 0.

Px is the position of the first byte of key x relative to the start of the record (byte 1). The default value is 1

s indicates whether sector alignment is chosen; a sector is 192 bytes. Sector alignment allows improved file access and throughput, but may require excessive mass storage file space.

- Y Yes
- N No; this is the default value

NOTE

If TY=S (sequential files) or D (direct files), all the Kx and Px parameters are suppressed.

The following example illustrates the interactive use of UTIL to define an indexed file.

```
DEFINE
FN=ORDERS
VL=VOLUME2
ED=022177
TY=0
LR=>
NR=>
K1=4
P1=>
K2=29
P2=5
K3=10
P3=34
K4=8
P4=44
SA=N
```

This creates an indexed-ordered file named ORDERS on volume 2. Each sector-length record has four keys: key 1 is a number of up to four characters (1 to 9999) and is the primary key. Key 2 is the customer name, an alphanumeric key 29 characters long, beginning immediately after key 1. Key 3 is a 10-letter regional key, starting immediately after key 2. Key 4 is a eight-character alphanumeric for the date of the order (mm/dd/yy) and it begins immediately following key 3. The file is not sector-aligned.

Delete

The delete command releases the space on mass storage held by the specified file. The named file's entry is removed from the file directory.

The command format is:

DELETE,FN=filename,VL=vlmlabel

Where: filename is the one to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default value specifies an automatic search of all mounted and ready volumes.

Clear

The clear command removes all records in the specified file but retains the file's entry in the file directory.

The command format is:

CLEAR,FN=filename,VL=vlmlabel

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default value specifies an automatic search of all mounted and ready volumes.

Direct files may not be cleared, since this invalidates this file's format.

List

The list command displays the contents of the designated file on the specified device in the mode selected (ASCII, hexadecimal, or EBCDIC). System peripheral devices may be used for output if the request is made from the master terminal. If the display is made to the user terminal, the word PAUSE appears at the end of the last complete record for this screen of data, allowing the operator to view the file listing each time the screen is filled. The operator presses the carriage return to receive the next screen of the listing.

The command format is:

LIST,FN=filename,VL=vlmlabel,M=m,L=lstdvnam,F=U/F

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default value specifies automatic search of all ready and mounted volumes.

m is the mode (A=ASCII, H=hexadecimal, E=EBCDIC); default specifies the file format is ASCII.

lstdvnam is the one- to eight-character list device name as indicated in table 2-2. The default is the user terminal.

U/F U = Unformatted without headers or record numbers.

F = Formatted with headers and record numbers.

Status

The status command prints the status of the specified file or the status of all files belonging to the specified owner. STATUS also indicates the number of free sectors remaining on the specified volume(s). This command prints only those files belonging to the user logged in at a terminal. All files in the system may be printed from the master terminal.

The command format is:

STATUS,FN=filename,OW=ownrname,VL=vlmlabel

Where: filename is the one- to eight-character file name.

ownrname is the one- to eight-character owner name; the default value specifies the common files.

vlmlabel is the one- to eight-character volume name; the default value specifies all mounted and ready volumes.

If all parameters are omitted, STATUS prints the status of all files on all mounted and ready volumes that belong to the user. This provides the operator with a file directory plus information about each file listed. If only the VL parameter is specified, the status of all files from that volume is listed.

Appendix C gives a sample status printout.

Rename

The rename command allows the user to change the name of a specified file and/or to change the expiration date of the file. The file and records remain in place, but are accessible only through the new name.

The command format is:

```
RENAME,FN=filename,VL=vlmlabel,F2=filenam2,
ED=mmddyy
```

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default causes an automatic search of all mounted and ready volumes.

filenam2 is the new file name.

mmddyy is the expiration date; the default value is the system date.

If F2 is defaulted, then only the expiration date is changed.

Copy

The copy command causes UTIL to copy an existing file to a new file on mass storage. The original file remains unchanged. The file type and record lengths must be the same for both files. The new file must contain at least as many records as the old file and may have more. The maximum record length for the files is 8000 bytes. Both files must have identical key definitions.

The command format is:

```
COPY,FN=filename,VL=vlmlabel,F2=filenam2,
OW=ownrname,V2=vlmlabl2
```

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default causes an automatic search of all mounted and ready volumes.

filenam2 is the new file name.

ownrnam is the one- to eight-character new owner name; the default specifies a common file.

vlmlabl2 is the volume containing the new file; the default value causes an automatic search of all mounted and ready volumes.

Compress

The compress command copies an existing file onto its own file space, but deletes any records that are marked to be deleted. New key lists are built as necessary.

The command format is:

```
COMPRE,FN=filename,VL=vlmlabel
```

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default value causes an automatic search of all mounted and ready volumes.

Purge

The purge command releases space on mass storage for all the designated owner's files that have an expiration date earlier than today's date. The file entry is removed from the file directory for the volume. This command is available only at the master terminal, and only after ITOS has been disabled.

Before purging any file, UTIL displays the name of the expired file from the class the operator specifies by the VL and OW parameters. The operator must verify that each named file is to be purged before the system will delete the file.

The command format is:

```
PURGE,OW=ownrname,VL=vlmlabel
```

Where: ownrname is the one- to eight-character owner name; the default value is all common files.

vlmlabel is the one- to eight-character volume name; the default value causes a search of all mounted and ready volumes.

Each time an expired file is located by UTIL, the message:

```
filename,ownername PURGE=>
```

is displayed on the screen. If the operator does not wish to purge that file, NO is entered. Any other entry causes the file to be deleted. A sample purge operation is shown in table 3-3.

Dump

The dump command transfers a specified file to a magnetic tape, together with sufficient information to allow the file to be redefined at a later time. The maximum record length is 512 bytes. Several files may be placed on a tape by repeating this command. All of a user's files may be dumped by specifying only the owner name; all files on a volume may be dumped by specifying only the volume name. This command can be requested only from the master terminal. Dumping clears records marked as deleted.

The command format is:

```
DUMP,FN=filename,OW=owner,VL=vlmlabel,P=tapex
```

Where: filename is the file name.

owner is the owner name.

vlmlabel is the volume name; the default value causes a search of all mounted and ready volumes.

tapex is the selected output unit.

TABLE 3-3. SAMPLE PURGING OPERATION

Display/Keyboard	Comments
REQUEST = >	System requests another task at the master terminal.
EX (cr)	Operator logs off ITOS.
CONTROL G	Operator manually interrupts MSOS.
MI >	ITOS acknowledges it is in interrupt mode.
STOP (cr)	Operator commands ITOS to stop.
VERIFY	ITOS requires confirmation.
OK (cr)	Operator confirms that system is to stop.
UNTIL HHMM	ITOS requests an expected restart time.
1425 (cr)	Operator enters the expected restart time.
ITOS OFF-LINE AT 1410	ITOS confirms that it is inactive at the current time.
+ (cr)	Operator simulates logging on to ITOS.
OFF UNTIL 1425	ITOS informs operator that it is inactive.
REQUEST = >	ITOS (in disabled mode) requests the next task.
UTIL (cr)	The operator requests UTIL.
UTIL IN READY>	System confirms that UTIL is active and ready for a command.
PURGE (cr)	Operator requests purging.
OW = SMITH (cr) VL = SYSVOL (cr)	Operator enters the parameters for the purging operation: all of Smith's expired files on the system volume.
NAMEA,SMITH PURGE = OK (cr) NAMED,SMITH PURGE = OK (cr) ⋮ NAMEN,SMITH PURGE = NO (cr)	System identifies these expired files and requests confirmation that they are to be deleted from the system. The operator deletes all but the last one.
READY = EXIT (cr) REQUEST = EXIT	UTIL replies that it is ready for another purge command. The operator chooses to exit from UTIL. The CPU is now under MSOS control.
CONTROL G	The operator manually interrupts MSOS.
MI >	MSOS confirms it is in manual interrupt mode.
START (cr)	The operator restarts ITOS. He must then log on to the system.

DUMP writes two end-of-file (EOF) marks after each file as it is copied onto tape. If another file is copied sequentially onto the tape, the second of these two EOF marks is eliminated.

If an end-of-tape is encountered, the tape is backed up to the end of the previous file. Two file marks are written and the tape is unloaded. A message is output to the operator:

MOUNT NEXT TAPE – (CR) WHEN READY

After the operator mounts and readies the next tape, he presses carriage return. The dump continues on the next tape.

Reload

The reload command allows files that were previously dumped via DUMP to be reloaded into the system. In order to be reloaded, the file(s) must not be defined in the system. (Note that the file name is a combination of FN, OW, and VL.) The reload utility processor creates the required dump from information saved on the magnetic tape during the DUMP operation. Specifying a file name and owner causes ITOS to search for that file/owner combination; specifying only an owner causes ITOS to search for all of the owner's files. This command is available only from the master terminal.

The format of the command is:

RELOAD,FN=filename,OW=owner,VL=vlmlabel,
I=inputdev

Where: filename is the one- to eight-character name of the file on the magnetic tape that is to be reloaded. The entire tape is searched for this file name. If FN is blank, all files on the tape are loaded.

owner is the owner name.

vlmlabel is the one- to eight-character volume name where the file is to be loaded. The default is volume 0 (SYSVOL).

inputdev is the one- to eight-character name of the input device.

If a single file is to be loaded, it is terminated by two EOF marks. If several files are to be loaded from the same tape, each file except the last is terminated by a single EOF mark; the final file is terminated by two EOF marks. As each file is located and loaded from tape, the message:

filename ownername LOADED

is displayed on the screen. If the file is already defined in the system prior to the reload request, the message reads:

filename ownername NOT LOADED, ALREADY PRESENT

If a file dump requires more than one tape, the RELOAD operation must be repeated for each tape of the dump.

Load

The load command allows a previously defined file to be loaded with data from an external device. This device may be the user terminal or a system peripheral, but the latter is only allowed from the master terminal. The maximum record length permitted is 512 bytes. Records are appended at the end of any existing records in the file.

The command format is:

LOAD,FN=filename,VL=vlmlabel,I=inputdev,M=mm

Where: filename is the one- to eight-character file name.

vlmlabel is the one- to eight-character volume name; the default causes a search of all mounted and ready volumes.

inputdev is the one- to eight-character device name; the default value specifies the user terminal.

mm is a mnemonic for the mode; the default value specifies that the source data is in ASCII mode.

A = ASCII
E = EBCDIC
OA = Ordered-indexed file (ASCII)
OE = Ordered-indexed file (EBCDIC)

If the input device is a system peripheral, loading continues until an end-of-file is encountered on the device, or until the designated file is full. If data is being input from the user terminal, an input prompter (>) is displayed as each record is requested. To terminate, the operator responds to the prompter with a program interrupt (CONTROL D).

INIT

INIT writes a volume label on the specified disk pack; it is used in interactive mode only, and can be used only from the master terminal. The volume in question must be dismounted prior to executing this command.

NOTE

This command may not be used to label the system disk (SYSVOL). SYSVOL is labeled as a part of MSOS system initialization.

The command format is:

INIT,VL=vlmlabel,NF=flno,DK=u

Where: vlmlabel is the one- to eight-character volume name.

flno is the maximum number of files to be written on this volume. The range is 1 to 2048; the default is 256.

u is the mass storage physical unit number. The range is 1 to 7.

If a volume label already exists on the disk pack, its volume name is displayed at the terminal, and the message:

VOLUME = >

is output. If the operator enters a carriage return to this request, a new volume label is created, and the file space directory is initialized, making the entire volume available for file storage. Any other response to VOLUME = (for example, NO carriage return) causes the volume name to be changed, but does not otherwise alter the disk pack.

Mount

The mount command places the specified volume online to ITOS. The volume should be loaded and ready on the specified disk drive prior to this request The ready condition is indicated by the READY light on the disk drive. This command is allowed only from the master terminal.

The command format is:

MOUNT,VL=vlmlabel,DK=u

Where: vlmlabel is the one- to eight-character volume name.

u is the mass storage physical unit number. The range is 1 to 7.

Dismount

The dismount command takes the specified volume offline from ITOS. The ready condition is not affected by DISMOUNT. This command is accepted only at the master terminal.

The command format is:

DISMOUNT,DK=u

Where: u is the mass storage physical unit number. The range is 1 to 7.

NOTE

DISMOUNT cannot be used to take SYSVOL offline from ITOS.

Save

The save command copies an entire disk volume to a specified output volume. It is available only at the master terminal, and only when ITOS is disabled.

SAVE operates off-line (MSOS system inoperative) so that a volume can be copied onto SYSVOL (volume 0).

The command format is:

SAVE,DK=u,D2=d

Where: u is the mass storage physical unit number from which to copy. The range of 0 to 7.

d is the mass storage physical unit number of the output volume. The range is 0 to 7.

After accepting the parameters, the following message is displayed:

```
TURN OFF PROTEC SWITCH (ESC J20@ ) AND
TYPE CARRIAGE RETURN
>
```

Perform the indicated operations on the keyboard: press ESCAPE, followed by J20@ and a carriage return. The following message appears:

```
SET UP VOLUME(S) TO BE SAVED AND TYPE
CARRIAGE RETURN
```

At this point, the volume disk packs can be installed on the drives as necessary. When all are ready, enter carriage return.

At the conclusion of the save operation, the message

VOLUME SAVE COMPLETE

is displayed, and the system must be autoloaded to continue.

To perform a save operation the operator must stop ITOS and then log onto the disabled system using the same method shown in the purge example in table 3-3.

Host

The host command allows an operator to add or delete entries in the $$HOST file. This command may only be executed from the master terminal.

This command format is:

HOST,HO=host,OP=opt,PT=proto

Where: host is the one- to four-character user-defined name for the host; for example, LOCL for local batch.

opt is ADD or DEL

proto is the type of protocol being used with this host (HASP or 200UT)

Set

The set command allows an operator to assign logical unit numbers to entries in the host file. The host entry must be previously defined by the host command before a logical unit can be assigned. A logical unit of zero causes the input batch driver to terminate processing jobs for that host.

This command may only be executed from the master terminal.

The command format is:

SET,HO=host,LU=lu

Where: host is the one- to four-character name of the previously defined host.

lu is a one- to two-digit logical unit number associated with the batch input driver.

The local batch input driver is always 13 in an ITOS system.

Batch

The batch command allows a terminal user to create entries in the batch file for processing by the batch input driver through the job processor, if local batch, or through a specified host. The file to be batched must be a source language text file (RPG II, FORTRAN, or macro assembler) and must contain an *JOB record somewhere in the file if it is for a remote host.

This command may be executed from the master terminal or the user terminal; however, job processing is initiated from the master terminal only.

The command format is:

BATCH,FN=filename,OW=ownrname,VL=vlmlabel,
HO=host,TY=t,PN=prgnam,M=A/R

Where: filename is the one- to eight-character file name of the file containing the source program. This file must not be sector-aligned.

ownrname is a one- to eight-character owner name. This entry is valid only from the master terminal; the user ID entered during log-in is automatically included in all other ITOS user terminals.

vlmlabel is a one- to eight-character volume name; the default value is assumed to be the system volume (SYSVOL).

host is the one- to four-character host name that has previously been defined by the host command.

ty is the mnemonic for the source language type:

R RPG II; this is the default value.
F FORTRAN
A Macro assembler

prgnam is the one- to six-character program name to be used in the MSOS program library directory. This parameter is required only for FORTRAN and macro assembler programs.

A/R A = Produce absolute binary output (*N,file,,,B)

R = Produce relocatable binary output (*L,entry)

The last three parameters (TY,PN,M) are required only for local batch processing.

Once the batch file entry is created, the job number is displayed on the terminal. This number has the format:

Jmnn

Where: m is a one-digit number identifying the host.

nn is a two-digit job number for that host.

To compile the program it is necessary to activate the MSOS batch processor, which may require that ITOS be disabled. Once the system is conditioned to run the batch processor, the following procedure should be employed.

Display/Keyboard	Comments
CONTROL G	Manual interrupt is performed.
MI >	Manual interrupt is active.
*BATCH,F	The MSOS batch processor is activated and directed to receive input from the batch input file.

Batch Status

The batch status command lists or displays the status of an individual active job, the status of all active jobs by host for a particular owner, or the status of all active jobs for all hosts for a particular owner. In addition, a summary may be requested to display or print a tabular summary of all jobs for all hosts or for a particular host

The summary option may be executed from the master terminal only. A user terminal may only receive the status of the jobs associated with that terminal's user ID. The master terminal user may obtain a status for all active jobs for a particular host or for all hosts.

The command format is:

BATS,JN=Jmnn,HO=host,L=lstdevnam

Where: Jmnn is the four-digit job jumber. The default is all jobs for a particular host or all jobs for all hosts. If the job number is SUMM, a summary of all jobs for a particular host or for all hosts is given (master terminal only).

host is a four-character host name. If the host name is not specified, a status (or summary) for all hosts is given.

lstdevnam is the output device name. The default output device is the terminal screen. The printer option may only be used from the master terminal. If the display is made to the terminal, the word PAUSE appears at the end of the last complete record for this screen of data, allowing the operator to view the status listing each time the screen is filled. The operator presses the carriage return to view the next screen of the status listing.

Discard

The discard command allows the terminal user to delete entries in the host file.

This command may be executed from a user terminal only for those jobs associated with that terminal's user ID. A discard may be performed for any job in the host file from the master terminal.

The command format is:

DISC,JN=Jmnn

Where Jmnn is a four-digit job number.

Flush

The flush command purges all jobs in the host file that are in the SENT status and that are n days old. If n days is a minus sign (-) and the host is inactive (in other words, the batch driver is not processing this host), all jobs are purged for the host.

This command may only be executed from the master terminal.

The command format is:

FLUSH,HO=host,DO=day

Where: host is a four-character name of the host.

day is a one- to three-digit number. If the number is a minus sign (-), all jobs for the inactive host are purged.

Dispose

The dispose command allows the terminal user to dispose of a job that is in the output received status. Three forms of disposal are available:

1. The PRINT option allows the master terminal operator to produce a hard-copy printout of the output job.
2. The MOVE option allows the terminal user to move the output job beginning at a designated character position to another file of a designated record length.
3. MOVEPR is a combination of MOVE and PRINT. If the output job is moved to a file with a record length of 80 characters, the new file may be processed through the text editor, but once the MOVE option is exercised, the PRINT option is no longer available.

The dispose command may be executed from a user terminal only for those jobs associated with that terminal's user ID. The master terminal may dispose of any job in the host file

The command format is:

DISPOS,JN=Jmnn,OP=option,NC=char,SC=stc,
V2=vlmlabel,FN=outflnam

Where: Jmnn is a four-digit job number.

option is a five-character name (PRINT), a four-character name (MOVE), or a six-character name (MOVEPR). There is no default for this parameter.

char is the one- to three- digit number of characters. The default value is 80. This parameter is not displayed if the PRINT option is selected.

stc is a one- to three-digit starting character. The default value is 1. This parameter is not displayed if the PRINT option is selected.

vlmlabel is a one- to eight-character volume name. This parameter is not displayed if the PRINT option is selected. This parameter defaults to the system volume.

outflnam is a one- to eight-character file name. There is no default for this parameter. This parameter is not displayed if the PRINT option is selected. The output file name must be unique and not previously defined.

Print

The print command allows an operator to print the output files that result from batch processing.

This command may be executed from the master terminal only.

The command format is:

PRINT,OP=jbno,L-lstdvnam

Where: jbno is a job number. Jmnn prints a specific job, PRxx prints an unidentified file xx, and PR00 prints all unidentified files. It can also be a one- to four-character host name to print all jobs with a print request status for a specified host.

lstdvnam is the one- to eight-character list device name, as indicated in table 2-4

ERROR MESSAGES

There are occasional instances where an error occuring during a file operation leaves that file in a locked condition and it is impossible to access that file again during the current UTIL operating cycle. The error message FILE IS CURRENTLY LOCKED is output for any operation requested for that file. If this condition occurs, exit from the utility processor and call it again. The operation unlocks any locked files and makes them available for use once again.

A list of the error messages used by UTIL is given in appendix E.

TEXT EDITOR

The ITOS text editor (EDITOR) is used to add or to change records in existing file manager files. Editing operations are performed directly on the file.

There are some restrictions on the type of file that can be edited:

- If the file is to be built initially with the text editor, the file must have previously been defined by UTIL as a direct file or a sequential file. The file uses 80-character non-sector-aligned records.
- If the file was not defined by UTIL, it must be a sequential file containing 80-character non-sector-aligned records in a format compatible with the specified compiler (RPG II, FORTRAN, macro assembler).

CAUTION

While EDITOR allows operation on both sequential and direct files, it really only operates on direct files. If the user's file is a sequential file and it is not completely full of records, the editor fills any unused records with blank records and changes the file to a direct file.

If the user requires that data records may not be blank (RPG data files may not be blank), it is the user's responsibility to ensure that the file is completely full of data before any editor operations are performed on it.

The text editor operates on a line-by-line (record) basis. The GET command specifies the file to be used. Alterations are possible in several modes:

- Changing a single line – The line is specified and the change is entered.
- Changing a specific character string in one or more lines – The operator specifies a sequence of records (lines), the old character string to be replaced, and the new character string to replace it. Then all the specified records that have the old character string are altered by replacing it with the new character string. The operator is informed of each line that is changed, but the contents of the lines are not displayed.
- Adding lines – The operator adds the text, including the line numbers in the proper field position. Each carriage return ends the current line entry. The text editor is ready for another EDITOR command when the carriage return is pressed. Special formatting is available when operating in this mode
- Adding lines with automatic line numbering – This mode is available with or without special line formatting (tabs set to define field boundaries). The text editor supplies the line number in the proper field. The operator enters data, ending the line with a carriage return. To leave automatic mode, the operator enters a carriage return at the beginning of a new line.

For text editor purposes, lines have a range of 1 through 32,767. Note that lines are not normally numbered sequentially. The default line number interval for the text editor is 10.

The text of the line is displayed in ASCII. RPG II files have the line number on the left; FORTRAN and macro assembler files have line numbers on the right. If the automatic editor mode is selected, the editor supplies line numbers for new records in the proper field position.

A tabulation mode is available to aid the user in entering (or altering) highly formatted records of the RPG II type. In this mode, the operator may skip past unused (or unchanged) fields to the field to be altered. The RESET key on the terminal keyboard is used as the tab key. A cursor positioning mode is then available to skip over unused/unchanged characters in the field.

NOTE

A previous tab field cannot be referenced by use of the left arrow (←) cursor control. Instead, RUBOUT may be used to locate the cursor at the beginning of the previous tab position.

If an editor operation has been performed in a file that could alter the line number sequence (for example, line insertion), an automatic resequence is performed in that file.

CALLING EDITOR

To call the text editor, the user replies to REQUEST=> with

ED or EDITOR (cr)

The text editor responds with:

```
EDITOR IN
READY>
```

The message READY> is repeated after each editor request is performed, indicating that the editor is ready to accept the next request.

The user may type in an abbreviation instead of typing in a full command (request) word. In this section the full command word is given for each editing request. A list of the editor commands is given in table 3-4, and the minimum abbreviation is underlined.

If an invalid command is entered, the following error message is displayed

INVALID COMMAND

followed by READY>. If an insufficient abbreviation is entered, this error message appears:

COMMAND NAME NOT UNIQUE

A list of the editor error messages is contained in appendix E.

Certain editor operations may be terminated by entering the program interrupt (CONTROL D), as described in section 2.

When data is listed on the display terminal, the message

PAUSE >

is output as the screen is filled. This allows the operator to examine the text. To continue, a carriage return is entered.

EXITING FROM EDITOR

To leave the text editor, the operator replies to READY> by typing

EXIT (cr)

The system responds with the message:

```
EDITOR OUT
REQUEST=>
```

FILE RETRIEVAL

The contents of an ITOS file are made available to the text editor by the GET command. The operator replies to READY> with:

GET,filename,R (cr)

Where: filename is the name of the desired file. It must be either a direct 80 character record file or must contain 80-character source text records. The file must not be sector-aligned.

R indicates that the file contains RPG II source text records.

The text editor locks the file during the entire edit operation, so that other terminal users cannot modify it.

CAUTION

When a file that should be usable by the text editor cannot be opened, the sequence command is provided to remedy this problem. Note that the file type must be sequential or direct and records must be 80 characters in length.

TABLE 3-4. EDITOR COMMANDS

Mnemonic †	Parameters † †	Function	Restrictions
AUTO	t, nn, ii, progid	Automatically assigns line numbers and formats new lines (records) to be added to file.	
CHANGE	*$a_1a_2 \ldots a_m$*, *$b_1b_2 \ldots b_n$*, k_1, k_2	Replaces existing character string with new character string from lines k_1 through k_2. This is the normal text modification command.	Character string can not exceed 20 characters.
CLEAR	None	Clears all references to the current file.	
COMMAN	None	Produces an abbreviated list of all text editor commands.	
CTAB	None	Clears all tab settings and sets tabs for non-RPG format.	
DELETE	k_1, k_2	Deletes all lines of text from k_1 through k_2.	
EXIT	None	Exits from editor, returns control to ITOS executive	
GET	filename, R	Finds file to be edited, opens it for use	Must be an editor-type file: 80-character records sequential or 80-character records direct if defined by UTIL.
LINE	nn, t	Change or add a line. If line nn exists, display it and position cursor at the beginning of the displayed record for corrections. If line nn does not exist, output the line number and format type in the proper format and set the cursor at the beginning of the record for entry.	
LIST	k_1, k_2	Lists the file records from lines k_1 through k_2.	
RESEQ	nn, ii	Renumbers all records in the file	
SEARCH	*$a_1a_2 \ldots a_m$*, k_1, k_2	Lists every line number between k_1 and k_2 that contains string $a_1 a_2 \ldots a_m$.	Character string can not exceed 20 characters.
SEQUEN	filename, R, m	Insert sequence numbers in the appropriate position in each record	Must be an editor-type file.
STAB	t or n,n, . . .n	Sets tab stops for text lines (records). Used with auto mode line entry.	Limited to 20 tab settings if locations are specified.

† Underlined letters are the minimum mnemonic that is acceptable.

†† Default values for each parameter are discussed in table 3-5.

TEXT ENTRY

There are two methods of adding records to a file, single line mode and automatic mode.

Single Line Mode

This mode is initiated when the operator replies to READY > with

LINE,n,t

Where: n is the line number of the line to be changed or added.

t is the format type.

H	Control card specification
F	File description specification
E	Extension specification
L	Line counter specification
I	Input specification
C	Calculation specification
O	Output specification
X	RPG trace specification
*	RPG array data (also used for /* terminator)
Blank	Non-RPG file format

TABLE 3-5. EDITOR PARAMETER DEFINITIONS

Parameter	Definition	Range
$a_1a_2 \ldots a_m$	Old character string	From 1 to 20 characters
$b_1b_2 \ldots b_n$	Replacement character string	From 1 to 20 characters
filename	Name of file to be edited	1 to 8 characters
ii	Line interval	Equal to or greater than 1; 10 is default.
k_1	Starting line	Equal to or greater than 1; 10 is default.
k_2	Ending line	Greater than starting line.
m	Line number and interval	Equal to or greater than 1; 10 is default
n	Character position within a line	1 to 80
nn	Line number	1 to 32,767
progid	Program ID	1 to 6 characters
R	Indicator for a file containing RPG II source text records	–
t	Format type	H,F,E,L,I,C,O,X,*, or blank
*	Delimiting character for a character string.	May be any character except comma (,).

In this mode the operator may modify an existing line of data, insert a line of data between existing lines, or add a line at the end of the file.

When one of the RPG format types is used, the appropriate tab positions for that format type are set up. (See appendix G.) If the format type is not specified, a non-RPG format is assumed that retains the last specified tab setting, if any, defined by the STAB command. The default tab settings for non-RPG format are 1 and 76. Any other specification for the t parameter results in an illegal format specification error.

To change an existing line, the operator enters the line number and the format type. The line is displayed on the screen with numbers appearing above and below the displayed line to indicate character positions. The cursor is positioned at the first tab position for this format type. The operator may re-enter the entire line at this point or he may use the tab and cursor position keys to correct any errors within the line. Only those characters actually entered are changed in the original record.

To add a new line, the operator enters the new line number and the format type. A blank line with the line number and format type, if necessary, is displayed on the screen. Numbers appear above and below the line to indicate character positions. The cursor is positioned at the first tab position for this format type. The operator may then enter the new line of text.

Auto Mode

Using one of the RPG format types, certain fields are preset and line numbers are automatically generated. Numbers appear below the line indicating character positions. The cursor is positioned at the first tab position for that format type. Each time AUTO is entered, the screen is cleared and a position header appears at the top of the screen.

To activate the automatic mode, the operator answers READY> with

AUTO,t,nn,ii,progid

Where: t is the format type

H	Control card specification
F	File description specification
E	Extension specification
L	Line counter specification
I	Input specification
C	Calculation specification
O	Output specification
X	RPG trace specification
*	RPG array data (also used for enter of /* terminator)
Blank	Non-RPG type

n is the base line number (the default is 10).

i is the line number increment (the default is 10).

progid is the program identification. This field is required on RPG type files for format type H. It is six positions long, and may appear on other RPG format types but is not required. It appears as positions 75 through 80 of the record.

The entry of any of the nine RPG II format types also sets the appropriate tab positions for the entry of source text (see appendix G). If t is blank, a non-RPG format is specified. This retains the last specified tab setting, if any, as defined by a STAB command. The default tab settings are 1 and 76. Any other specification for the t parameter causes an illegal format specification error.

The program identification is optional for non-RPG type files. If it is included it may not exceed three characters and is truncated if it exceeds three characters; it appears in positions 73 through 75 of the record.

NOTE

If n is blank, and records exist in the file, the next record is numbered m+I where m is the line number of the current last record and I is the specified increment.

Examples:

To change an existing line the operator replies to READY> with

```
LINE,40 F
```

The screen is cleared, the character position indicator lines are output and the existing line is displayed:

```
1234567..................  7890
00040F>TIFIL  IF F 80 DISKCYB0060
1234567..................  7890
```

The cursor and > are at position 7. The operator moves the cursor to the position in error with the tab and cursor position keys and then corrects the error.

To add a new line the operator replies to READY> with

```
LINE,45,F
```

The screen is cleared, the character position indicator lines are output and the line number and format type for the new line are displayed:

```
1234567890123.....  .....  ... 67890
00045F>                      CYB060
1234567890123.................  67890
```

The cursor and > are at position 7. The operator may enter the new line.

To enter an RPG H format record, the operator replies to READY> with

```
AUTO H,,,progid
```

The screen is cleared and the display is:

```
1234567890.....................1234567890
00010H>                              progid
1234567890.....................1234567890
```

The character positions appear at the top of the screen and below the data line. The cursor and > are at position 7, the first tab position for RPG format type H. Since this is a new file (no records) and nn and ii are omitted, nn is 10, and ii is 10.

Under AUTO with non-RPG type format, the functions are similar but there are no tabs for non-RPG format types. For example, when the operator enters:

```
AUTO,,,,TID
```

The screen is cleared and the display is:

```
1234567890.................  .....34567890
>                                 TID00010
1234567890.....................34567890
```

The cursor and > are at position 1. The program ID and page/line number appear in positions 73 through 75 and 76 through 80, respectively. The next line will be 00020 since default for both nn and ii is 10. This is a new file (no records).

After the operator enters a line of text terminating with a carriage return, the next line is displayed. The operator either enters text, continuing the process, or enters a carriage return, returning the editor to the ready condition.

INITALIZING EDITOR

To initialize the editor so that a new file may be processed, the operator replies to READY> with

CLEAR (cr)

DELETING TEXT

To delete a specified record (11) from the file, the operator enters

DELETE,k_1 (cr)

To delete a number of consecutive records of text, the operator enters

DELETE,k_1,k_2 (cr)

where k_1 and k_2 are the first and last lines to be deleted. Trailing commas need not be entered.

NOTE

If k_1 and k_2 do not designate existing lines, the editor uses the next higher existing line.

RESEQUENCING TEXT LINES

To resequence the line numbers of the file, the operator enters

RESEQ,nn,ii (cr)

Where: nn is the base line number where resequencing begins.

ii is the increment between successive lines of text.

The starting line is given line number nn, and successive lines are numbered nn=ii, nn=2ii, . . . If nn and/or ii are

omitted, the values are assumed to be 10. Trailing commas need not be entered.

LISTING TEXT LINES

To list all or part of the file, the operator enters

LIST,k_1,k_2 (cr)

where k_1 and k_2 are the first and last lines to be listed.

The editor lists the contents of lines k_1 and k_2. If k_2 is omitted, only line k_1 is listed. If both k_1 and k_2 are omitted, the entire file is listed. If k_2 is larger than the largest line number previously entered, the listing ends at the last line of the file. Trailing commas need not be entered. If k_1 is greater than k_2, a message indicating an invalid command is displayed.

NOTE

As the screen is filled, text listing is suspended by the PAUSE message. A carriage return must be entered to continue. Text listing may be interrupted at any time by the program interrupt (CONTROL D) procedure described in section 2.

TEXT MODIFICATION

Text may be modified by replacing a line of text with a new line (refer to the LINE command).

If the operator wishes to change only part of a line or to substitute every occurrence of a character string in the file with another character string, he replies to the READY> message with

CHANGE,*$a_1a_2 \ldots$ am*,*$b_1b_2 \ldots b_n$*,k_1,k_2 (cr)

Where:

- $a_1a_2 \ldots a_m$ is the character string to be replaced.
- $b_1b_2 \ldots b_n$ is the new character string to be inserted.
- * is the delimiting character for the character strings. The delimiting character may be any valid keyboard character with the exception of a comma (,).
- k_1 is the starting line to be changed.
- k_2 is the ending line to be changed. If k_2 is omitted, only k_1 is changed; if k_1 and k_2 are omitted, every occurrence of the character string in the entire file is changed. If k_2 is smaller than k_1, an error message is displayed.

It is not necessary that m equal n. However, both m and n must be less than or equal to 20. If m is greater than n, care should be taken to ensure that text characters or line numbers are not lost at the end of the line when the existing line is expanded to accept the new text. Characters beyond the new 72nd character are truncated on a one-for-one basis. Trailing commas should be omitted when parameters are omitted.

As the substitutions are made, the message

CHANGES HAVE OCCURRED ON LINES

is displayed, followed by a list of line numbers. The line number is listed once for every substitution that has occurred, so repeated line numbers represent multiple changes on the same line. If the substitution results in a line that is larger than the original, the message

LINE TRUNCATED

is appended to that particular line number and the data preceding the line number is lost.

As the screen is filled, line number listing is suspended by the PAUSE message. A carriage return must be entered to continue.

TEXT SEARCH

If the operator wishes to examine the text to find those places where a given character string occurs, the search procedure may be used. To search text for character string a1a2 . . . an, the operator replies to the READY> message with

SEARCH,*$a_1a_2 \ldots a_m$*,k_1,k_2 (cr)

Where:

- $a_1a_2 \ldots a_m$ is the character string being sought.
- k_1 is the first line being sought.
- k_2 is the last line being sought. The file is searched from line k_1 through line k_2. If k_2 is omitted, the rest of the file beyond k_1 is searched. If both k_1 and k_2 are omitted, the entire file is searched.

Trailing commas should be omitted when parameters are omitted.

As strings are located, the message

STRING FOUND IN LINE

is displayed, followed by a list of line numbers. The line number is listed once for every occurrence of the string, so that repeated line numbers indicate multiple occurrences of the string in the same line.

As the screen is filled, line number listing is suspended by the PAUSE message. A carriage return must be entered to continue. The listing may be interrupted by entering CONTROL D, as described in section 2.

TABULATION

The text editor provides the ability to set tab stops in various character positions within the record. This function is used to set tab stops when entering records in a nonautomatic mode. It is especially applicable to

formatting RPG II records. To set the tabs, the operator replies to READY> with

STAB,t (cr)
or
STAB,n,n,n, . . . n (cr)

Where: t is the format type. This sets tab stops for various RPG form specifications. Positions of tabs are given in appendix G.

H	Control card specification
F	File description specification
E	Extension specification
L	Line counter specification
I	Input specification
C	Calculation specification
O	Output specification
X	RPG trace specification
*	RPG array data (also used for entry of /* terminator)
Blank	Non-RPG type

n is various character positions within a line. These positions are limited to a maximum of 20 and must be entered in ascending order.

The default condition for the editor specifies tabs in positions 1 and 76.

To clear tabs, the operator replies to READY> with

CTAB

This leaves the editor with tabs specified in positions 1 and 76 (non-RPG format).

NOTE

In AUTO mode, the selected format sets the tabs for that format automatically.

CHARACTER POSITIONING

ITOS itself provides for repositioning a character in the same line (backspacing). This is activated by the ← key, which moves the cursor back a single position each time it is depressed. By using this key line corrections are easily made.

The text editor provides a forward positioning function. This uses the → key and moves the cursor forward a single position. Characters of the current line (record) are not changed as the cursor moves past them.

COMMAND LIST

The operator may obtain an abbreviated list of all editor commands by replying to the READY> message with

COMMAN

The list appears as:

```
AUTO,TYPE,BASE,INCR,IDENT
CHANGE,OSTR,NSTR,STRT,END
CLEAR
COMMAN
CTAB
DELETE,STRT,END
EXIT
GET,FN,R
LINE,NMBR,TYPE
LIST,STRT,END
RESEQ,BASE,INCR
SEARCH,STR,STRT,END
SEQUEN,FN,R
STAB,TYPE/N1,N2, . . . ,N20
```

SEQUENCING UNNUMBERED TEXT FILES

To insert sequence numbers into records of a file to be processed by the text editor, the operator enters

SEQUEN,filename,R,n

Where: filename is the name of the desired file.

R indicates that the file contains RPG II source text records. If R is omitted, filename is assumed to be a non-RPG file.

n is optional. If it is included, n specifies both the base line number and the interval between lines. Default value is 10.

NOTE

Failure to include the R indicator on an RPG II source text file may result in the destruction of that file.

This command places a sequence number in each text record in the appropriate position for that file type. The base line number and increment are both 10.

SORT UTILITY

The sort utility (DSORT) is called only from a procedure stream (that is, it is not used interactively). A single call to sort allows one or more files to be sorted by use of one or more key values. Records in different files must be of the same length. Note that there is only one output file regardless of the number of input files.

The capabilities of sort are:

- Sorting one or more files that have equal record lengths. The number of records in each file is immaterial.
- Sorting by arbitrary keys. There are no restrictions on key field of length. These keys need not be (and usually are not) the keys that are defined for file manager indexed files.
- Hierarchy of sorts by several keys. The first key presented causes the primary sorting, the next key presented sorts within the primary sort, the third key sorts within secondary sort, and so forth.
- Keys may overlap.

- The single sorted output file contains one of the following:

  Full records (tag-along sorts)

  -Only the portion of the record that remains after keys are removed (data sorts). In data sorts, records are compressed by omitting keys.

  -ADDROUT sorts that contain only the file manager's record numbers pointing to the input file records. ADDROUT sorts can process only one input file per sort.

- Records may be selectively included or excluded from the file. Criteria for this selection are based on a single key (which need not be the same as the other keys used for the sorting) having a specified relation (=, ≠, >, <, ≥, or ≤) to another key in the record or to a specified Hollerith constant (for instance, all records with 'dept. 0510' could be excluded from the sort. Conversely, only those records including 'depth. 0510' in the specified key would be included in the sort operation.)

- Data in records is normally in ASCII and the normal collating sequence used for sorting is ASCII. However, the sorting may be made in EBCDIC code, if desired.

Sorting is controlled by a series of contiguous commands called input directives. These directives (statements) start in column 1. Blanks are meaningful; they should appear only in the file or owner name in the Hollerith constant field, or as a separator for comment information. Comments may appear on any card after the required information has been provided.

The procedure stream that executes DSORT is summarized in table 3-6.

CALL THE UTILITY

The initial call to the sorting utility has the format:

```
DSORT
```

DEFINE INPUT FILES

Each file is defined on a separate input record (card). The format of the statement is as follows:

```
FN=filename,owner
 :
 :
FN=filename,owner
```

Where: filename is the name of the file. The field is up to eight characters in length. Leading blanks that are a part of the name are required; trailing blanks may be omitted.

owner is the file owner's name. Leading and trailing blanks follow the same rule as for the filename parameter.

NOTE

SORT does not follow the usual ITOS restrictions on use of files by owner name. Any file, regardless of owner, may be sorted by any user. In cases where several input files are sorted, these files need not have the same owner, nor need the owner be the same as the user ID that was entered at log-in time by the user who calls the procedure stream that starts the sorting operation.

If the owner field is left blank, common files are specified. That is, if the FN statement omits the owner, it must still have a comma following the filename.

It is assumed that the filename/owner are unique for all volumes on the system. If this is not the case, the file processed will be the file on the first volume encountered by the file manager that has that filename/owner.

The record length for all input files must be the same.

There is no theoretical limit to the number of input files for a sorting operation, except for ADDROUT sorts (ADDROUT can sort only one file). However, the mass storage space required for a sort may be as great as three times the size of all the input files used. The workspace required for sorting is defined by the SYSDAT variable WKSPLU, which is an installation time parameter.

DEFINE OUTPUT FILES

The file may be one that has been defined previously. If so, that file is released prior to storing output records and is redefined as a sequential file. The format of the statement is as follows:

```
F2=filename,owner,volume
```

Where filename is the file name, up to eight characters in length. Leading blanks, if any, must be retained. Trailing blanks are optional.

owner is the owner name, up to eight characters in length. It follows the rules for owner specified in the input file statement above.

volume is the name of the volume that will receive the output file. It is up to eight characters long. If it is omitted, the volume defined by WKSPLU is selected to receive the file.

NOTE

If both owner and volume are omitted, all commas must still be included; for instance,

```
F2=FILE1,,
```

TABLE 3-6. CONTROL STATEMENTS FOR DSORT

Statement	Comment
Call SORT	
DSORT	Prepares for sorting operation
Input File(s) Definition	
FN=filename,owner ⋮ FN=filename,owner	filename Up to eight characters owner Up to eight characters Statement per input record.
Output File Definition	
F2=filename, owner,volume	filename Up to eight characters owner Up to eight characters volume Up to eight characters
Selecting Sort Options	
OP=ADDR/TAG,F/D,A/E	ADDR/TAG or A/T: A selects ADDROUT sort; T selects tag-along sort F/D: F selects full record output; D selects data portion of record only A/E: A selects ASCII sort; E selects EBCDIC sort
Designating Keys Used for Sorting	
KF=A/D,keycol,keycols,A/D,keycol,keycols, . . . ,A/D,keycol,keycols	A/D: Ascending or descending sort keycol: Starting column of key keycols: Length (in characters) of key
Record Selection	
SL=OMIT/INCLUDE,keycol1,keycols1,op, {'Hollerith constant' keycol2}	OMIT/INCLUDE or O/I: O specifies omit these (and no other) records; I specifies include these (and no other) records. keycol1: Starting column of key 1 keycols1: Length (in characters) of key 1 and key 2 op: Comparison factor: EQ - K1 = K2 GT - K1 > K2 NE - K1 ≠ K2 LE - K1 < K2 LT - K1 < K2 GE - K1 ≥ K2 'Hollerith constant': Value of key 2 is the 1 to 20 Hollerith characters delimited by ' '. keycol2: Designates starting column of key 2. Length of key 2 is assumed same as length of key 1.

The output file name can be the same as one of the input file names. If the output file is predefined, and if the total record count is greater than DSORT requires, the total record count for the existing file is used. If the output file has a total record count less than DSORT requires, DSORT recreates the file with a total record count equal to the number of records sorted.

SELECT SORT OPTION

The options statement must be included. It has the following format:

OP=ADDR/TAG,F/D,A/E

Where: ADDR/TAG (may be abbreviated A or T) specifies whether the output file includes only relative record number (ADDR) or includes tagalong (TAG) data. ADDR can be used only if a single input file is designated.

F/D further defines the TAG parameter. The F option specifies that the entire input record (including the keys used for sorting) is placed in the output file. The D option specifies that the keys are eliminated from the record, and only the remaining data appears in the output record. (If the keys

occurred before the end of the record, the remaining data is moved to the left to omit the space formerly occupied by the keys.)

A/E — specifies the collating sequence used for sorting. A signifies ASCII, E signifies EBCDIC. The ASCII/EBCDIC collating sequence is shown in appendix H.

DESIGNATE KEYS USED FOR SORTING

One key statement defines all keys to be used. Each key requires three consecutive parameters. The order of the keys in each key statement determines the sorting hierarchy: the first key presented is the primary key. Records are first sorted by this key, then they are sorted according to the second key presented, then by the third key presented, and so forth, until the records have been sorted by each of the specified keys. Records not distinguished by the entire series of key sorting may appear in any order (not necessarily in input order).

The key information can be extended to a new card if column 72 is reached.

The key statement has the following format:

KF=A/D,keycol,keycols,A/D,keycol,keycols, ...,
A/D,keycol,keycols

Where: A/D — Indicates the collating order to be used.

A = ascending
B = descending

keycol is the character position within the record for the first character of the key. The first character position of the record is numbered 1.

keycols is the length of the key in characters. The minimum key length is a single character.

Keys may overlap. The key statement should not terminate with a comma.

RECORD SELECTION

This statement allows the user to select one key in the record as a basis for including or excluding records. The statement has the following format:

SL=OMIT/INCLUDE,keycol1,keycols1,op,
{'Hollerith constant' / keycol2}

Where: OMIT/INCLUDE may be abbreviated to O or I. For OMIT, all records meeting the criteria are excluded. For INCLUDE, only those records meeting the creteria are included.

keycol1 is the starting character position of key 1, the key that governs the record selection.

keycols1 is the length (in characters) of key 1 and, if it is used, also the length of key 2 or Hollerith constant.

op is the comparison type

EQ Key 1 = key 2/Hollerith value
NE Key 1 ≠ key 2/Hollerith value
LT Key 1 < key 2/Hollerith value
GT Key 1 > key 2/Hollerith value
LE Key 1 ≤ key 2/Hollerith value
GE Key 1 ≥ key 2/Hollerith value

'Hollerith constant" is any value enclosed by single quote marks (' '). It is 1 to 20 characters in length.

keycol2 is the starting character position of key 2. Length of key 2 is keycols1.

Note that either 'Hollerith constant' or key 2 can be used for the comparison.

The designated key (field of length keycols1 beginning at keycol1 character position) may or may not be the same as that of the sorting keys designated in the KF statement. The value of this key is compared either to a second key in the record (field of length keycols1 beginning at keycol2 position) or to a Hollerith constant. The second key may or may not be one of the sorting keys designated in the KF statement. The comparison may be equality or one of several types of inequality. As a result of this comparison, which is made to the input records, either the records that meet the criteria are included and all other input records are excluded, or the records that meet the criteria are excluded and all other input records are included.

OPTIONAL COMMENTS STATEMENTS

Comment cards (statements) have a comments indicator and a comment in ASCII in the remaining columns. There may be one or several comment cards. It is recommended that one comment card be used to precede the DSORT card (for instance, START OF SORT FOR NEWFILES), since this statement clears the screen and shows that any error messages that follow relate to a sorting operation and not to some previous operation that may still display a message on the screen. In addition, comments may be included on any statement card following the blank (separator) that ends the required input information.

TERMINAL MESSAGES

The following is a typical output at the display of the terminal that is executing the program with a single input file that called the sorting operation:

Display	Comments
VOLUME=SYSVOL	The input file is on the system volume.
FILNAM=OLDFILE, ownername	File to be sorted
PASSED=162	162 records were read
DONE=50	50 records were processed
VOLUME=SYSVOL	The output file is on the system volume.

Display	Comments
FILNAM=NEWFILE7, ownername	The name of the output file
PASSED=50	50 records were output.
DONE=50	50 records were output.

If the sorting operation fails, one of the error messages listed in appendix E is displayed at the terminal.

SAMPLE PROCEDURE STREAM FOR SORT

Procedure Stream	Comments
*SORT OF 3 NEWFILES BEGIN	Comment card to clear screen and to alert operator that sorting of the newfiles will begin immediately
DSORT	Procedure call
FN=ΔΔNEWFL1,SMITH FN=NEWFL2, FN=NEWFL3ΔΔ,SMITH	Files can belong to different owners; all files have equal length records.
F2=ΔΔOUTFL1,,	Output file is a common file. It appears on the volume specified by WKSPLU.
OP=T,F,A	Output file contains full input records, is sorted using ASCII collating sequence.
KF=A,15,10,D,5,10, A,25,20	Three 10-character keys are used. Primary key begins at position 15, secondary key at position 5, and tertiary key at position 25. First and third keys use ascending sorting, second key uses descending sorting
SL=O,40,2,EQ,'CA'	Omit all records that have characters CA in columns 40 and 41. Include all others.

Terminal responses, normal or error, follow.

The ITOS executive sequences and controls user programs and terminal operations. It intercepts all requests from user programs and from terminals, and provides the interface between user programs and the following:

- The MSOS monitor and drivers
- The file manager
- The communications line I/O driver

The ITOS executive, operating as a protect processor, validates all user programs and requests, including requests made from the terminals in interactive mode. Validation prevents one user program from interfering with another user program or with that program's data.

System throughput is improved by overlapping user program execution with I/O terminal operations, file manager processing, and swapping of programs between main and mass memory.

The executive makes use of the memory protect bounds registers and the extended memory paging features of the CYBER 18 computers. An overall block diagram of the executive is shown in figure 4-1.

The executive consists of three functionally distinct processors: the user protect processor, the user task processor, and the input/output completion processor. These three subsystems operate in concert to sequence and control the execution of user programs.

All user programs are executed in unprotected memory, based on the setting of the protect bounds registers. Each time the user program performs a system request, a program protect interrupt occurs that is serviced by the user protect processor. This routine validates the request, together with all request parameters that could interfere with other user programs or the remainder of the system. Once the request is validated, all necessary parameters are moved from the user area so that the request may be performed independently of the memory page configuration. The protect processor then performs the required function by making an MSOS system request. Once the request is accepted, the user task processor is entered to start another user program.

The function of the user task processor is to sequence and control user program execution. Program sequencing is provided by a set of three queues: one controlling the execution of all programs in main memory, one for all programs swapped on mass memory, and one for all programs in main memory that are eligible to be swapped.

Each time it is entered, the user task processor attempts to remove the user program from the top of the mass memory queue by unswapping it and, if this procedure is possible, making a mass memory transfer request. Once the request is accepted, the task processor removes a user program from the top of the main memory queue and places it into execution after mapping its physical memory into the logical memory area reserved for user program execution. In this way, the mass memory I/O activity required to remove a user program from a swapped state is performed in parallel with another user program's execution. When the swapped user program has been read into main memory, its position on the various queues is updated so that it may subsequently be placed into execution.

If the task processor finds no user program currently ready for execution, it lapses into an idle condition to await the completion of the swapped user transfer or some other external event, such as user program I/O completion.

The input/output completion processor is responsible for responding to all external events associated with user program execution. This consists of all user-generated input/output completions, as well as the expiration of the user program execution timeslice and the completion of unswap mass memory read requests. The completion processor operates at a priority level above that of the user program, protect processor, and task processor.

When an I/O completion occurs, the processor places the user program on the appropriate queue, since during the input/output operation the user was suspended and did not appear on any execution queue. In addition, the completion processor updates a status word and a state index, which are associated with the user program. Finally, if the task processor is in an idle condition, it is initiated by the completion processor. Otherwise, the completion processor goes into an idle state to await another event.

To control the length of execution of user programs that do not perform frequent input/output, a timeslice delay is initiated every time a user program is put into execution. If this delay expires, the completion processor removes the user program from execution and places it on the appropriate queues. It then initiates the task processor so that a new user program can be started.

A subsidiary executive function is concerned with logging on and logging off terminal users and initiating user-requested programs. This is performed by the log-on processor, which is executed in unprotected memory in the same manner as the user program. A special interface to the protect processor is provided to allow the log-on processor to perform its required tasks.

EXTENDED MEMORY MANAGEMENT

The executive makes use of a facility that allows convenient operation with the extended memory available with the CYBER 18 computer systems. The facility involves the use of memory control points that specify the mapping of physical memory to logical memory.

The CYBER 18-20 computer may contain up to 256K bytes of physical memory storage, but only 128K bytes may be logically addressed at any time. Mapping a physical memory configuration into logical memory is performed by the executive as indicated in figure 4-2. This example indicates how user programs may be suspended by being mapped out of logical memory. Note also that user number 1 consists of two physically distinct segments. This represents the multiuser/root configuration described later in this section.

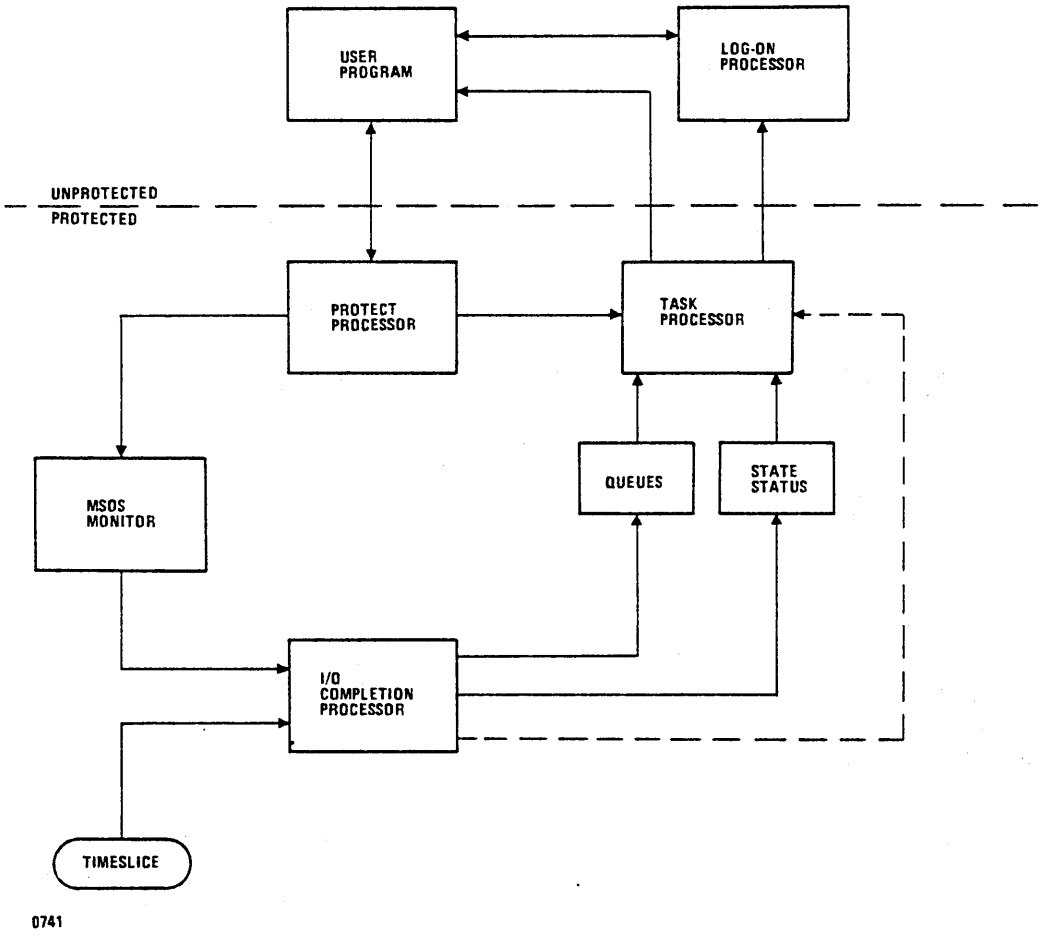


Figure 4-1. ITOS Executive Functions

The use of control points allows the executive/MSOS system to perform the following necessary functions:

- Execute mass memory (direct memory access) transfers to/from physical memory locations that are not currently in the logical memory space.
- Allow the movement of data between physical memory locations that are not in the logical memory space existent at the time the movement is required.

The above functions are required to efficiently perform both word-addressed mass memory transfers and file manager requests in ITOS.

A control point is defined as the arrangement of the page memory file that results in the logical memory addressed by a user program. These control points may be defined, modified, released, activated, and deactivated by the ITOS executive.

Several areas of MSOS are cognizant of control points to allow efficient use of the system resources by the executive.

The mass memory I/O driver makes use of control points so that direct memory access transfers may occur in parallel with user program execution. The following restrictions are applied to the requesting program to allow mass-memory requests to utilize extended memory control points:

- The monitor request must be indirect, and the parameter list must not reside in a control point area.
- The I/O data buffer must be located in the control point that is active at the time the monitor request is made, so that the parameter list specifies a logical address.
- The current control point value must be contained in the location directly following the least significant bit of the mass memory address (word 8 of the parameter list). In addition, this must be indicated by setting bit 15 of the most significant bit (word 7) to 1.

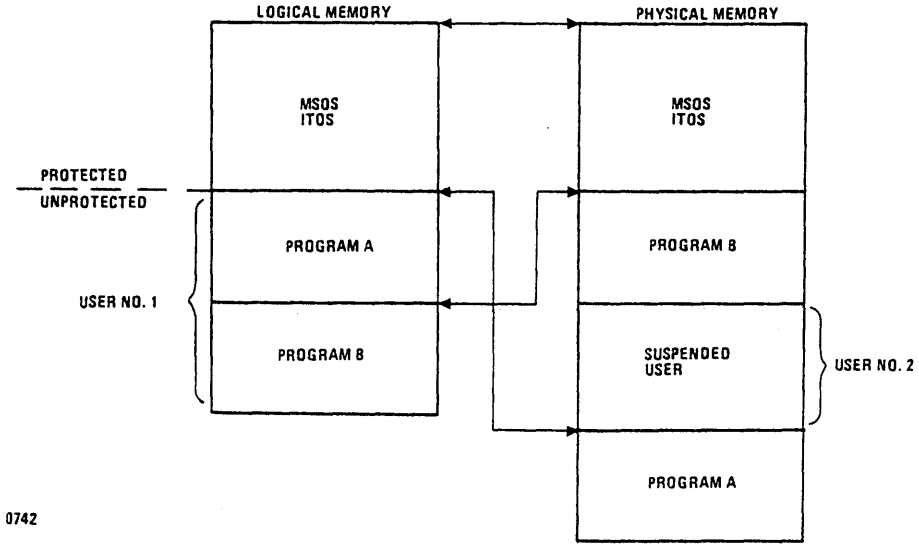


Figure 4-2. Memory Mapping

These restrictions do not affect the user program, since the executive intercedes between the requestor and the MSOS monitor and actually performs the I/O request.

The file manager also has the capability of processing requests that reside in extended memory. The following restrictions are applied to the user of the file manager so file requests may utilize extended memory:

- Words 0 through 3 of the file request buffer must not reside in a control point area. The request itself and the emainder of the parameters may reside there, however.
- The file request must pertain to the control point that is currently active, so that all parameter addresses are current logical addresses.

These restrictions do not affect the ITOS programs, since the executive intercedes between the requestor and the file manager and actually makes the file request.

The current control point is saved by the MSOS monitor when a program is interrupted by a higher priority level, and it is restored when the interrupted program is returned to execution. The control point is not restored when a program regains control after exiting to the dispatcher (for example, at an I/O completion).

MEMORY ALLOCATION

The executive must manage work areas in both main memory and mass memory Main memory areas must be allocated and released during the course of user program execution. Mass memory is allocated and released during the swap and unswap operations performed by the executive.

Blocks of main memory may be reserved for non-ITOS use (for example, for batch or COMM 18).

The minimum main memory allocation is 4096 bytes (one memory page), and it is allocated on a first-fit basis. Allocation requests are divided into two classes: large and small. All large requests are served by obtaining the lowest address piece that fits, and all small requests are served by obtaining the highest address piece that fits. This strategy tends to separate large blocks from small blocks and alleviates memory fragmentation. The definition of large and small is a system parameter and is adjusted to separate multiuser routines from their root programs.

The minimum mass memory allocation is 16,384 bytes, and is allocated on a first-fit basis. The pool area used for swapping is defined as a system file ($$SWPBUF).

USER EXECUTION

Control of user execution is maintained by means of three items associated with each user: the user program state, the user input/output status, and program control queues. These form the primary means of communication between the several modules of the executive.

USER STATES

A user program can exist in any of the 29 states allowed by the executive; a list of these is shown in table 4-1. The state index is used by the executive during user program control.

USER INPUT/OUTPUT STATUS

An input/output status word is maintained for each terminal that contains information concerning the current activity on that device.

TABLE 4-1. USER STATES

State Index	Definition
0	Not defined
1	Executing in main memory
2	Suspended in main memory – Timeslice complete
3	Suspended in main memory – Mass memory I/O active
4	Suspended in main memory – Mass memory I/O complete
5	Suspended in main memory – File I/O active
6	Suspended in main memory – File I/O complete
7	Suspended in main memory – Terminal I/O active
8	Suspended in main memory – Terminal I/O complete
9	Suspended in main memory – Data I/O active
10	Suspended in main memory – Data I/O complete
11	Suspended in main memory – Attach active
12	Suspended in main memory – Attach complete
13	Reserved
14	Swapped on mass memory – Timeslice complete
15	Reserved
16	Swapped on mass memory – Mass memory I/O complete
17	Reserved
18	Swapped on mass memory – File I/O complete
19	Swapped on mass memory – Terminal I/O active
20	Swapped on mass memory – Terminal I/O complete
21	Swapped on mass memory – Data I/O active
22	Swapped on mass memory – Data I/O complete
23	Swapped on mass memory – Attach active
24	Reserved
25	Suspended on mass memory – Unswap active
26	Suspended in main memory – Unswap complete
27	Suspended on mass memory – Multiuser read
28	Suspended on mass memory – Initial log-in
29	Suspended on mass memory – Job step

In the status word, a 1 indicates that the condition is true, and a 0 indicates that it is false.

Group	Bit	Meaning
Input/output request type	12	Terminal I/O request
	11	Mass memory I/O request
	10	Input/output complete
	9	Input/output active
	8	Data input request
Input/output error	6	File request error
	5	Mass memory error
	4	Terminal disconnect
Unsolicited input	2	Terminal escape request
	1	Manual interrupt request
	0	Terminal log-on request

USER PROCESSING QUEUES

Control of user execution is based on three threaded queues that are used by the executive to determine the order in which user programs are executed.

Each queue entry consists of the thread to the next entry, and an entry of -0 terminates the queue. The top of each queue is located in the MSOS SYSDAT and is referenced by an entry point name. Each queue contains four priorities, and each priority has a separate starting address in SYSDAT.

The queues are threaded through the user table address, which contains two thread words: one for the execution thread and one for the swap thread.

- Next executing user (core) – Each time a core-resident user's input/output is completed, a user timeslice expires, or the user program is unswapped, it is threaded onto a queue with the entry point name NXUC. This is a first-in/first-out queue with one priority.
- Next executing user (mass) – Each time a user program is swapped, its input/output is completed (and it is not core-resident), or a mass-resident program is initiated, the user is threaded onto a queue with the entry point name NXUM. This is a first-in/first-out queue with four priorities. These are, from highest to lowest:

  4 Multiuser program initiated (ATTACH)

  3 Terminal read complete or new program initiated

  2 Not used

  1 User I/O complete

- Next swapped user – Each time a user program initiates input/output, has a timeslice expire, completes mass memory input/output, or performs an ATTACH request in which the multiuser is not resident, the user may be threaded onto a queue with the entry point name NSWP. This is a last-in/first-out queue with one priority.

Note that user programs performing mass memory I/O are not eligible for swapping until the I/O is complete.

Placement of the user program on the NSWP queue is conditional; this allows the program to perform several system requests prior to being swapped. A set of request limits are located in the MSOS SYSDAT and, together with a user program request counter, are used to determine if the user is to be placed on the NSWP queue:

- The user program is never placed on the NSWP queue if its request counter is less than the lower limit.
- The user program is always placed on the NSWP queue if it is performing terminal input, or if its request counter is equal to or greater than the upper limit.
- The user program is not placed on the NSWP queue if its request counter lies between the two limits, unless there is an entry on the NXUM queue at priority 3 (terminal input active).

USER SWAPS TO MASS MEMORY

User swaps are required whenever the main memory resource available to the executive is exceeded. There are two operations used by the executive during swap processing.

SWAP

A swap is the process of writing one or more user programs to mass memory to make execution memory available to another user.

UNSWAP

Unswap is the process of reading a user program into main memory so that it may be placed into execution. There are two forms of this operation, each of which is attempted in turn.

- A normal unswap occurs when there is a block of unoccupied execution memory sufficient to contain the user program.

- A collective swap consists of writing several user programs to mass memory to obtain a block of memory large enough to contain the user program.

If the executive fails in its attempt to perform the unswap, all queues remain unchanged and the process is repeated at the next opportunity.

MULTIUSER PROGRAMS

The executive contains the capability of executing serially re-entrant user programs. These programs are divided into two distinct areas: the root portion and the multiuser portion.

The root portion contains all user-dependent variables, the execution control program, all I/O routines, and any nonre-entrant code. The multiuser program contains all common routines required to perform the user program's function. During execution, one multiuser program is shared by several roots, thus allowing additional users to reside in main memory simultaneously.

By using the page memory system and the control point facility, the multiuser program and its roots may be physically separate in memory, but are logically adjacent during execution.

Each multiuser program is defined by an index associated with a table located in SYSDAT. This table contains unique information about the multiuser program, such as its location in memory and the number of active users currently attached to it.

The root and the multiuser program are loaded and absolutized as separate files in the program library. The multiuser program load must contain a copy of the root so that all external references from the multiuser program may be resolved. The root must be loaded as a single program, so that external references from it to the multiuser program are not allowed.

A single root may be associated with several multiuser programs. The only restriction is that all multiuser programs that are run with a set of roots must begin at the highest possible page boundary in logical memory so that all address references are the same. A special loader command (*PAGE) is provided to force the starting address of a multiuser program to the desired location.

When a multiuser function is requested, the executive initially transfers control to the root. The root must contain the logic necessary to attach it to the desired multiuser program. Once the attachment is complete, the program may operate as a single entity. When the program requests I/O or completes its execution period, the root is suspended. The next program placed into execution may be a root that uses the same multiuser program, in which case the new control point combines the root and multiuser into the proper configuration. When the original root is restored to execution, the executive passes control to the completion address as with a normal user program. However, if the root was swapped during its suspension, the executive must transfer control to the beginning of the root, since the multiuser program may no longer be resident in memory. Thus, the root must contain logic to repeat the attachment that existed at the time of suspension. In this event, the executive supplies the root with the original completion address so that execution may continue normally following the attachment.

Since it may not be possible to design the multiuser program to be re-entrant with respect to timeslice expiration, the executive does not interrupt a multiuser program when its timeslice expires. Instead, the global system variable named EXPIRE is set to nonzero. The multiuser program must frequently examine this variable and, if it is found to be set, must allow itself to be interrupted at the next convenient point. While this can greatly simplify the design of a multiuser program, great care must be taken to ensure that EXPIRE is examined frequently so it does not distort user terminal response.

The executive contains two externally-called subroutines that allow multiuser programs to operate in ITOS.

ATTACH

This subroutine allows a user root to move from one multiuser program to another and to correctly re-attach to the multiuser program following a swap to mass memory. The root is automatically detached from the multiuser program upon program termination.

The calling sequence is:

```
CALL ATTACH (idx, iloc)
```

Where: idx is the multiuser index.

iloc is a location that may be used to continue execution if control is passed to the start of the root after an unswap of the multiuser program.

The ATTACH subroutine saves and restores all user registers and requires that all parameters be absolute.

SLICUP

This subroutine allows pseudo re-entry by the multiuser program. The only time the executive is allowed to interrupt a root is when an I/O request is made and when control is passed to the SLICUP subroutine.

The calling sequence is:

```
CALL SLICUP
```

All registers are saved and restored by the subroutine.

INPUT/OUTPUT DEVICE MANAGEMENT

The ITOS executive allows the terminal operator and the user program the capability to access system peripheral devices logically or directly from the master terminal.

User programs may also control the position of the cursor on the terminal display.

DEVICE ACCESS

All non-mass-memory I/O requests are buffered through the user's terminal line buffer, which is normally 136 bytes in length. However, each line buffer may be individually sized to accommodate specific device requirements. The size of the buffer is set at customization time as a parameter in SYSDAT

The executive places the number of characters actually read during an MSOS input operation into the last word +1 of the requestor's buffer, even if the specified number of characters is read. For this reason, all input data buffers must contain at least one word more than is required for the input request.

MSOS monitor requests may directly specify any legal system logical unit.† However, requests to a device other than mass memory or the user terminal must be made only from the master terminal unless the device is a work station unit. The logical unit of the user's terminal may be obtained by a call to PGMIN.

Read-type requests may be made from any user program to a mounted or unmounted mass memory volume. Write requests to unmounted volumes can be made only from the master terminal.

Write requests may be made to the MSOS comment device (the master terminal).

In summary:

Logical Unit	Read Operation	Write Operation
4	Illegal	Master terminal
Communications line adapter controller††	Users terminal	Users terminal
Mounted mass memory	Specified unit	Illegal
Unmounted mass memory†††	Specified unit	Specified unit
All other logical units†††	Specified unit	Specified unit
Associated work station unit	Specified unit	Specified unit

Terminal operators may specify the system device or file to be used as a substitute for terminal input, data output, or both. This capability is described fully in section 3. This same type of specification can be made within a user program by using one of the subroutines described below. In addition, a user program can relate a device name to an MSOS logical unit number within the system.

The specifications are in the form of FORTRAN-compatible subroutines.

The calling sequence is:

```
CALL INPEQ (name,status)

CALL OUTEQ (name,status)

CALL LUNEQ (name,logunt)
```

Where: name is a four-word array containing an ASCII file name or system peripheral name.

status is the completion status:

Positive - Accepted

Negative- Rejected

logunt is the system logical unit specified by name (-1 indicates an invalid name).

CURSOR POSITIONING

User programs are provided with two methods of controlling the position of the cursor on the display screen.

- Cursor position codes may be embedded within the message text of output messages. These are transformed by the system into horizontal and vertical cursor position. The format of the embedded cursor position code is:

```
(ESC) 1
xx yy
```

Where: xx refers to the horizontal cursor position (0 through 79).

yy refers to the vertical cursor position (0 through 23).

The hexadecimal equivalent of the escape sequence used is 1B 31 xx yy

- Using the write-then-read request, the program may specify the position of the cursor at the time the screen is read. The write-then-read request is discussed in the following section.

USER PROGRAM INPUT/OUTPUT REQUESTS

User programs that run under the ITOS executive may control data transmission by means of three classes of requests: standard MSOS input/output requests (READ, WRITE, FREAD, FWRITE, and MOTION), described in the MSOS reference manual; file manager requests described in the File Manager reference manual; and special ITOS requests described in this section.

WRITE-READ REQUEST

The use of the write-then-read request eliminates the extra delay that can occur when other user programs execute between a terminal write request and an immediately following read request. The write-then-read request may also be used to position on the CRT screen cursor in combination with terminal writes or reads, again without an intervening delay.

† All assembler, FORTRAN, and RPG II programs must use actual logical units, so that there is no conflict with the batch output driver.

†† As obtained from the PGMIN call

††† Can be specified only from the master terminal

In addition, a termination code is returned to the requesting program. This code may be used to establish the cause of input completion (for example, line feed, rubout, and so forth).

A single MSOS logical unit is used to specify the device on which to write as well as the device from which to read. Input and output may be characters or words, depending on bit 12 of the logical unit word: 0 equals character mode and 1 equals word mode.

The following rules apply with regard to the use of write-then-read when an INPUT= or OUTPUT= command is evoked.

- If the input portion of a write-then-read request specifies a device other than a user terminal, the output portion of the request is ignored.
- If the input device specifies the terminal but a separate output device has been specified, the terminal is used for all write-then-read outputs.

The write-then-read request is in the form of a FORTRAN-compatible subroutine. The requesting program does not regain control until the read portion of the request is complete.

The calling sequence is:

CALL WTREAD (lu, xy1, obuf, n1, xy2, ibuf, n2, tc)

Where:
- lu is the MSOS logical unit.
- xy1 is the cursor position prior to output.
- obuf is the output buffer address.
- n1 is the number of words/characters to be output.
- xy2 is the cursor position prior to input.
- ibuf is the input buffer address.
- n2 is the number of words/characters to be input.
- tc is the returned value of the termination code.

All variables are integers, and all parameters must be absolute. A value of -1 indicates a null parameter for xy1 and xy2. If n1 is zero, no message text is output and xy1 is ignored. If n2 is zero, completion occurs immediately after the output and xy2 is ignored.

TERMINAL MANAGER REQUEST

The ITOS terminal manager provides an interface between an RPG II applications program and the ITOS system for terminal input/output. The terminal manager is called by the application by means of an EXIT request and communicates with the ITOS system by using the ITOS write-then-read request. If requested, the terminal manager checks terminal input data for validity.

A description of the EXIT request from RPG II is contained in the RPG II reference manual. This request translates into a FORTRAN subroutine call with the following format:

CALL SUBRCM (buffer,t,bl,r,m,d)

Where: buffer is the variable in the program that is used to pass information to and from the terminal manager.

t is a one-byte code used to specify the type of function the terminal manager is to perform:

- A Accept bl characters from the screen buffer. The operator enters the character on the keyboard, and they are echoed on the screen.
- B Set the buffer length to blank filled on type A or type N requests.
- C Clear the screen and position the cursor at (0,0).
- F Terminate the job.
- N Accept blngth characters from the keyboard. The characters are not echoed on the screen.
- P Position the cursor at the XY coordinates specified in buffer words 0 and 1.
- R Display a 50-character message; then accept the response of C (terminate job). A response of E or R is returned in r.
- S Display the message in the buffer; position the cursor at the first character past the end of the message.
- T Display the message in the buffer; position the cursor at the start of the following line.

bl is the buffer length, the number of characters to be accepted or typed. Must be greater than 0 and less than 80.

r is the return code, a variable used to report to the calling program the method of termination for A-type requests or the action taken for R-type requests.

m is the mask, a code used to determine what action can be taken by the operator when an R request is used; may be from 1 to 7.

d indicates the method used to determine the type of data expected on A or N requests.

A description of each request type follows:

- Request types S and T – bl, the number of characters in the buffer, is printed on the CRT screen. If the type is S, the cursor is positioned at bl +1 on the same line; if the type is T, the cursor is positioned at the first position of the next line.

- Request type A – bl number of characters can be entered through the CRT keyboard and are echoed on the screen. When the keyboard entry is terminated, the characters are transferred into the buffer in the program with trailing blanks. There are five methods of terminating a keyboard entry. The type of termination is returned to the caller in the return code.

Return Code	Method of Termination
C	Carriage return. The cursor goes to the first position of the next line.
E	The end of the line was reached (character position 80). The cursor goes to the first position of the next line.
L	Line feed. The cursor is positioned one character beyond the last character typed.
R	Rubout. The cursor is positioned one character beyond the last character typed.
O	Overflow. bl number of characters were entered, terminated by a carriage return. The cursor is positioned one character beyond the last character typed.

If d, the data type, is nonzero, the data entered by the terminal operator is checked for validity before being passed to the caller. If the data is not valid, the terminal bell is sounded and the cursor is repositioned at the first character of the field ready for further input.

Data Type	Checking Performed
0	None
1	The field must contain only numeric characters (0 through 9).
2	The field must be nonblank.
3	The field must contain a valid date. A valid date is defined as follows: -Exactly six characters were input. -All input characters were numeric. -The month is from 1 to 12. -The day is from 1 to 31.

- Request type R – 50 characters from the buffer are output to the screen. Up to three responses can be accepted depending on the value of the mask.

Mask Value	Possible Valid Responses
1	C
2	E
3	E or C
4	R
5	R or C
6	R or E
7	R or E or C

If the user enters an invalid response, the terminal bell is sounded, and a new response is requested. If the response is a C, the terminal manager terminates the job. If an E or an R is entered, the response is returned to the caller in r.

- Request type P – The cursor is positioned at the X and Y coordinate specified by the first four bytes in the buffer. The X coordinate (the first two bytes in the buffer) must be between 00 and 79. The Y coordinate (the third and fourth bytes in the buffer) must be between 00 and 23.

- Request type C – The display is cleared and the cursor is positioned at (0,0).

- Request type F – The user application job is terminated.

- Request type N – The result is the same as described for type A requests above, except that the data input at the keyboard is not echoed on the display (the display remains unaltered).

- Request type B – When the terminal manager is called with type B, the blngth parameter specifies the length of the buffer that is filled with blanks when processing type A or N requests. Any type A or N requests processed prior to a type B request clear only the buffer size specified in the type A or N request. If a type B request is received with lbngth equal to 0, this cancels the effect of a previous type B request.

The terminal manager terminates the user application program in a number of cases when an abnormal condition is detected. One of the following error messages is displayed at the terminal specifying the reason for termination.

SUBRCM CALL ERROR– ILLEGAL REQUEST TYPE

SUBRCM CALL ERROR– X COORDINATE LESS THAN ZERO

SUBRCM CALL ERROR– X COORDINATE GREATER THAN 79

SUBRCM CALL ERROR– Y COORDINATE LESS THAN ZERO

SUBRCM CALL ERROR– Y COORDINATE GREATER THAN 23

SUBRCM CALL ERROR– BUFFER LENGTH LESS THAN 1

SUBRCM CALL ERROR– BUFFER LENGTH GREATER THAN 80

SUBRCM CALL ERROR– ILLEGAL 'DTYPE'

TERMINAL I/O DRIVER INTERFACE

The ITOS executive provides an interface between the user program and the system communications terminal I/O driver.

All non-MSOS features of this interface are invisible to the user program, since the ITOS executive performs all reformatting prior to making the request.

The executive operates all terminals in an unsolicited input mode. This mode of operation requires that the executive establish a logical connection with the communications terminals prior to any driver requests. Once this connection is made, all input received on the port is placed in the specified input buffer, and the specified completion address scheduled by the driver when the input termination character is entered. The ITOS executive does not perform READ or FREAD requests. When this type of request is received by the protect processor from a user program, it is translated into a WRITE OR FWRITE request with no completion address specified. Following the entry of data by the terminal user, the unsolicited input location is scheduled by the driver, which forms the completion of the user program READ request.

The MSOS monitor requests specified by the executive is standard, except for the following:

- Request mode bit A specifies the data element of the request. All requests are ASCII, either character- or word-oriented:

  A = 0 indicates that the request length specifies characters.

  A = 1 indicates that the request length specifies words.

- All communications terminals are defined as a single logical unit in MSOS. The individual communications line is specified by a logical port number contained in the message header.

- The Q register contains the V field and the logical port number at the completion of an I/O request.

- 'S' in the request points to a five word header. The format of the five-word header is:

Word 0	Bits 15-8:	Reserved for future use
	Bits 7-0:	Contains the logical port number (1 through 255) of the communications line to be used. Set by the ITOS executive on WRITE or FWRITE. Set by the driver on unsolicited input.
Word 1	Bits 15-5:	Contain the status set by the driver only if bit 15 of the request V field is equal to one.

Bit	Meaning
15	Communications subsystem down
14	Spare
13	Spare
12	Lost data
11	Framing error
10	Request timeout
9	Illegal request
8	Parity error
7	Spare
6	Spare
5	Spare

Bits 4-0: Contains the subrequest code:

0	Normal mode
1	Logical connect
2	Logical disconnect
3	Write-read operation
4-31	Reserved for further use

Word 2	Bits 15-8:	Cursor X position prior to data input.
	Bits 7-0:	Cursor Y position prior to data input. Set by the executive on write-then-read operations. The driver returns the termination code in word 2 on unsolicited input.
Word 3	Bits 15-0:	Contains the number of characters requested. Set by the executive for read operations. The driver returns the actual number of entered characters upon read completion.
Word 4	Bits 15-0:	Contains the address of the I/O message buffer.

USER APPLICATION PROGRAMS

User application programs may be written in FORTRAN, RPG II, or macro assembler language. The object code for each program must be installed in the MSOS program library as a file absolutized at memory partition 2, via the *P,F,2 and *N,NAME,,,B procedures of the MSOS library editing (LIBEDT) utility. User programs must be written using absolute addressing conventions, and the D bit is required in all MSOS monitor requests. No other special requirements are imposed on the user program.

Data transmission is limited to file manager requests and MSOS read and write operations. The only other legal MSOS requests are MOTION, STATUS, and EXIT.

Programs can be exited in any of three ways:

- By an MSOS EXIT request
- By a call to PGMOUT, described below
- By an exit to the MSOS dispatcher when no I/O completion is pending

User program initiation is performed by the executive log-on processor. This processor is executed when a terminal is activated or when a user program terminates.

Five optional subroutines are available to the user program. In the following descriptions, all ASCII parameters are assumed to be left-justified.

PROGRAM INFORMATION

The PGMIN subroutine supplies information from the ITOS executive concerning the execution environment of the user program. The calling sequence is:

CALL PGMIN (iduser,lunit,mode,noport)

Where: iduser is a four-word array in which the ASCII user identification is passed.

lunit is a location in which the MSOS logical unit of the user terminal is passed.

mode is the location in which the current mode of operation is passed:

0	Interactive
Nonzero	Procedure stream †

noport is a location in which the user terminal port number is passed.

PROGRAM INTERRUPT

The program interrupt (PGMINT) subroutine specifies the action taken by the executive if a program interrupt is entered from the terminal keyboard. The program interrupt is defined to be CONTROL D.

The calling sequence is:

CALL PGMINT (iaddr,iflag)

Where: iaddr is the location to which the executive passes control if a program interrupt is entered.

iflag is a location that is set nonzero by the executive if iaddr is specified as zero.

PROGRAM MESSAGE PROCESSOR

SYSMSG is a general message processing routine that utilizes the system message file for the message skeleton and allows program supplied variables to be included.

The calling sequence is:

CALL SYSMSG (index,idata)

Where: index is the message index. This corresponds to a relative record number in the message file.

idata is an array that may contain program-supplied data to be included in the message. idata may be specified as zero if no data is required.

Message file records may contain three special function characters:

@ specifies that the next ASCII character from idata is to be placed in the message text.

specifies that the next two words of idata are to be converted to decimal ASCII and placed in the message text. The least significant digits are placed in the text, based on the number of # characters specified.

$ specifies that the next word of idata is to be converted to hexadecimal ASCII and placed in the message text. Data is placed right-justified as in the decimal specification.

PROGRAM EXIT

The PGMOUT subroutine returns control to the ITOS executive at the end of user program execution. There is no return from this subroutine.

The calling sequence is:

CALL PGMOUT

PROGRAM INITIATION

The CHAIN subroutine initiates the execution of another program, function, or procedure stream. No parameters are passed to the requested program, so that any necessary information must be supplied in a file accessible to both calling and called programs.

The calling sequence is:

CALL CHAIN (name)

Where: name is a four-word array that contains the ASCII name of the user program, user function, or user procedure file.

DATA STRUCTURE

The ITOS system requires a data structure that is located in MSOS SYSDAT and in a protected area within the user program's execution space.

- A general data section in SYSDAT contains all data and tables that are not directly related to individual user programs.
- An input/output data section in SYSDAT contains all data and tables required for user program I/O and file request processing.
- A user data section in SYSDAT contains data and tables relating to active user and multiuser programs.
- The user linkage buffer is located immediately adjacent to the user's execution space. The buffer is 256 words long and contains all data and tables that must remain with the user program when it is swapped to mass storage.

† This nonzero value is the address of a location that must be set to nonzero to terminate procedure stream mode if an error is detected by the user program.

ITOS uses File Manager Version 2 to process all files. The file manager has two principal types of files: sequential and indexed. ITOS has several additional subcategories of files defined to satisfy specific system uses. These include:

- Private (user) files
- Common files
- System files

PRIVATE FILES

Private user files consist of the set of files within the system that contains nonblank owner names. These may be created by the user either internally within a user program by means of the file manager CREATE request or by use of the file utility command DEFINE. In either case, the file owner name consists of the name the operator used to log onto the system.

COMMON FILES

Common files may be created by the same methods described above for private files; the difference is that the operator logged onto the system using a blank response to the USER ID request. In addition, the applications system supplied by Control Data contains numerous common files.

It should be noted that any user can alter, use, or delete any common file if he knows the file name and logs onto the terminal with a blank ID.

SYSTEM FILES

A set of private files within the system is referred to as the system files. This set has an owner name $$xxxxxx and is used by the ITOS executive during user program execution.

System files can be either indexed or sequential. They are distinguished by the starting characters $$ and can be reached for maintenance purposes by an operator who logs onto the master terminal with the user ID of $$. The ITOS executive reserves these files for system uses.

The first seven files named below are always part of ITOS; the remaining files (used to select tasks to be performed) depend on the system configuration.

- $$PGMNAM – Directory to program library; indexed by program name
- $$SWPBUF – Buffer for swapping user programs
- $$DAYFILE – Used to record system usage
- $$USERID – Valid user identification directory
- $$SYMSGF – System message
- $$PROCED – Procedure file directory; indexed by procedure name
- $$SYMENU – System menu contains a list of the functions available for execution at an ITOS terminal
- $$xxMENU – Application module function menu. Up to 13 application menus may be included in a system, depending on the applications purchased. Each of these applications presents the user with a choice of specific procedures to be executed, as follows:

$$OEMENU	Order entry/invoicing/sales analysis
$$OFMENU	Order entry file maintenance
$$ARMENU	Accounts receivable
$$INMENU	Inventory control
$$APMENU	Accounts payable
$$PRMENU	Payroll
$$GLMENU	General ledger
$$BMMENU	Bill of materials processing
$$MRMENU	Material requirements planning
$$POMENU	Purchase order processing
$$RTMENU	Routing
$$WRMENU	Work in process
$$PIMENU	Physical inventory

- $$MOUNTS – Mount utility tape directory file. This file is not included in systems without magnetic tapes.
- $$HOST – Directory for local and remote host batch processing
- $$BATCH – Local and remote batch job queue; indexed by job number
- $$PRINT – Deferred batch print queue; indexed by print file name

PROGRAM NAME DIRECTORY

The program name file, $$PGMNAM, is an indexed file that relates the program name to its location and length in the MSOS program library. The record format is:

Word Position	Field
0, 1, 2	Program name in ASCII. This is the record key.

Word Position	Field
3	Program sector most significant bit with control point indicator (bit 15)
4	Program sector least significant bit
5	Program length (sectors)

This file is rebuilt and re-indexed each time ITOS is activated via the start operation. This process consists of searching the MSOS program library directory for all library files and adding an entry in $$PGMNAM for each one found. This process requires approximately 30 seconds and ensures rapid access of requested user programs.

If programs are added to the MSOS program library while the system is active (as in a concurrent batch system), an entry does not exist in the directory. However, the ITOS executive searches the MSOS program library directory for a requested program if an entry cannot be found in $$PGMNAM.

SWAPPING BUFFER

$$SWPBUF is a sequential file that provides buffers for swapping and unswapping user programs. Records are assigned to be one sector long (192 bytes), and allocation is managed internally by the ITOS executive. The size of this file is based on the number of user terminals in the system.

ITOS USAGE

$$DAYFIL is a sequential file that records system activity. The record format is:

Word Position	Field	
0	Eight-character (four-word) ASCII code of user ID	
1		
2		
3		
4	Month	Time when the operator logged onto the system (each field has two integers in decimal format).
5	Day	
6	Year	
7	Hour	
8	Minute	
9	Second	
10	Runtime in tenths of minutes (integer)	
11	Terminal number (integer)	
12	Reserved	

The dayfile contains 1250 records and is not updated by the ITOS executive once it is full.

This file may be processed by a user-supplied accounting program if desired.

TERMINAL USER FILE

$$USERID is a sequential file that relates the terminal number to a valid user identification. Each record contains a terminal port number, 0 through 16, with 0 assigned to the master terminal. The record format is:

Character Position	Field
1-2	Terminal port number in decimal ASCII
3-9	Not used
10-19	User identification in the form (aaaaaaaa); right blank fill within the ( ) delimiters
20-29	
30-39	
40-49	
50-59	
60-69	
70-72	Not used
73-80	Record sequence

Multiple records may be used for the same terminal port number, but records must be arranged within the file in ascending port number order. Since this is an 80-character sequential file, it may be modified by the ITOS text editor.

SYSTEM MESSAGE FILE

$$SYMSGF is a sequential file that contains system messages and is used by the system message processor SYSMSG. Each message is contained in a single 80-byte record. The format of the record is:

Character Position	Field
1-72	Message text
73-80	Record sequence

PROCEDURE DIRECTORY

$$PROCED is an indexed file that contains one record for each ITOS procedure. Each procedure name has an associated file/device name that holds the actual procedure. The record format is:

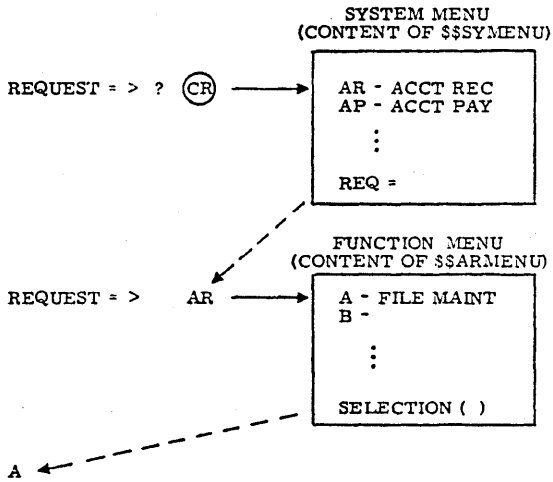
Character Position	Field
1-8	Procedure name, up to eight characters; this is the key for this file.

Character Position	Field
9-10	Reserved
11-18	Procedure file name or device name, up to eight characters, containing the procedure.

A typical procedure directory file format is:

Key	File Name
ARMFUPD	PRFAR001
⋮	⋮

Figure 5-1 shows an example of ITOS menu selection from the request through the location of the file in the procedure directory.



NOTE: ARMFUPD IS THE KEY INTO THE PROCEDURE DIRECTORY FILE.

Figure 5-1. ITOS Menu Selection

SYSTEM MENU

The system menu file $$SYMENU is a master directory for each of the standard application products. Each product has a single 80-byte record in the system menu file. The format of the record is:

Character Position	Field
1-8	Program name of the application product executive or ITOS utility
9-10	Not used
11-12	Selection mnemonic
13-72	Selection description
73-80	Record sequence

When the system menu is output in response to the operator entry of ?, only positions 11 through 72 are actually output. The operator entry is compared to the selection mnemonic, and if a match is found, the program name in positions 1 through 8 is initiated. This file may be modified via the ITOS text editor.

STANDARD APPLICATION PRODUCT MENU

Each of the standard application product files ($$xxMENU) is a directory to the procedures that can be executed as a part of that product. Each directory record is 80 bytes long and has the following format:

Character Position	Field
1-8	Procedure name, up to 8 characters
9-10	Not used
11	Selection mnemonic
12-72	Selection description
73-80	Record sequence

The product menu is processed by an ITOS executive program named MNUPRO. The two-character selection mnemonic that is used to select the proper product menu is passed to FUNSEL. Application procedures are then initiated, based on the entry of a single character selection.

This file may be modified via the ITOS text editor.

TAPE UTILITY DIRECTORY

$$MOUNTS in an indexed-linked file that contains information regarding magnetic tapes referenced by RPG II programs. The primary key is the record number, the secondary key is the tape file name.

The format of the record is:

Word Position	Field
0	Record number (primary key)
1-4	Eight-character (four-word) ASCII code of tape file name (secondary key)
5	Tape file indicators
6-8	File expiration date (ASCII)
9-17	Data set name (ASCII)
18-20	Volume serial number (ASCII)

LOCAL AND REMOTE HOST DIRECTORY

$$HOST is a sequential file that contains information regarding the local and all remote batch hosts. The first record of the file always specifies the MSOS batch processor and is identified by the name LOCL. The remaining seven records are reserved for use by COMM 18 Version 2.

The record format is:

Word Position	Field
0-1	Four-character (two-word) ASCII code of the host name
2	Host identification indicators
3-17	Job status bytes, indicating the status of every job associated with the host. A maximum of 60 jobs may be active.

LOCAL AND REMOTE JOB QUEUE

$$BATCH is an indexed file that contains a queue of all active jobs for the local and all remote host processors. The key is the job number as assigned by the BATCH utility processor. Sixty entries are reserved for local batch, and 60 entries are also reserved for each of the remaining seven host processors.

The record format is:

Word Position	Field
0-1	Four-character (two-word) ASCII code of the job number (J001-J760). This is the record key.
2-13	24-character (12-word) ASCII code of the volume-owner-filename string for the job input file
14-19	12-character (six-word) ASCII code of the date/time that the job was requested
20-25	12-character (six-word) ASCII code of the date/time that the job was transmitted
26-31	12-character (six-word) ASCII code of the date/time that the job was received

DEFERRED BATCH PRINT QUEUE

$$PRINT is an indexed file that contains a queue of all print output from the local and all remote host processors. The key is the print file name (PR01-PR50).

The record format is:

Word Position	Field
0-1	Four-character (two-word) ASCII code of the print file name (PR01-PR50). This is the record key.
2	Reserved
3-6	Eight-character (four-word) ASCII code of the print file volume name
7	Reserved
8-13	12-character (six-word) ASCII code of the date/time of print submission
14	Reserved
15-16	Four-character (two-word) ASCII code of the print file size (hexadecimal records)

SYSTEM FILE MAINTENANCE

Modification of the ITOS system files is not required to use the standard applications products as released by Control Data. This section contains the information necessary to modify these files, should a particular installation require it.

INITIAL SYSTEM FILE DATA

When the system is received from Control Data, initial data for the following files is contained in the MSOS program library:

- $$SYMSGF
- $$SYMENU
- $$xxMENU; based on the applications system
- $$PROCED; based on the applications system
- Procedure files; based on the applications system

When the initial start operation is performed, each of these files is created and loaded from the data contained in the program library.

Each segment of initial data is specified as an MSOS program library file, loaded under the A option. The name of the program library file relates to the ITOS file name by using the six least significant characters of the ITOS file name to specify the MSOS program library file name. For example, data for the ITOS file $$SYMSGF is loaded under the MSOS name SYMSGF.

The following MSOS batch processor control statements are used to load the initial data for the system message file and the system menu file:

```
*JOB
*LIBEDT
*N,SYMSGF,,,A
    message text records
*N,SYMENU,,,A
    menu text records
*U
```

Function menu files are automatically built by the start operation from the system menu file. This operation uses the two-character selection mnemonic in each system menu record to create the function menu file and to locate the initial data. For example, an AR selection in the system menu results in an ITOS file named $$ARMENU, which is initially loaded with data from the MSOS program library file named ARMENU.

Procedure files are also automatically built by the start operation from the procedure directory file. In this case, the file name is obtained directly from the directory record, but only the last six characters are used to locate the initial data. For example, a procedure directory entry:

```
ARMFUPD PRFAR001
```

results in an ITOS file with the name PRFAR001, which is loaded with data from the MSOS program library file FAR001. This file contains the necessary control statements for executing procedure ARMFUPD. If the MSOS name is not present, the ITOS file is not created.

The process of loading system files with initial data occurs only once, when the MSOS-ITOS system is first built.

PROCEDURE STREAM FILES

This section describes the methods used to set up a procedure file. Procedure files may contain the following types of records:

- Comments
- Device control statements
- Program names
- Program parameters

Procedures may not be included in procedures.

Comment records must contain a blank in the first character position, followed by an asterisk (*) in the second. When a comment record is encountered, the screen is cleared and the contents of the record (positions 1 through 72) are displayed. Several comments may be sequentially displayed following the screen clear.

Device control statements of the form

```
INPUT = name
OUTPUT = name
```

may also be included. The statement must begin in the first position of the record. The INPUT= statement is required if program parameter records are contained in the file, and the name specified must be the name of the procedure file. A new device control statement must precede each new program name if required for parameters.

Program names must begin in the first position of the record, and program parameters must conform to the conventions specified by the related program.

The following example illustrates a procedure file (PRFAR001) used to define and load a data file into the ITOS system and to execute a program:

```
 *DEFINE A FILE
INPUT=PRFAR001
UTIL
DEFINE,FILE,. . .
EXIT
 *FILE DEFINED
 *EXECUTE RPG PROGRAM AR005
AR005
 *RPG PROGRAM EXECUTED
 *EXECUTE RPG PROGRAM AR025
SWITCH=0100010
AR025
 *RPG SORT PROGRAM EXECUTED
 *SORT DATA
INPUT=PRFAR001
DSORT,. . .
    :
 *SORT EXECUTED
MNUPRO
```

When a procedure file is added to the system, an entry must be added to the procedure directory file. The LOAD utility (part of UTIL) can be used to add this record to the directory.

The ITOS file manager is a general-purpose file management system. It provides management for sequential and indexed files, as well as for a special type of sequential file called a direct file. The file manager is described in detail in the File Manager Version 2.0 Reference Manual.

A set of file management utilities (UTIL) is also provided. UTIL allows interactive maintenance of files. UTIL is described in detail in section 3.

The file manager maintains files on a direct access mass memory device only; it does not support magnetic tape files. However, the files may be dumped onto magnetic tape as a temporary storage device and re-entered onto a mass storage disk using the UTIL commands.

The file manager executive resides in main memory; it utilizes a reserved memory area for its mass memory resident request processors.

The file manager supports RPG II Version 2.0 and Sort/Merge Version 2.0. MSOS job processor files, pseudo tapes, and the MSOS background editor are not supported by this file manager.

FUNCTIONS

The file manager supports ITOS user programs under control of the ITOS executive. Users can create and maintain either sequential or indexed files. Records within sequential files can be retrieved sequentially or according to relative record number. Records within indexed files can be retrieved sequentially by relative record number or by an identifier (key) value. A given indexed file can have one to four keys.

The file manager provides for deleting records, updating records, clearing all records from a file, or deleting an entire file. In the last case, the file identification is removed from the file directory. A file or particular records within a file can be locked by a user to prevent concurrent use by another file user.

SEQUENTIAL FILES

In a sequential file, each new record is added immediately following the last record stored in the file. Records within a sequential file may be retrieved consecutively in the order in which they were stored. Alternatively, a particular record may be retrieved by specifying its relative position within the file (relative record number). The relative record number is used to position the sequential file; then records may be sequentially retrieved from that record position forward. Direct files, a special class of sequential files, can be created only by a DEFINE command from the UTIL utility. File manager treats these as any other sequential file; however, since all the records were filled with blanks at the time of file creation, the records are identified and added to the file by relative record number rather than sequentially. (To the file manager it appears that the record is being updated rather than being added for the first time.) Naturally, once a record has been initially added to any sequential file, that record can be accessed, cleared, changed, or marked as logically deleted by relative record number.

INDEXED FILES

In an indexed file, each record has at least one key (for instance, a surname, social security number, age, sex, or record number). The file manager allows each record to have up to four keys. Keys range in length from 1 to 29 bytes (8 to 232 bits). Each key value must be stored within its associated record. Primary keys must have unique values; multiple occurrences of key values are permitted in any of the other three keys. Keys may overlap. This is also true of the primary key so long as the primary key remains unique even though it contains all or part of duplicate secondary keys. An index is established for each key in a given file so that records may be retrieved in order according to the binary representation of the key value. Records with binary coded decimal (BCD) keys may be retrieved in numerical order and records with EBCDIC or ASCII keys may be retrieved in alphabetical order.

Each new record of an indexed file is added immediately following the last record stored in the file, as in a sequential file. The positioning of each such record in the indexed file is known by relative record number so that indexed files are actually indexed sequential files.

Records may be retrieved from an indexed file in one of the following ways:

- All records corresponding to a given key value may be retrieved.
- A set of records may be retrieved in order according to the numeric representation of key values.
- A specific record may be retrieved according to storage position within the file.
- A set of records may be retrieved in the order they were initially stored in the file.

FILE MANAGER SPACE

The space allocated to the file manager is defined at the time of system customization. It is usually not expanded or diminished; however, a change of space allocation can be accomplished offline by the methods described in the file manager reference manual.

Within the space allocated, the space that was formerly used by a deleted file is subsequently made available to the file manager for a new file. Space used by deleted records can be used for new records after the file is compressed or if the file is cleared. Mass memory space initially defined for file storage cannot be used for other system purposes.

FILE ORGANIZATION

The file manager consists of a main-memory-resident file manager executive, a set of main-memory-resident request processors, a set of mass-memory-resident request processors and support subroutines, and an interceptor routine that must be in main memory whenever a program makes a file manager request.

The files reside on mass memory together with information describing how to find information about a specified file (file definition directory) and how to find a record within a file (file control block table).

Indexed files require additional pointers. For a given key in a given file, these pointers are stored in conjunction with a pyramid structure as described in the file manager reference manual.

REQUESTS

Two types of file requests are provided: file specification requests and record access requests. Table 6-1 summarizes the file manager requests. Associated with each use of a particular file is a 24-word request buffer used to process the request. The same request buffer must be used for a sequence of requests referring to the same file. The buffer is used by the file manager to process requests, to save information between related requests, and to pass information back to the caller. The buffer may not be altered by the user between successive file manager calls. This buffer is normally located within the user's program. For an unprotected user program, the buffer must be in unprotected main memory.

A status indicator word is also associated with each request. Upon completion of the request, the status indicator word contains request execution status information.

FILE IDENTIFICATION

At the time of file creation, each file name is defined by three characteristics.

- A user-specified file name up to eight characters in length
- An owner name up to eight characters in length
- A volume name up to eight characters in length; this identifies the disk pack upon which the file is physically written. Each disk pack has a volume name label in its lowest numbered sector.

The file name is made up of a combination of the file/owner/volume names. All ITOS users are required to supply a user identification (user ID) when logging onto the system. The ITOS system executive replaces the file owner name with the user ID each time the owner name is required by the file manager.

An ITOS user may maintain a set of private files by logging onto the terminal with a private user ID. As an alternative, the user may access common files by logging onto the terminal with a blank user ID.

RECORD SAFEGUARDS

The file manager provides a technique for recovering records after a system failure. If the mass memory device fails during a transfer of records, however, these records cannot be recovered. An indication of the irrecoverable error is sent to the user when the system is again started and the file is reopened.

SPECIAL CHECKS

To indicate that a record has been deleted, the file manager stores a record-deleted code into the record. To denote the last record in a file, the file manager stores an end-of-file code in the beginning of the space where the next record is to be stored.

Periodically, the file manager checks each volume in current use to ascertain that the volume label is the one expected for that disk drive. The file manager laso automatically disables a volume when a mass memory I/O error is detected.

TABLE 6-1. SUMMARY OF FILE MANAGER REQUESTS

Command Mnemonic	Type of Parameters	Comments
FILE LEVEL COMMANDS		
CREATE	File/owner/volume name; definition parameters	Creates a new file. Rejected if there is no more file or file directory space, if the volume is not mounted/ready, or if the name is not unique
CLEAR	Name	Clears all records from the file but retains the assigned name. Rejected if the file is open, if the file cannot be found, or if the volume is not mounted/ready
DELETE	Name	Deletes all records and the file name itself. Rejection conditions are the same as for CLEAR
OPENFL	Name Number of records to retrieve and retrieval mode; locking flag	Opens the file for use; generates the REQBUF for a file. REQBUF holds all pertinent file processing information while the file is being used. Several programs can use the same open file. Rejected if too many files are open, if the file is locked or cannot be found, or if volume is not mounted/ready

TABLE 6-1. SUMMARY OF FILE MANAGER REQUESTS (Contd)

Command Mnemonic	Type of Parameters	Comments
FILE LEVEL COMMANDS (Contd)		
CLOSFL	REQBUG (see OPENFL)	Closes a previously opened file
LOKFIL	REQBUF (see OPENFL)	Prevents another user from using this file. Rejected if the file was opened for record locking, if another user has the file open, or if REQBUF has errors
UNLFIL	REQBUF (see OPENFL)	Unlocks a locked file. Rejected if the file is already unlocked or if REQBUF has errors
GETFCB	REQBUF (see OPENFL); buffer to contain the file control block. Parameters defining which file control block is to be retrieved	Gets file control block; rejected if the index is invalid, if the volume is not mounted/ready, or if the definition parameters have errors
UPDFCB	Same as GETFCB	Updates the portion of the file control block that is not controlled by the file manager. Rejected if the definition parameters have errors
RENAME	Names, old and new	Renames the file without disturbing the location of records. Rejected if the file is open or cannot be found, if the new name is not unique, or if the volume is not mounted/ready
REDUCE	Name and new number of records	Reduces the number of records that the file is defined to contain. Rejected if the file is open, if the file is an indexed file, if the file cannot be found, or if the new number of records is greater than the current number of records or is less than the number of existing records.
RECORD LEVEL COMMANDS		
PUTS	REQBUF (see OPENFL); number of records to add. Buffer to hold the records to be added.	Adds records sequentially; rejected if the file overflows or if REQBUF has errors
WRITER	REQBUF (see OPENFL); buffer for holding record. Primary key of record	Adds records sequentially and updates key indexes. Rejected if the file overflows, if the primary key is not unique or is in error, or if REQBUF has errors
READR	REQBUF (see OPENFL); buffer for holding record, Key used to find the record or relative position of the first record	Indexed files: reads record by key value or reads records by record position starting from a specified position. Sequential files: reads records by record position. Rejected if the record is locked, if the lock table overflows, if the file end is encountered, or if REQBUF has errors
GETS	REQBUF (see OPENFL); buffer for holding records relative position of first record or read	Reads records sequentially, starting from a specified record. Rejection criteria is the same as for READR.
UPDREC	REQBUF (see OPENFL); buffer for holding record	Substitutes new record for existing record. Rejected if neither file nor record is locked or if REQBUF has errors
DELREC	REQBUF (see OPENFL); buffer for holding record	Marks a record as deleted. If the file is indexed, key indexes are modified to delete references to the deleted record. Rejected if neither file nor record is locked, if key indexes are not fully updated, or if REQBUF has errors

TABLE 6-1. SUMMARY OF FILE MANAGER REQUESTS (Contd)

Command Mnemonic	Type of Parameters	Comments
RECORD LEVEL COMMANDS (Contd)		
COMFIL	REQBUF (see OPENFL); buffer to be used for record I/O	Compresses files by deleting all records that are marked to be deleted. Rejected if REQBUF has errors or if compression has been completed. The request must be repeated until completion status is obtained.
VOLUME LEVEL COMMANDS		
VOLUSE		Enables loaded and ready volume for use by the file manager or disables the use of the currently enabled volume so that it may be dismounted.

The remote batch subsystem included in ITOS allows deferred local batch submission and processing as a standard feature. It can also support job submission and processing by remote host processors if the COMM 18 Version 2 product is included in the system.

The batch utility section of this manual contains a description of the methods which may be used by the terminal operator to submit batch jobs. Local batch jobs are automatically provided with the necessary MSOS job control statements, but all remote host job control language must be supplied by the user. In addition, an ASCII record containing *JOB in column 1 must be included somewhere in the remote host input stream and in the print stream that returns from the host processor. It is the responsibility of the user to determine the mechanisms that allows this to occur.

BATCH SUBMITTAL

This section contains a summary of the user procedures necessary to submit a local or remote batch job:

- The terminal user defines a text file by means of the UTIL function DEFINE.

  NOTE

  This file must not be sector-aligned.

- The user then enters data in the text file (including all remote host job control language and the *JOB record if the file is to be submitted remotely).

- The user submits the job for deferred batch processing by means of the UTIL function BATCH. This function assigns a job number to the file, adds this to the ASCII *JOB record, and then adds an entry to the batch queue file ($$BATCH). The text file is then moved to the job file that corresponds to the assigned job number. Finally, the utility displays the job number on the user terminal.

- The terminal user may obtain a status of the job at any time via the UTIL function BATS, using host or job number. The batch status indicates the current position of the job in the batch stream (for example, not sent, being sent, output received, and so forth). The user may remove the job from the batch stream by using the UTIL function DISCARD; however, removal does not occur until the job output is received.

- Once the batch status indicates that the user's job is complete, it may be processed by means of the UTIL function DISPOS. There are three options involved with DISPOS:

  -The user may submit the output for printing on the system device.

  -The user may copy the output to another file.

  -The user may specify that both of the above operations are to be performed.

DEFERRED BATCH CONTROL

Both remote and local batch control is performed at the master terminal by the system computer operator. When used as an ITOS user terminal, the master terminal may also be used for batch submittal. The following operations may be used by the system operator to control deferred batch:

- The operator may obtain a master summary of all batch jobs for all hosts.

- The operator may set up local or remote batch processing at any time, based on system resources and the contents of the batch queue for the various hosts.

- For local batch, the operator performs a *BATCH,F. Once this has been initiated, local batch continues automatically until disabled via the UTIL function SET,LOCL,0.

- If the system contains COMM 18 Version 2, remote batch is initiated by associating the batch input to the desired host via the UTIL function SET. The operator then dials up the proper host line and continues operation.

- The computer operator may periodically status the $$HOST file and, if there are print requests pending, may output them to the system line printer by means of the UTIL function PRINT.

- The files specified in the $$HOST and $$BATCH files may be purged at any time via the UTIL function FLUSH.

BATCH INPUT DRIVER

The batch input driver reads input from file manager files for jobs submitted for deferred batch processing from an ITOS user terminal.

INTERFACE DRIVER REQUESTS

Unless the request is a motion request, each request is processed as an MSOS FREAD request. The driver picks up the next queued job in the $$HOST file that is not sent. The job status is changed to being sent and that job's batch file is retrieved. The $$BATCH file is updated with the date and time of day transmitted. Records are retrieved one record at a time from the job text file and returned to the requestor until all job records are processed.

MOTION REQUESTS

If the request is a skip file forward, the pointers in the $$HOST file are adjusted to point to the next queued job, and the current job is terminated. If the motion request is a backspace file, the $$HOST file pointers are adjusted to the current job, and the current job is restarted.

STATUS

When the driver detects an end-of-file in the text file, an EOF condition is returned to the requestor and the job status in the $$HOST file is updated to SENT. When all queued jobs for the host have been sent, the driver returns an end-of-batch indication to the requestor. For HASP and 200UT this is an end-of-file condition (0200_{16} for 200UT and $0F00_{16}$ for HASP), and for MSOS end-of-batch is ASCII *Z.

ERROR CONDITIONS

All errors arise from file manager error status and result in the job being terminated and marked inactive in the $$HOST file. The driver outputs the error message, specifying the job name, file manager error status, request type, file name, and user name.

BATCH OUTPUT DRIVER

The batch output driver routes the output from deferred batch processing to file manager files for subsequent listing or review at a print device or at an ITOS user terminal. All FORTRAN, assembly, and RPG II application programs must use actual logical units, so that there is no conflict with the batch output driver. If not, the job being batched disperses application output through the output. In addition, the batch output file may be closed before the job is completely finished.

INTERFACE DRIVER REQUESTS

Unless the request is a motion request, each request is handled as an MSOS FWRITE request. The driver creates a scratch file and searches the requestor's buffer for the character string, *JOB,Jmnn, where mnn are unique digits. The job ID, if any, is saved and the requestor's record and each subsequent record is put in the scratch file.

If the job is unidentified (if there is no *JOB,Jmnn character string), the $$PRINT file is accessed for an available entry and updated with the file volume name, the date and time of entry, and the number of records in the file. The scratch file is then renamed PRxx, where xx is based on the $$PRINT file entry.

MOTION REQUESTS

An EOF motion request indicates the end of the job for this logical unit. The number of records in the scratch file is then reduced to the actual number of records stored in it. If the job was identified, the status is updated in the $$HOST file. If the job status was discard pending, the scratch file is deleted. Otherwise, the $$BATCH file entry is updated and the scratch file is renamed Jmnn.

Backspace one file is the only other valid motion request. This causes the scratch file to be deleted.

ERROR CONDITIONS

All errors result from file manager error status. A diagnostic message is output to the console, as for the batch input driver, and exit is made to the alternate device handler.

AUTOMATIC BATCH

The automatic batch processor monitors the $$HOST file for jobs submitted for local deferred batch processing from an ITOS user terminal.

The processor is initiated by:

```
MI
*BATCH,F
```

Subsequent processing of the local $$HOST is handled automatically until stopped by setting the local $$HOST logical unit to zero with the SET utility.

ABORT – The process of terminating an ITOS program by operator intervention. Performed by the entry of CONTROL A

ABSOLUTE MODE – The computer mode that specifies program execution without the use of page memory registers

APPLICATION MODULE – A series of programs used to perform several related types of tasks. ITOS supports 12 standard application modules written in RPG II language.

ASCII – An eight-bit character code. The standard I/O code for ITOS terminals

ATTACH – The process of joining a root program and a multiuser program in main memory so that the two programs are executed as a unit

AUTOLOAD – The process of loading the MSOS/ITOS system into the CYBER 18 computer from the disk image of the system

AUTOMODE (Editor) – The process of automatically supplying line (record) numbers to new records being entered into a file by the text editor utility

BACKGROUND – For MSOS/ITOS, background is the lower priority mode in which user application programs and job processor programs are executed. Foreground mode is reserved for a few system processes (such as I/O operations) generated as a result of unsolicited operator entries from terminals or mass storage data transfers.

BACKSPACE TEXT – The ability to move the cursor to the left on the display when correcting text in the line

BATCH MODE – Under MSOS/ITOS, batch mode is used to process programs in the unprotected batch area of main memory. Batch jobs are executed under the job processor and can be called only from the master terminal.

BLANK ID – A user ID consisting of all blanks (that is, a carriage return response to USER ID = message). This response permits the operator to access common files.

BYTE – An eight-bit field. The CYBER 18 computer word contains two bytes.

CATALOG – An RPG II utility that places a new RPG II application program on the program library in a format that ITOS can execute. Catalog is available only in batch mode.

CATALOGING – The process of entering a program onto the program library and indexing the program in the program library directory

CHARACTER STRING – A sequence of characters. The text editor searches on delimited character strings.

COLLECTIVE SWAP – The process of swapping several user programs at once to gain a large enough main memory area to load one large program

COMMENT DEVICE – The user terminal where MSOS functions can be called and executed. Also called the master terminal when ITOS is active

COMMON FILES – Files that are available to anyone who logs onto the system with a blank user ID. Contrast with private files and system files

COMPILER – A program that translates source language (such as FORTRAN or RPG II) into machine language. ITOS provides a compiler for each source language.

COMPRESS FILE – The process of rewriting a file to remove all records in that file previously marked for deletion. The remaining records are rewritten sequentially.

CONTROL A – The program abort command (pressing the CONTROL and A keys simultaneously)

CONTROL D – The program interrupt command (pressing the CONTROL and D keys simultaneously)

CONTROL G – The MSOS manual interrupt command (pressing the CONTROL and G keys simultaneously)

CONNECT REQUEST – The ITOS request that brings a terminal on-line to ITOS

CONTROL POINT – The page memory file configuration of an ITOS user program

CORE QUEUE – The ITOS queue holding the programs awaiting execution

CPU – Central processing unit, the CYBER 18-10M or 18-20 computer

CR – Carriage return (key), sometimes called return key

CREATE FILE – The process of defining file parameters so that the file manager allocates space and includes the file in the file directory for that volume

CURSOR – The marker beneath the line on the display indicating the relative position in the data buffer currently being read or written

CYBER 18-10M and 18-20 – Two CDC computers, either of which will support an ITOS system

DATA MANAGER – A set of RPG II runtime programs with data management functions

DEFINE (file) – A utility command that creates a new file

DELIMITER – A character used before and after a character string (for example, *) to set the enclosed character string apart from the rest of the data in the record

DIRECT FILE – A type of sequential file created by the DEFINE command of UTIL. A direct file has 80-character records with all records filled with blanks at file creation. Records can be added to a direct file by relative record number rather than sequentially.

DIRECTORY – An index to programs, files, or other system information. ITOS has the following major directories: program library directory, program name directory, procedure directory, and file directories (one per volume).

DISPLAY – The CRT portion of the terminal

DRIVER – The program that transfers data between the CPU and I/O or pseudo I/O device. All drivers are a part of MSOS.

ECHOING – The process of displaying data on the screen as it is typed on the keyboard. Almost all information from the keyboard is echoed in the ITOS system (exceptions: PASSWORD and USER ID).

EDITOR – The text (line-by-line) editing utility for ITOS. Since each line is a complete record, this is a record editor.

EXIT – 1. The request that allows the operator to log off his terminal. 2. The request that ends an MSOS program.

EXTENDED MEMORY – Main memory in the CYBER 18-20 above 128K bytes. In ITOS, this memory is executable only in page mode.

FIFO – First-in/first-out; a method of processing queued requests

FILE – A collection of data. ITOS uses File Manager Version 2.0 to process file data. All files have one or more records. Each of a file's records contains data in the same format. A file cannot span two volumes. Files can be sequential or indexed.

FILE MANAGER – The set of programs that manages ITOS files

FILE MANAGER REQUEST – Specially formatted requests that cause the file manager to perform a file-related task

FILE MANAGER UTILITIES – The set of utility functions operating under UTIL

FILE NAME – The file identifier. It is derived from the file name plus the volume name plus the owner name.

FILE REQUEST BUFFER – A buffer of information associated with each file that is open for processing

FOREGROUND – For MSOS and ITOS, the foreground mode consists of those high-priority system tasks performed by MSOS or the ITOS executive (contrast with Background).

FORTRAN – A high-level programming language supported by ITOS

FREAD – The formatted MSOS read request

FREE FIELD – A format in which the next field begins immediately after the previous field's end. Field length is not predefined (contrast with fixed field format such as RPG II utilizes).

FWRITE – The formatted MSOS write request

IDLE – The condition in which the computer processes only the idle loop program while waiting for a task request to perform

INDEXED FILE – A file in which records can be accessed by one or more key words

INPUT BUFFER – A buffer designated to receive data from an I/O device

INTERACTIVE MODE – The ITOS terminal processing mode; it allows the operator to communicate with the programs by entering unsolicited requests and by supplying data in response to messages displayed at the terminal.

INITIALIZE – The process of placing system programs in the computer and readying the system for execution

I/O – Input/output

I/O DATA BUFFER – A buffer to hold data during a transfer with an I/O device such as a terminal, printer, magnetic tape transport, or disk

ITOS – The Interactive Terminal-Oriented System

ITOS EXECUTIVE – A group of programs that performs system executive functions (scheduling, queueing, translating requests) in the ITOS system. MSOS also has programs that perform system executive functions.

JOB PROCESSOR – The MSOS executive for batch-oriented operations

KEY – An index to a file. Indexed files may have up to four keys.

KEYBOARD – The set of keys and switches at an ITOS terminal (see appendix B)

KEYWORD – A key

LABEL – An identifier for disk packs (volumes)

LIBRARY – A collection of programs. MSOS has two principal libraries: the program library, containing user and job processor programs, and the system library, containing foreground programs

LIBRARY UNIT – The disk pack containing the libraries. It must always be mounted and ready. It is also called the system volume (SYSVOL).

LIFO – Last-in/first-out; a method for processing queued requests

LIST DEVICE – The peripheral unit used to receive listed outputs. It is usually a line printer for system outputs; it is the terminal display for terminal outputs.

LOGGING OFF – The process of taking a terminal offline from ITOS

LOGGING ON – The process of bringing a terminal online to ITOS

LOGICAL MEMORY – The mapped memory of ITOS. By use of paging registers, user programs may be executed in logical memory while residing in another area of physical memory. If the execution specifies a root/multiuser program, the two parts are not usually contiguous.

LOGICAL UNIT – An I/O or pseudo I/O device. Each logical unit has a number that identifies it to MSOS. The same physical I/O device may have more than one logical unit number.

LU – Logical unit

MACRO ASSEMBLER – The lowest level programming source language available to MSOS

MAIN MEMORY – The memory that is an integral part of the CYBER 18 computer. Maximum size of main memory for CYBER 18-10M is 128K bytes; maximum size of main memory for CYBER 18-20 is 256K bytes. Main memory is physical memory.

MAPPING – The process of using paging registers to map programs in physical main memory into logical memory

MASS MEMORY – The disk memory for ITOS. It consists of one to eight disk drives and the associated controller for CYBER 18-20 configurations; CYBER 18-10M configurations may use one to four cartridge disks.

MASS QUEUE – The ITOS queue holding the swapped or non-resident user programs awaiting further processing in main memory

MASTER TERMINAL – The terminal from which ITOS is started and stopped; the only terminal that can be used for batch processing. It is also the MSOS comment device.

MENU – A selection display from which the operator may choose any of several tasks listed on the display

MERGING – The process of building a new file, sorted according to selected key(s), from the records of two or more existing files

MI – Manual interrupt. Activated by CONTROL G under MSOS (accepted only at the master terminal)

MNEMONIC – An abbreviation or representation of a name or command

MODULE – A group of programs that performs a system task, such as one of the user application modules

MOUNT – The MOUNT command for disks under ITOS

MOUNTED AND READY – The disk condition in which the disk pack (volume) is ready for I/O transfers and is online to ITOS

MULTIUSER PROGRAM – Programs that can be used by several other programs. Multiuser programs are pseudo-re-entrant. They are attached to a root during execution.

NSWP QUEUE – The ITOS queue holding those programs in main memory that are eligible to be swapped

OUTPUT DEVICE – An I/O device that can accept data, such as a terminal display or printer

OWNER – The name of the user who created a private file, or who had the file created using his name

OWNER NAME – The name of the person owning a file. It is part of the file name.

PAGE REGISTER – Hardware registers in the CYBER 18 CPU used for main memory paging

PAGING – In main memory: the process of reserving a user program area in which programs are loaded into physical memory in blocks of 2048 contiguous words (one page). The page registers are then used to map the pages onto the region of logical memory where the absolute addresses of the program are set to be executed.

PASSWORD – A security measure available in ITOS that requireS that a user enter the secret password before he can bring a terminal online

PERIPHERAL DEVICE – Any I/O device attached to the CYBER 18 except a user terminal. Usual peripheral devices for ITOS are disk, card reader, printer, and magnetic tape.

PHYSICAL MEMORY – The complete main memory of the system's computer. See LOGICAL MEMORY.

PRIVATE FILES – Files created by a user who logged onto the system using a nonblank user ID. These files can be accessed only by an operator who uses the same user ID when he logs onto the system.

PROCEDURE DIRECTORY – An index to all ITOS procedure streams in this system

PROCEDURE STREAM – A group of related programs, commands, and parameters

PROGRAM – A related group of instructions and data areas that are run as a unit under ITOS

PROGRAM LIBRARY – The library of all background and ITOS programs. It is modified through LIBEDT utility under MSOS.

PROGRAM NAME DIRECTORY – The index to ITOS programs. The directory is held in the program name file ($$ PGMNAM).

PROMPTING CHARACTER – The character (>) that is displayed to remind the operator that he must enter data via the keyboard to continue the program processing

PROTECT BOUNDS REGISTER – The registers holding the first and last addresses of the user's unprotected area

PROTECT PROCESSOR – An element of the ITOS executive that checks access to protected main memory

PROTECTED MEMORY – That area of main memory that is flagged so that it can be accessed only by other protected instructions. This prevents inadvertent misuse of important system programs and parameters.

PURGE (FILE) – The process of removing outdated files from the system

QUEUE – A sequence of related types of requests threaded together throughout memory or threaded together in stacks. Entries in the queue contain information necessary to start the requests; position in the thread determines the order of performance. Entries in queues may be threaded on a FIFO, LIFO, FIFO within priority, or LIFO within priority basis. MSOS, ITOS, and drivers all use and maintain various queues that determine the order of processing tasks.

READ – The ITOS or MSOS unformatted request to read data from an I/O device

READ MODE – The terminal mode wherein the terminal is ready to accept operator entered data. All terminals other than the master terminal are always in read mode

(except during the time when data is being written onto the display). The master terminal is placed in read mode by depressing CONTROL and G.

RECORD – The smallest accessible data unit in a file. All files contain one or more records.

RECOVERY – The process of resuming processing following an error

RE-ENTRANT PROGRAM – A program that can be used by several other programs simultaneously. It is characterized by numerous entry points and modularized format. Multiuser programs are pseudo-re-entrant, in that they can be entered at any time when they allow themselves to be interrupted.

RELATIVE MODE – A method of loading programs so that all internal addresses are relative to the program's starting address. User programs must be written in absolute mode.

RELATIVE RECORD – The position of a record relative to the first record in the file

REQUEST PARAMETERS – The list of parameters accompanying a request call. The list specifies all values and addresses necessary to start the request.

REQUESTOR'S BUFFER – A data buffer supplied with the request, and used to hold data during the processing of the request

RESIDENT PROGRAM – A program normally stored on the specified medium. Main-memory-resident programs include most MSOS executive functions, many drivers, and much of the ITOS executive. Other programs are mass-memory-resident and are called into main memory only when needed to process a request.

ROOT – A portion of a user program. The other portion is one or more multiuser programs. The root must contain all the information and data that is unique to the program. It can read a multiuser into main memory, attach itself to the multiuser, detach itself from the multiuser, and be swapped.

ROUTINE – The smallest programming element in ITOS that can be called independently. Programs contain one or more routines.

RPG II – Report Program Generator II; source language and compiler. RPG II includes runtime programs that are in multiuser format under ITOS.

RUNTIME PROGRAM – For ITOS, programs associated with FORTRAN and RPG II that provide the instructions necessary to execute the object language output of the compiler

SCREEN OF DATA – One full data display. In extended data output, a screen consists of at least 23 data lines followed by the word PAUSE at the end of the last record.

SECTOR – 96 contiguous words on mass memory

SEQUENTIAL FILES – Files in which records are entered in sequence. Records may be retrieved in sequence or by relative record number.

SMD – Storage module disk. One of the types of mass memory used by ITOS. Up to eight disk packs (for a total of 400 million bytes) can be managed by the SMD.

SORTING – The process of sequencing records alphanumerically according to one or more keys supplied in the sorting request

SORT/MERGE – The ITOS sorting utility. It can be called as a part of a procedure stream.

SOURCE CODE – For ITOS, the coded statements as written by a programmer according to the syntax rules of RPG II, FORTRAN, or macro assembler language

SPACE ALLOCATION – The process of making a block of main memory available so that a program can be read into that space

SPECIFICATION FORMS – RPG II coding sheets that show the field length and locations for each of the eight input formats recognized by ITOS RPG II

START UP – The process of bringing ITOS online

STANDARD LOGICAL UNIT – The logical unit that performs a system task unless another logical unit is specified for this task. The list of standard logical units is given in appendix D.

STATE INDEX – An ITOS index indicating the current location (main memory or swapped) and stage of execution for active user programs

STATUS – The request (MSOS or ITOS) that generates the current status of an I/O device

STATUS WORD – Any word containing (in bit-by-bit format) a series of indicators for program or device status

SWAPPING – The process of moving one or more suspended programs from main memory to mass memory so that space is available to read other requested programs into main memory

SYSDAT – The MSOS and ITOS system data base

SYSTEM FILES – A set of files used by the ITOS executive. Files are identified by the name $$xxxxxx. They can be reached for maintenance purposes by a user who logs onto the master terminal with USER ID = $$.

SYSTEM VOLUME – The volume (disk pack), called SYSVOL, which contains all MSOS and ITOS programs. It must always be mounted and ready.

SYSVOL – The system volume

TAPE SIMULATOR FILES – A method of formatting mass storage to simulate a magnetic tape

TERMINAL – The I/O device used by an operator to communicate with the ITOS system. It consists of a display and a keyboard (see appendix B).

TERMINAL MANAGER – The ITOS module responsible for managing data flow between the computer and the terminal

TEXT EDITING – A capability, provided by the text editor utility, to add, alter, or delete text from records

TIMESLICE – The maximum period of execution time allocated to a program before that program must release the computer. If the timeslice is exceeded, the program is interrupted and then suspended so that other programs can be executed.

UNPROTECTED MAIN MEMORY – Any part of main memory that can be accessed by any program, protected or unprotected. User programs and batch programs run in unprotected memory.

UNSOLICITED INPUT – Any request entered at a terminal that is not in response to an MSOS read request

UNSWAP – The process of returning a swapped program to the location in main memory where it was placed prior to swapping

USER ID – The code entered by an operator in response to USER ID = at the time the operator logs onto the terminal. The user may use his own (secret) code to gain access to his private files. He may use a carriage return to gain access fo common files, or he may use $$ to gain access to system files.

USER PROGRAMS – Programs written to be executed in unprotected main memory under the ITOS executive. Applications modules are user programs.

USER TERMINAL – Synonymous with terminal

UTIL – The file management utility

VERIFY – For certain ITOS requests (such as STOP) a message (VERIFY) is presented to the operator. The operator must confirm that the request is to be performed before ITOS will perform it.

VOLUME – A disk pack that is labeled for file manager use

VOLUME LABEL – A sector at the start of a disk pack containing a unique ID (and other data) for the disk pack

VOLUME NAME – Mnemonic (up to eight characters in length) identifying the disk pack

WORD – The 16-bit basic unit of computer or mass memory storage

WRITE – The MSOS and ITOS unformatted write request

WTREAD – The ITOS write-then-read request used for terminal output

KEYBOARD

The keyboard format is shown in figure B-1.

The use of the special keys is discussed below. Other keys are used as normal typewriter keys are used.

MAIN KEYBOARD

CLEAR should never be needed since the ITOS programs clear the screen automatically before displaying the next screen of data.

BREAK, CO, and ETX are ignored by the system.

When logging onto the system, 96/64 and PAGE are initially pressed. They remain in the down position and should never be deactivated (pressed again).

The switches in the top row should be set to the following positions:

EVEN PARITY
FULL DUP
ON LINE
HIGH RATE

The use of RUBOUT and RESET is described in table 2-1 of this manual.

ESC is used only at the master terminal to protect the system. The key is used to respond to the protect request, which occurs during system autoload. The protect is accomplished by pressing ESC and typing J28@. No carriage return is needed.

NOTE

The ESC key should not be used at the master terminal at any other time.

Uppercase control symbols in rows 2, 3, and 4 are ignored by the system. All other symbols are legal.

The carriage return is the standard entry key; its use is described in table 2-1.

↑ moves the cursor up one line each time it is pressed. The cursor wraps around to the bottom line of the same display if it is on the top line of the display when ↑ is pressed.

LINE FEED normally moves the cursor down one line. Its use is described in table 2-1.

← is the backspace key. It moves the cursor back one space each time it is pressed. The character that the cursor previously marked remains unchanged. The cursor wraps around to the end of the previous line if ← is pressed when the cursor is in the first character position.

REPEAT causes any other character that is pressed at the same time to be repeated in successive character positions.

→ is the space forward key. It moves the cursor forward one position each time it is pressed. The character that the cursor previously marked remains unchanged. The cursor wraps around to the start of the following line if → is pressed when the cursor is in the last character position. The use of this key is described in section 3.

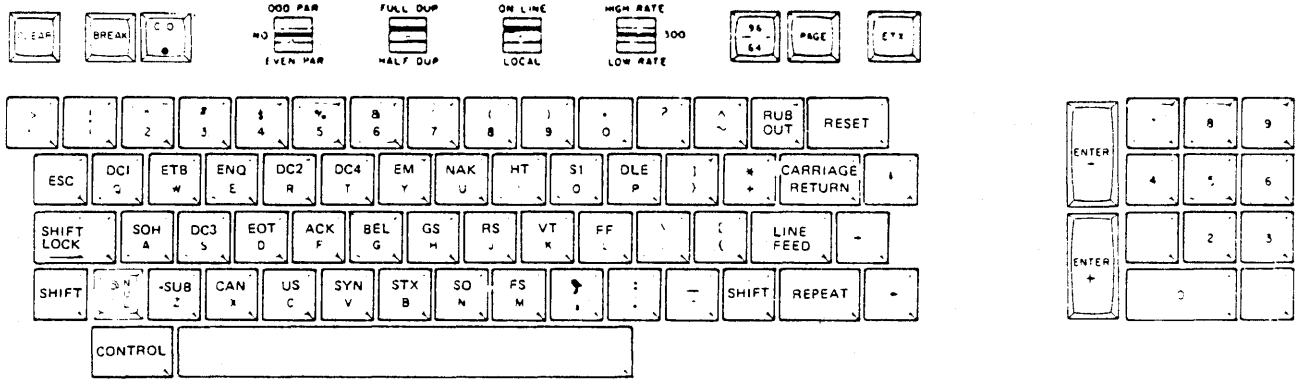


Figure B-1. ITOS Terminal Keyboard

CONTROL is used with one of three other keys. It must be pressed simultaneously with the other key being used:

1. CONTROL G places the terminal in read mode so MSOS instructions can be entered (master terminal only).
2. CONTROL A causes the running program to be aborted.
3. CONTROL D is a program interrupt for ITOS.

ARITHMETIC KEYBOARD

ENTER- is ignored by ITOS. The remaining keys (numerals and period) may be used interchangeably with the same keys on the main keyboard. ENTER+ is used as a normal terminator for fields with brackets [ ] for the applications programs.

ECHOING

All operator characters entered on the keyboard are echoed immediately on the screen except for replies to PASSWORD = > and USER ID = >.

In these cases, nothing is echoed except at the master terminal, where full echoing occurs. The cursor and prompting mark remain unaltered at other terminals despite the keyboard entry of characters.

CURSOR AND PROMPTING MARK

The cursor underlines the character on the screen that can currently be altered. If the prompting mark, >, is displayed, the cursor is positioned beneath this mark. The operator should type in the character for that cursor position, overwriting the prompting mark. For example,

REQUEST = >	The system requests another task.
REQUEST = UTIL (cr)	The operator enters the requested task, overwriting the prompting mark. Entry of a carriage return only does not move the cursor.

752 TERMINAL SETUP

The terminal must be set up as follows:

- Internal Switches

-Upper Set	Switches 2, 4, 5, and 7 are on. Switch 3 should also be on if the system is powered by a 50 Hz power source. All other switches are off.
-Lower Set	Switches 1 and 4 are on, switches 2 and 3 are off. Switches 5 through 8 are not used. †

NOTE

A switch is on if the lower side of the switch rocker arm is depressed.

- External Switches

The switches above the keyboard must be set as follows (from left to right):

EVEN PAR
FULL DUP
ON LINE
HIGH RATE

In addition, the 96/64 and PAGE keys must be depressed.

1843-2 1 × 8 CLA BOARD SETUP

First, orient the 1843-2 board in front of you as follows:

Component side up
Edge connector to the left

With the board oriented in this manner, the switch is off if the left side is depressed and on if the right (+) side is depressed.

Set all switches to their off positions, then set each of the switches marked + in the table below to on.

	1	2	3	4	5	6	7	8
S1		+		+				
S2	+		+					
S3								
S4								
S5								
S6	+							
S7	+							
S8-S15		+	+					

Note that if two 1843-2 boards are installed, the second board (terminals 9-16) must have S2 set as follows:

	1	2	3	4	5	6	7	8
S2		+	+					

NOTE

If the 1827-7 matrix printer is on the system, the switch from S8 - S15 that corresponds to the printer must be set as follows:

	1	2	3	4	5	6	7	8
Sx	+							

(x is found by adding 7 to the number of the CLA port attached to the printer)

NOTE

If a matrix printer is the standard list device in the system, it must be located on channel 8 of the first CLA board.

† For terminal 0 only, these switches must match the baud rate selected on the CPU's I/O-TTY board.

Figure C-1 shows the status of all system files on the system volume (SYSVOL). At the time status was taken, only system files were installed on this volume.

```
VOLUME:SYSVOL     DATE 05/17/77

AVAILABLE SPACE ON VOLUME    192604 SECTORS

FILENAME OWNER    FILEDATE FILE RECORD KEY1 KEY1 KEY2 KEY2 KEY3 KEY3 KEY4 KEY4 START RECORD EXPIRE   MAX.   STATUS
                           TYPE LENGTH LNG  POS  LNG  POS  LNG  POS  LNG  POS  SECT. COUNT  DATE   RECORD

$$PGMNAM $$       051777   I      12    6    1    0    0    0    0    0    0 103E0    356 999999    356 CLOSED
$$SWPBUF $$       051077   S     192    0    0    0    0    0    0    0    0 0740B      0 999999   3224 CLOSED
$$DAYFIL $$       051777   S      26    0    0    0    0    0    0    0    0 10174     81 999999   1000 CLOSED
$$USERID $$       051077   S      80    0    0    0    0    0    0    0    0 101FD      4 999999      4 CLOSED
$$SYMSGF $$       051077   S      80    0    0    0    0    0    0    0    0 10200    375 999999    375 CLOSED
$$PROCED $$       051077   I      18    8    1    0    0    0    0    0    0 1029E    208 999999    312 CLOSED
$$SYMENU $$       051077   S      80    0    0    0    0    0    0    0    0 1037E     16 999999     16 CLOSED
$$OEMENU $$       051077   S      80    0    0    0    0    0    0    0    0 1038F     17 999999     17 CLOSED
$$OFMENU $$       051077   S      80    0    0    0    0    0    0    0    0 10398     22 999999     22 CLOSED
$$ARMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103A3     15 999999     15 CLOSED
$$INMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103AB     18 999999     18 CLOSED
$$APMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103B4      1 999999      1 CLOSED
$$PRMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103B6     21 999999     21 CLOSED
$$GLMENU $$       051077   S      80    0    0    0    0    0    0    0    0 074CB     21 999999     30 CLOSED
$$BMMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103CA      1 999999      1 CLOSED
$$MRMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103CC      1 999999      1 CLOSED
$$POMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103CE      1 999999      1 CLOSED
$$RTMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103D0      1 999999      1 CLOSED
$$WPMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103D2      1 999999      1 CLOSED
$$PIMENU $$       051077   S      80    0    0    0    0    0    0    0    0 103D4      1 999999      1 CLOSED
```

Figure C-1. SYSVOL Status Printout

ITOS runs under the MSOS 5.0 operating system. This operating system contains a large variety of features and utilities. The following is a list of the MSOS features and utilities that are not included or supported under ITOS:

1. Job processor pseudo tapes and job files
2. Job processor text editor
3. COSY input driver
4. System (foreground) common

The following related MSOS products are not supported with ITOS:

1. Magnetic Tape Utility Processor (MTUP)
2. MSOS 5.0 RPG II Runtime (non-ITOS version)

Two types of messages are generated while ITOS is active:

- ITOS-related messages are delivered to the terminal where the connected program is requested. These messages are usually informational but occasionally request immediate action from the operator. UTIL and text editor messages are listed separately since these messages are likely to occur while the utility is being used in interactive mode. Sort messages are also listed separately.
- MSOS or source language related messages are delivered to the master terminal as a result of an MSOS problem or a problem related to a batch program. MSOS-related messages are listed in the MSOS 5 diagnostic handbook. Compiler/runtime/source language utility messages are described in the appropriate source language reference manual: RPG II Reference Manual, MS FORTRAN 3A/B Reference Manual, or Macro Assembler Reference Manual.

This appendix lists alphabetically, by logical grouping, all the ITOS messages (see tables E-1 through E-4). It should be noted that the file manager does not produce error messages as such except to save the hexadecimal representation of the status word. User programs are expected to check the status word (istat) to find the status of the requested file manager operation.

The section labelled ITOS Diagnostic Messages contains an index number. This is an index to that message in the system message file. If only the message file index is delivered to the screen because of an internal error, the operator may find the message text by use of this index number.

TABLE E-1. MIPRO ERROR MESSAGES (MASTER TERMINAL ONLY)

Message	Meaning/ Action
ITOS ACTIVE – REQUEST REJECTED	1. Attempted to start with ITOS already active. No action necessary. 2. Tried an INIT with ITOS active. Stop system and try INIT again.
BATCH PROCESSOR ACTIVE – REQUEST REJECTED	Tried to start ITOS in a system where MSOS background cannot be run concurrently with ITOS, and background is now running. Wait until background processing is completed, or terminate background and start ITOS.
EXECUTIVE PROGRAM NOT LOADED – REQUEST REJECTED	Tried to start with programs TSLOG or ULBUFF not loaded in program library, or ULBUFF buffer not sized correctly for the number of terminals in the system. Check and load programs or modify buffer size then restart.
FILE NAME: xxxxxxxx ERROR STATUS= $nnnn –REQUEST REJECTED	During an attempt to start ITOS, a file error occurred while building system files. Try restarting with the backup system volume (if any), or call for system maintenance on drive 0, or INIT and reload files. The last step is a drastic action.
ATTENTION: xxxxxxxx HAS BEEN DISMOUNTED	File manager detected a mass memory error on a volume other than SYSVOL. Try to remount. If error persists, call for system maintenance on disk drive.
Jmnn FM RTJC=$nnnn,reqtyp file/user	A file error was received by a deferred batch driver. Jmnn refers to the job number. The file manager error status, file request name, file name, and owner are also included in the message.

TABLE E-2. ITOS DIAGNOSTIC MESSAGES

Message	Meaning/ Action	Index
xx IS INVALID	Code entered in reply to menu is incorrect. Check entry and enter a legal code.	0015
mon dd yy hh:mm:ss	Time in month day year hour:minute:second format. Message is informational only.	0013
ACCESSING THE FILE MANAGER ERROR IN BATCH FILE	A file manager error was returned to the batch status program.	0418
A MASS MEMORY ERROR HAS OCCURRED. PROGRAM TERMINATED	Probable disk failure. Call for hardware maintenance.	0010
BATCH DRIVER BUSY ON THIS HOST	A SET cannot be performed on this host name because because it is currently being processed.	0405
CLASS CODE IS NOT A DISK	System configuration error. Volume information table (VIT) is not set up correctly in SYSDAT. Call for program maintenance.	0074
DIRECT FILE RECORD LENGTH EXCEEDS 256 WORDS	Records in a direct file are limited to 512 characters in length. Re-enter length parameter in DEFINE in UTIL.	0062
DUPLICATE HOST NAME	Cannot add a host that is already in the $$HGST file	0403
DUPLICATE LOGICAL UNIT	The specified logical unit is already assigned to another host.	0411
EQUIPMENT TYPE NOT FOUND	Invalid device name. Re-enter valid name.	0046
ERROR - ATTEMPT TO PERFORM STACKED I/O REQUESTS AT LOCATION xxxx	Debugging problem; fatal error. A previous I/O request was still active when the request at location $xxxx_{16}$ was made. Call for program maintenance.	0008
ERROR - ILLEGAL I/O REQUEST LOGICAL UNIT AT LOCATION xxxx	Debugging problem; fatal error. The I/O request at $xxxx_{16}$ specifies a logical unit that is not allowed for ITOS. Call for program maintenance.	0007
ERROR - ATTEMPT TO MODIFY THE FILE REQUEST BUFFER AT LOCATION xxxx	Debugging problem; fatal error. File request at $xxxx_{16}$ contains parameters within REQBUF. Call for program maintenance.	0011
ERROR - ILLEGAL FILE REQUEST PARAMETER AT LOCATION xxxx	Debugging problem; fatal error. File request parameter at $xxxx_{16}$ is illegal. Call for program maintenance.	0004
ERROR - ILLEGAL FILE MANAGER REQUEST AT LOCATION xxxx	Debugging problem; fatal error. File manager request interceptor presented invalid index to file manager executive. Error in interceptor operation. Call for program maintenance. Location is in hexadecimal format.	0003
ERROR - ILLEGAL PROGRAM ATTACHMENT AT LOCATION xxxx	Debugging problem; fatal error. Either ATTACH executive presented an invalid multiuser program to be attached, or total program size (root and multiuser) exceeds user area size. Call for program maintenance.	0002
ERROR - PROGRAM PROTECT VIOLATION AT LOCATION xxxx	Debugging problem; fatal error. An illegal reference to protected memory was detected. Call for program maintenance.	0001
ERROR - xxxxxxxx IS TOO LARGE TO BE EXECUTED	Debugging problem; fatal error. xxxxxxxx is a program library program that is too large for the user area. Call for program maintenance.	0018
ERROR - ILLEGAL REQUEST PARAMETER AT LOCATION xxxx	Debugging problem; fatal error. A system request contains the illegal parameter. Call for program maintenance.	0006

TABLE E-2. ITOS DIAGNOSTIC MESSAGES (Contd)

Message	Meaning/Action	Index
ERROR – ILLEGAL MONITOR REQUEST AT LOCATION xxxx	Debugging problem; fatal error. The monitor request at $xxxx_{16}$ is not available to ITOS. Call for program maintenance.	0005
ERROR IN COMPUTING MM WORD ADDR	Debugging problem; fatal error. The file manager has an internal error. Call for program maintenance.	0071
ERROR STATUS xxxx DURING GETS OF FILE xxxxxxxx		0022
ERROR STATUS xxxx DURING FORCE FILE CLOSE		0023
ERROR STATUS xxxx DURING READR OF FILE xxxxxxxx	Debugging problem. Find status bit meanings from file manager reference manual. Call for program maintenance to take appropriate action. Value $xxxx_{16}$ is in hexadecimal format.	0021
ERROR STATUS xxxx DURING CLOSE OF FILE xxxxxxxx		0020
ERROR STATUS xxxx DURING OPEN OF FILE xxxxxxxx		0019
FCB INDEX OUT OF RANGE	Status command error. Call for program maintenance.	0058
FILE COULD NOT BE LOCATED	Check for erroneous file/owner/volume name. Retry.	0034
FILE FORMERLY OPENED FOR COMPRESSION	File cannot be used until compression is completed. Try again later.	0072
FILE HAS A SET OF LOCKED RECORDS	Locked records cannot be used. Try again later.	0061
FILE IS CURRENTLY LOCKED	Locked file cannot be used. Try again later.	0042
FILE IS CURRENTLY OPEN	File cannot be used by this owner at this terminal. Try again later.	0038
FILE NAME/OWNER NOT UNIQUE	Check for erroneous file/owner/volume name. If correct, rename the file.	0057
FILE REQUEST BUFFER NOT PROPERLY INITIALIZED		0079
FILE REQUEST ILLEGAL	Illegal file manager request. Call for program maintenance.	0037
FILE REQUEST BUFFER NOT INITIALIZED		0059
FILE REQUEST REJECTED		0033
FILE SPECIFIED SHOULD BE A DIRECT FILE	Copy error. Redefine file into which data is to be copied as a direct file.	0080
FILE TYPE NOT EQUAL	Check for erroneous file name or improper file definition. Correct and retry.	0066
HOST NAME NOT FOUND	Cannot perform SET request because host name is not in $$HOST file.	C401
ITOS LOG OFF hh:mm:ss	Informational only. Time in hour:minute:second format.	0014
ILLEGAL COMMAND FORMAT	Re-enter command correctly.	0032
ILLEGAL GETFLD STATUS	Utility error. Call for program maintenance.	0040

TABLE E-2. ITOS DIAGNOSTIC MESSAGES (Contd)

Message	Meaning/Action	Index
ILLEGAL LOG IN	Check password and user ID and proper terminal ($$). Then retry logging on operation.	0016
ILLEGAL PARAMETER RECEIVED	Re-enter parameter correctly.	0069
ILLEGAL TO DELETE LOCAL HOST	The host name LOCL must always be present in the $$HOST file.	0412
INDEX TOO BIG FOR UTILITY ORDERED LOAD	The files index requires more levels than the ordered LOAD module can support. Load the file with an unordered LOAD.	0082
INSUFFICIENT MM FILE SPACE	1. Insufficient space in file; compress or define new file with more space; then copy. Rename after deleting former file. 2. Insufficient space on volume; purge and delete or use another volume.	0055
INSUFFICIENT FID SPACE FOR FILES	Ran out of total mass memory space. Purge or delete files and retry.	0056
INTERNAL FM ERROR	Call for program maintenance.	0043
INTERNAL UTILITY ERROR	Call for program maintenance.	0070
INVALID JOB NUMBER	The format of the job number is incorrect.	0415
INVALID LOGICAL UNIT	The specified logical unit is not valid for the request.	0408
INVALID OWNER IDENTIFICATION	The file owner specified in the BATCH command is invalid.	0416
INVALID SYSTEM PERIPHERAL NAME	Re-enter proper peripheral name.	0063
JOB ALREADY DISCARDED	Cannot discard this job because it is already discarded.	0417
JOB INACTIVE	Cannot discard this job because it does not exist.	0420
JOB NOT FOUND	The job number specified does not exist.	0419
JOB(S) PENDING FOR THIS HOST	A DEL cannot be performed on this host because it is currently being processed.	0406
LOGICAL UNIT ALREADY SET FOR THIS HOST	Cannot SET a host on this logical unit because it is already SET. Perform a SET with Lu=0 first.	0410
MAX. NO OF OPEN FILES, RETRY	Try again later. If this appears frequently, may need to resize table in SYSDAT.	0045
MAX. NO OF OPEN FILES FOR A SINGLE USER	Try again later. If this occurs frequently, may need to resize table in SYSDAT.	0044
MISSING PARAMETER	Supply the parameter.	0039
MM I/O ERROR WAS NOTED	File manager error. Call for program maintenance.	0035
MOUNTED VOLUME HAS OPEN FILES	Tried to dismount while volume was being used. Take status to find which file(s) open. Close, if necessary having users log off terminals. Then retry.	0050
NO *JOB RECORD IN THE INPUT FILE	A *JOB record for job number tagging was not found in the input file.	0414
NO JOBS TO PRINT	There are no jobs in the print queue for this host.	0421
NO KEY ENTERED FOR INDEXED FILES	Missing parameter for DEFINE. Enter the primary key at a minimum.	0054

TABLE E-2. ITOS DIAGNOSTIC MESSAGES (Contd)

Message	Meaning/Action	Index
NO ROOM IN BATCH FILE	All queue entries for the host are used. Either start processing the host jobs or discard jobs which have already been processed.	0413
NO ROOM IN HOST FILE	Cannot add anymore host names. Delete unused or inactive names.	0400
NO SUCH HOST	The host name specified in the PRINT request does not exist.	0422
NOTICE – THE ITOS SYSTEM HAS BEEN DISABLED	Informational only. System can be restarted from the master terminal.	0009
OUT OF ORDER PRIMARY KEY	A records primary key value was not greater than the previous records key value.	0083
OUTPUT NOT RECEIVED	Cannot dispose of the job because it has not yet been received from the host.	0424
PARAMETER ENTRY ERROR	Re-enter parameter correctly.	0052
PARAMETER MUST BE ADD OR DEL	The only options allowed are ADD or DEL.	0402
PRIMARY KEY NOT UNIQUE	LOAD: During loading of an indexed file's records a second record was found to have a primary key identical to a previous record's primary key. Change the record key and load again.	0067
PROCEDURE xxxxxxxx CANNOT BE FOUND	1. Procedure not defined, or 2. Procedure does not have 80-character non-sector aligned records, or 3. Error occurred while trying to get file. If error caused by 1 or 2, verify procedure; if error not 1 or 2, call for program maintenance.	0024
PROGRAM ABORTED	Informational only. CONTROL A has been accepted and executed.	0012
PROTOCOL TYPE MUST BE 200UT OR HASP	The only valid remote host protocol types supported by COMM 18 are 200UT or HASP.	0404
RECORD IS LOCKED BY ANOTHER USER	Try again later.	0078
RECORD LOCK-TABLE IS CURRENTLY FULL	Try again later. If this occurs frequently, may need to resize table in SYSDAT.	0077
RECORD LENGTH TOO LARGE FOR THIS COMMAND	Records can be no larger than 512 characters for LIST, COPY, LOAD, DUMP, and RELOAD.	0064
RECORD LENGTH NOT EQUAL	COPY error. Redefine the file into which records will be copied so record length is same size as other file. Then retry.	0065
REQUEST xxxxxxxx CANNOT BE FOUND	xxxxxxxx is neither a procedure nor a program name. Re-enter correct name.	0017
REQUESTED COMMAND IS NOT LEGAL	Check for erroneous command. Re-enter correct command.	0031
REQUEST UTILITY PROCESSOR NOT FOUND	Installation error or internal error that removed a utility program from the library. Call for program maintenance.	0030
SUBRCM CALL ERROR – BUFFER LENGTH GREATER THAN 80	Terminal manager request error; correct the erroneous request parameter.	0097
SUBRCM CALL ERROR – ILLEGAL 'DTYPE'		0098

TABLE E-2. ITOS DIAGNOSTIC MESSAGES (Contd)

Message	Meaning/Action	Index
SUBRCM CALL ERROR – BUFFER LENGTH LESS THAN 1	Terminal manager request error; correct the erroneous request parameter.	0096
SUBRCM CALL ERROR – ILLEGAL REQUEST TYPE		0091
SUBRCM CALL ERROR – X COORDINATE GREATER THAN 79		0093
SUBRCM CALL ERROR – X COORDINATE LESS THAN ZERO		0092
SUBRCM CALL ERROR – Y COORDINATE GREATER THAN 23		0095
SUBRCM CALL ERROR – Y COORDINATE LESS THAN ZERO		0094
TAPE RECORD EXCEEDS INTERNAL BUFFER SIZE	The tape being reloaded was dumped on a system with a larger I/O buffer than this system.	0081
THIS COMMAND IS ALLOWED ONLY IF ITOS IS DISABLED	ITOS must be stopped to execute PURGE and SAVE. Stop system and re-enter request.	0075
THIS IS A SUPERVISOR COMMAND ONLY	Re-enter command from master terminal.	0076
THIS VOLUME IS DISMOUNTED	Internal utility error. Call for program maintenance.	0049
UNDEFINED FM STATUS ERROR	Bits in status word that should not be there. Call for program maintenance.	0060
VIT COULD NOT BE FOUND	Invalid DK or D2 parameter. Correct parameter and re-enter.	0041
VOLUME 2 MAY NOT BE MOUNTED	Volume onto which another volume is being saved cannot be mounted. Dismount the volume and retry.	0073
VOLUME HAS OPEN FILES, NOT	Tried to dismount while volume was being used. Take status to find which file(s) are open. Close files, if necessary, by having users log off terminals. Then retry.	0053
VOLUME SPECIFIED NOT MOUNTED AND READY	Check for erroneous volume name. Otherwise mount the correct volume.	0036
VOLUME WAS ALREADY MOUNTED	Informational only. Volume is now mounted and ready for use.	0048
WRONG KEY VALUE	File manager error during creation of file. Call for program maintenance.	0068
WRONG MM UNIT DEFINED	Error in DK or D2 parameters; correct and retry.	0047
WRONG VOLUME MOUNTED	MOUNT used wrong volume name. The volume on the specified disk drive has not been mounted. Check for wrong name or wrong disk pack and take appropriate action. Then retry.	0051
Unused numbers	These should never appear	0025 - 0029, 0081 - 0090, 0099, 0100, 0261 - 0300

TABLE E-3. EDITOR DIAGNOSTIC MESSAGES

Message	Meaning/ Action	Index
AN ERROR OCCURRED WHEN BUILDING THE STATEMENT LAB LE INDEX	Internal index for line numbers failed. Use GET command again. If this does not correct error, log off and relog onto terminal and call editor again.	0304
AN ERROR OCCURRED WHEN INITIALIZING THE EDITOR ISTAT = xxxx	Editor cannot operate. Call editor again. ISTAT is file manager status word in hexadecimal.	0336
CHARACTER STRING TOO LONG	More than 20 characters in a string. Correct SEARCH or CHANGE command and re-enter it.	0317
COMMAND NAME NOT UNIQUE	Add at least one more character to the editor command call being used and re-enter it.	0308
COULD NOT LOCATE FILE filename USER owner id	Filename not defined or operator logged onto terminal using a user ID that won't permit access to this file.	0332
DELIMITER MISSING	Correct CHANGE or SEARCH command and re-enter it.	0316
FILE filename IS NOT AN EDITOR FILE	File must be a direct file if created by UTIL. Otherwise it must be a sequential file with 80-character records.	0303
FILE filename IS LOCKED. TRY AGAIN LATER	Someone else locked the file for use or updating. Try again later.	
FILE MANAGER ERROR IN STATEMENT LABEL INDEX FILE ISTAT = xxxx	File manager problem. Find status bit from File Manager Reference Manual. Call for program maintenance to take appropriate action. ISTAT value is in hexadecimal format.	0337
FILE MANAGER ERROR WHEN CLOSING FILE filename ISTAT = xxxx		0339
FILE MANAGER ERROR WHEN CLOSING SCRATCH FILE ISTAT = xxxx		0346
FILE MANAGER ERROR WHEN CREATING SCRATCH FILE ISTAT = xxxx		0343
FILE MANAGER ERROR WHEN DELETING FROM FILE filename ISTAT = xxxx		0335
FILE MANAGER ERROR WHEN INITIALIZING SCRATCH FILE ISTAT = xxxx		0340
FILE MANAGER ERROR WHEN OPENING FILE filename ISTAT = xxxx		0331
FILE MANAGER ERROR WHEN OPENING SCRATCH FILE ISTAT = xxxx		0344
FILE MANAGER ERROR WHEN READING FILE filenam ISTAT = xxxx		0334
FILE MANAGER ERROR WHEN RENAMING SCRATCH FILE ISTAT = xxxx		0347
FILE MANAGER ERROR WHEN UPDATING FCB FILE ISTAT = xxxx		0338
FILE MANAGER ERROR WHEN UPDATING FCB FOR FILE filename ISTAT = xxxx		0342
FILE MANAGER ERROR WHEN UPDATING FILE filename ISTAT = xxxx		0341
FILE MANAGER ERROR WHEN WRITING SCRATCH FILE ISTAT = xxxx		0345

TABLE E-3. EDITOR DIAGNOSTIC MESSAGES (Contd)

Message	Meaning	Index
ILLEGAL LINE NUMBER xxxxxx	No file name or name less than 8 characters. Re-enter proper name.	0305
ILLEGAL LINE NUMBER xxxxxx SPECIFIED	Illegal character (not a numeral) in current line number filed. Re-enter line number.	0301
INCORRECT TAP STOP ORDER	Tabs must be in ascending order. Re-enter tabs in that order.	0313
INVALID COMMAND	Not the name of an editor command. Re-enter a proper command name.	0309
INVALID DELIMITER	Used a comma (,) as a delimiter. Choose another character for delimiter and re-enter command	0315
INVALID FIELD	Information in parameter field cannot be interpreted, or field is too large. Correct and re-enter.	0306
INVALID FORMAT SPECIFICATION	AUTO command: t must be a blank or one of H, F, E, L, I, C, O, or *. Re-enter using a valid t value.	0311
INVALID NUMERIC VALUE	N, I in AUTO; nn, i in RESEQ; N in STAB; or k_1, k_2 were not pure numerics or value was too large. Correct and re-enter.	0307
INVALID RPG ARRAY DATA LINE NUMBER	*format for AUTO or STAB: Consecutive line numbers must ascend in value. Re-enter in correct order.	0314
LINE NUMBER OVERFLOW	Line number greater than 32,767.	0312
NO FILE OPEN FOR EDITOR USE	Enter a GET command to open the file	0319
NO PROGRAM ID ON H FORMAT SPECIFICATION	AUTO: must supply P parameter if t = H. Re-enter AUTO with P specified.	0302
RESEQUENCE WITH A LOWER BASE AND/OR INCREMENT	Resequencing with the specified base and/or increment caused a line number overflow. Choose smaller base or increment.	0318
SAVRTN STACK OVERFLOW	Not used	0320
SAVRTN STACK UNDERFLOW	Not used	0321
WRONG FORMAT TYPE FOR THIS FILE	T parameter incorrect: using an RPG format for non-RPG file or the reverse. Re-enter proper command or use proper format for command entered.	0310
Unused numbers	These should never appear.	0322 - 0330, 0348, 0349

TABLE E-4. ITOS SORT DIAGNOSTIC MESSAGES

Message	Meaning	Index
aa . . . aa	Numerical data with a prefix.	0362
aa . . . aa	Card image. Appears with other message specifying error.	0370
ABNORMAL ERROR = (error)	Unusal error condition detected.	0360
ADDROUT SORTS ONLY 1 FILE	Use only one input file for an ADDROUT sort.	0378
BLKSIZ/RECLTH .NE. 1,2,3, . . .	Record length parameter is not a divisor of block size parameter.	0361
CANNOT OPEN INPUT FILE	Cannot sort the requested file since it cannot be opened to read.	0367
CLOSEFL REQIND = $xxxx	Status word for CLOSEFL operation when CLOSEFL failed.	0350
CREATE REQIND = $xxxx	Status word for CREATE operation when CREATE failed.	0353
DELETE REQIND = $xxxx	Status word for DELETE operation when DELETE failed.	0352
DONE = (number)	Number of records processed.	0364
EXPECTED aa . . aa FOUND bb	Sort did not find the type of parameter expected. Sorting is aborted.	0366
FATAL ERROR	Sorting operation was aborted.	0363
FILNAM = aaaaaaaa, bbbbbbbb	File name owner name (reconstructed)	0376
FN = aaaaaaaa, bbbbbbbb	Input filename, owner (input and output)	0359
GETFCB REQIND = $xxxx	Status word for GETFCB operation when GETFCB failed.	0356
GETS REQIND = $xxxx	Status word for GETS operation when GETS failed.	0354
INTERPHASE RECORD COUNTS DISAGREE	Number of output records does not equal number of input sort records.	0369
INPUT FILE LENGTHS ARE NOT EQUAL	Cannot sort files with unequal record lengths.	0375
KEY FIELD EXTENDS BEYOND END OF RECORD	Key ends outside of record.	0381
OPENFL REQIND = $xxxx	Status word for OPENFL operation when OPENFL failed.	0351
OUTPUT FILE RECORD LENGTH IS ZERO	Data only sort option where all of input record was used for keys.	0377
OUTPUT RECORD COUNT BAD	Improper number of records in output file.	0374
PASSED = (number)	The specified number of records were either processed or skipped.	0371
PUTS REQIND = $xxxx	Status word for PUTS operation when PUTS failed.	0355
SEQ. DIR. ERROR	Sequence directory read or write error.	0372
START OF KEY FIELD OUTSIDE OF RECORD	Key position starts before or after record.	0380
TOO LITTLE CORE	Requested inputs cannot be processed in amount of core space available.	0368

TABLE E-4. ITOS SORT DIAGNOSTIC MESSAGES (Contd)

Message	Meaning	Index
TOO LITTLE DISK	Inadequate disk space for sorting operation.	0373
TYPE-IN ERROR	Sort cannot interpret command statement in the procedure stream.	0365
UPDFCB REQIND = $xxxx	Status word for UPDFCB operation when UPDFCB failed.	0357
VOLUME = (name)	Volume name	0358
VOLUME (name) NOT MOUNTED	Volume specified for output file is not mounted.	0379

The following tables show the DSORT collating sequence and correlate ASCII and EBCDIC collating sequences.

TABLE F-1. EBCDIC COLLATING SEQUENCE AND HEXADECIMAL CHARACTER EQUIVALENTS

Collating Sequence	Character	Hexadecimal Equivalent	Collating Sequence	Character	Hexadecimal Equivalent
1	Blank	40	33	F	C6
2	¢	4A	34	G	C7
3	.	4B	35	H	C8
4		4C	36	I	C9
5	(	4D	37		D0
6	+	4E	38	J	D1
7		4F	39	K	D2
8		50	40	L	D3
9		5A	41	M	D4
10	$	5B	42	N	D5
11	*	5C	43	O	D6
12	)	5D	44	P	D7
13	;	5E	45	Q	D8
14		5F	46	R	D9
15	- (minus)	60	47	S	E2
16	/	61	48	T	E3
17	,	6B	49	U	E4
18	%	6C	50	V	E5
19	_ (underscore)	6D	51	W	E6
20		6E	52	X	E7
21	?	6F	53	Y	E8
22	:	7A	54	Z	E9
23		7B	55	0	F0
24		7C	56	1	F1
25	'	7D	57	2	F2
26	=	7E	58	3	F3
27	"	7F	59	4	F4
28	A	C1	60	5	F5
29	B	C2	61	6	F6
30	C	C3	62	7	F7
31	D	C4	63	8	F8
32	E	C5	64	9	F9

TABLE F-2. PRINTABLE CHARACTERS GROUPED BY EQUAL ZONES

Group	Character
1	¢ . < (†
2	! $ *) ;
3	/ , (comma) % _ (underscore) > ?
4	: ' (apostrophe) = "
5	A B C D E F G H I
6	- (minus) J K L M N O P Q R
7	S T U V W X Y Z

TABLE F-2. PRINTABLE CHARACTERS GROUPED BY EQUAL ZONES (Contd)

Group	Character
8	blank 0 1 2 3 4 5 6 7 8 9

TABLE F-3. PRINTABLE CHARACTERS GROUPED BY EQUAL DIGITS

Group	Character
1	blank - (minus) 0 (zero)
2	/ A J 1
3	B K S 2
4	C L T 3
5	D M U 4
6	E N V 5
7	F O (letter O) W 6
8	G P X 7

TABLE F-3. PRINTABLE CHARACTERS GROUPED BY EQUAL DIGITS (Contd)

Group	Character
9	H Q Y 8
10	I R Z 9
11	¢ | :
12	. $, (comma)
13	< * %
14	( ) _ (underscore) ' (apostrophe)
15	+ ; > =
16	! ? "

TABLE F-4. ASCII CHARACTER SET

The 1963 American Standard Code for Information Interchange (ASCII) is used by MSOS. ASCII code uses eight bits: bit 8, which is always zero, is omitted in the table below. Bits 1 through 4 contain the low-order four bits of code for the character in that row. Bits 5 through 7 contain the high-order three bits of the code for the character in that column. The code is given in ascending sequence.

ASCII Symbol	Bit Configuration	Hexadecimal Number	Meaning
NULL	000 0000	0	Null/idle
SOM	000 0001	1	Start of message
EOA	000 0010	2	End of address
EOM	000 0011	3	End of message
EOT	000 0100	4	End of transmission
WRU	000 0101	5	Who are you
RU	000 0110	6	Are you
BELL	000 0111	7	Audible signal
FE_0	000 1000	8	Format effector
HT/SK	000 1001	9	Horizontal tab skip (punched card)
LF	000 1010	A	Line feed
V_{TAB}	000 1011	B	Vertical tabulation
FF	000 1100	C	Form feed
CR	000 1101	D	Carriage return
SO	000 1110	E	Shift out
SI	000 1111	F	Shift in
DC_0	001 0000	10	Device control/data link escape
DC_1	001 0001	11	Device controls
DC_2	001 0010	12	
DC_3	001 0011	13	
DC_4 (STOP)	001 0100	14	Device control/stop
ERR	001 0101	15	Error
SYNC	001 0110	16	Synchronous idle
LEM	001 0111	17	Logical end of media
S_0	001 1000	18	Information separators
S_1	001 1001	19	
S_2	001 1010	1A	
S_3	001 1011	1B	
S_4	001 1100	1C	
S_5	001 1101	1D	
S_6	001 1110	1E	
S_7	001 1111	1F	

TABLE F-4. ASCII CHARACTER SET (Continued)

8-Bit ASCII Codes	171x-1 TTY Array	171x-2 TTY Array	026 Punches	029 Punches	6-Bit Ext. BCD Mag Tape	8-Bit ASCII Codes	171x-1 TTY Array	172x-2 TTY Array	026 Punches	029 Punches	6-Bit Ext. BCD Mag Tape
20_{16}	Space	Space	No Punch	No Punch	20_8	40_{16}	±	±	0-8-7	8-4	37
21			11-8-2	12-8-7	52	41	A	A	12-1	12-1	61
22	]	]	8-7	8-7	17	42	B	B	12-2	12-2	62
23	•	•	12-8-7	8-3	77	43	C	C	12-3	12-3	63
24	$	$	11-8-3	11-8-3	53	44	D	D	12-4	12-4	64
25	%	%	0-8-5	0-8-4	35	45	E	E	12-5	12-5	65
26	†	†	8-2	12	00 (35)	46	F	F	12-6	12-6	66
27	'	'	8-4	8-5	14	47	G	G	12-7	12-7	67
28	(	(	0-8-4	12-8-5	34	48	H	H	12-8	12-8	70
29	)	)	12-8-4	11-8-5	74	49	I	I	12-9	12-9	71
2A	*	*	11-8-4	11-8-4	54	4A	J	J	11-1	11-1	41
2B	+	+	12	12-8-6	60	4B	K	K	11-2	11-2	42
2C	,	,	0-8-3	0-8-3	33	4C	L	L	11-3	11-3	43
2D	-	-	11	11	40	4D	M	M	11-4	11-4	44
2E	.	.	12-8-3	12-8-3	73	4E	N	N	11-5	11-5	45
2F	/	/	0-1	0-1	21	4F	O	O	11-6	11-6	46
30	0	0	0	0	12	50	P	P	11-7	11-7	47
31	1	1	1	1	01	51	Q	Q	11-8	11-8	50
32	2	2	2	2	02	52	R	R	11-9	11-9	51
33	3	3	3	3	03	53	S	S	0-2	0-2	22
34	4	4	4	4	04	54	T	T	0-3	0-3	23
35	5	5	5	5	05	55	U	U	0-4	0-4	24
36	6	6	6	6	06	56	V	V	0-5	0-5	25
37	7	7	7	7	07	57	W	W	0-6	0-6	26
38	8	8	8	8	10	58	X	X	0-7	0-7	27
39	9	9	9	9	11	59	Y	Y	0-8	0-8	30
3A	:	:	8-5	8-2	15	5A	Z	Z	0-9	0-9	31
3B	;	;	11-8-6	11-8-6	56	5B			12-8-5	12-8-2	75
3C			12-8-6	12-8-4	76	5C			0-8-2	0-8-2	36
3D	=	=	8-3	8-6	13	5D			11-8-5	11-8-2	55
3E			8-6	0-8-6	16	5E			11-8-7	11-8-7	57
3F	?	?	12-8-2	0-8-7	72	5F		-	0-8-6	0-8-5	32

Refer to note 2.
Refer to note 4.

NOTES

1. The 171x-2 teletypewriter (TTY) array is the ASCII 68, 64 character subset. This array is the same as used on the 171x-3 devices which receive from a 1774.
2. To operate in 026 punched card mode, ASCII 63 options are selected. To operate in 029 punched card mode, ASCII 68 options are selected. These options are assembly-time options for each driver affected.
3. The CDC Standard 1.10.003 is supported by an assembly option. For CDC ASCII mode of operation, the card punches 12-8-2 and 12-0 are stored internally as 7B. The card punches 11-8-2 and 11-0 are stored internally as 7D. For line printer operations, the internal codes 7B and 7D are converted to 5B and 5D to allow printing the hardware compatible graphic characters (left bracket) and (right bracket).
4. Since 173x magnetic tape controllers do not provide any code conversion, BCD code 00 is illegal and causes a noise record or BCD code 35 is substituted for the illegal 00 code to prevent tape errors.

   On tape write operations the ASCII codes 25_{16} (%) and 26_{16} (†) are written as BCD 35_8.

   On tape read operations the BCD code 35_8 is always translated to an ASCII $25 (%).

TABLE F-5. ASCII TO EBCDIC TRANSLATION TABLE

CYBER 18 Graphic	ASCII Code	EBCDIC $Code_{16}$	CYBER 18 Graphic	ASCII Code	EBCDIC $Code_{16}$	CYBER 18 Graphic	ASCII Code	EBCDIC $Code_{16}$	CYBER 18 Graphic	ASCII Code	EBCDIC $Code_{16}$
Null	00			40	7C		80			C0	23
SDM	01		A	41	C1		81			C1	41
STX	02		B	42	C2		82			C2	42
ETX	03		C	43	C3		83			C3	43
EOT	04		D	44	C4		84			C4	44
ENQ	05		E	45	C5		85			C5	45
ACK	06		F	46	C6		86			C6	46
BEL	07		G	47	C7		87			C7	47
BS	08		H	48	C8		88			C8	48
HT	09		I	49	C9		89			C9	49
LF	0A		J	4A	D1		8A			CA	
VT	0B		K	4B	D2		8B			CB	
FF	0C		L	4C	D3		8C			CC	
CR	0D		M	4D	D4		8D			CD	
SO	0E		N	4E	D5		8E			CE	
SI	0F		O	4F	D6		8F			CF	
DLE	10		P	50	D7		90			D0	29
DCI	11		Q	51	D8		91			D1	24
DC2	12		R	52	D9		92			D2	2E
DC3	13		S	53	E2		93			D3	3C
DC4	14		T	54	E3		94			D4	28
ANK	15		U	55	E4		95			D5	2B
SYN	16		V	56	E5		96			D6	2A
ETB	17		W	57	E6		97			D7	26
CAN	18		X	58	E7		98			D8	51
EM	19		Y	59	E8		99			D9	52
SUB	1A		Z	5A	E9		9A			DA	
ESC	1B		[	5B	4A		9B			DB	
FS	1C			5C	5F		9C			DC	
GS	1D		]	5D	27		9D			DD	
RS	1E			5E	4F		9E			DE	
US	1F			5F	6D		9F			DF	
Blank	20	40		60	2D		A0			E0	
	21	5A	a	61	2F		A1			E1	
"	22	7F	b	62			A2			E2	53
	23	7B	c	63			A3			E3	54
$	24	5B	d	64			A4			E4	55
%	25	6C	e	65			A5			E5	56
	26	5O	f	66			A6			E6	57
'	27	7D	g	67			A7			E7	58
(	28	4D	h	68			A8			E8	59
)	29	5D	i	69			A9			E9	21
*	2A	5C	j	6A			AA			EA	
+	2B	4E	k	6B	2C		AB			EB	
,	2C	6B	l	6C	25		AC			EC	
-	2D	6O	m	6D	3B		AD			ED	
.	2E	4B	n	6E	3E		AE			EE	
/	2F	61	o	6F	3F		AF			EF	
0	30	F0	p	70			B0			F0	30
1	31	F1	q	71			B1			F1	31
2	32	F2	r	72			B2			F2	32
3	33	F3	s	73			B3			F3	33
4	34	F4	t	74			B4			F4	34
5	35	F5	u	75			B5			F5	35
6	36	F6	v	76			B6			F6	36
7	37	F7	w	77			B7			F7	37
8	38	F8	x	78			B8			F8	38
9	39	F9	y	79			B9			F9	39
:	3A	7A	z	7A	3A		BA			FA	
;	3B	5E		7B	C0		BB			FB	
<	3C	4C		7C	20		BC			FC	
=	3D	7E		7D	D0		BD			FC	
>	3E	6E		7E	3D		BE			FE	
?	3F	6F	Delete	7F	22		BF			FF	

NOTE: Blank entries indicate that the code is not changed (ASCII 01 is EBCDIC 01).

TABLE F-6. EBCDIC TO ASCII TRANSLATION TABLE

EBCDIC $Code_{16}$	ASCII Code	System/3 Graphic	EBCDIC $Code_{16}$	ASCII Code	System/3 Graphic	EBCDIC $Code_{16}$	ASCII Code	System/3 Graphic	EBCDIC $Code_{16}$	ASCII Code	System/3 Graphic
00			40	20	Blank	80	80		C0	7B	
01			41	C1		81			C1	41	A
02			42	C2		82			C2	42	B
03			43	C3		83			C3	43	C
04			44	C4		84			C4	44	D
05			45	C5		85			C5	45	E
06			46	C6		86			C6	46	F
07			47	C7		87			C7	47	G
08			48	C8		88			C8	48	H
09			49	C9		89			C9	49	I
0A			4A	5B		8A			CA	C	
0B			4B	2E	.	8B			CB		
0C			4C	3C	<	8C			CC		
0D			4D	28	(	8D			CD		
0E			4E	2B	+	8E			CE		
0F			4F	5E		8F			CF		
10			50	26		90			D0	7D	
11			51	D8		91			D1	4A	J
12			52	D9		92			D2	4B	K
13			53	E2		93			D3	4C	L
14			54	E3		94			D4	4D	M
15			55	E4		95			D5	4E	N
16			56	E5		96			D6	4F	O
17			57	E6		97			D7	50	P
18			58	E7		98			D8	51	Q
19			59	E8		99			D9	52	R
1A			5A	21		9A			DA		
1B			5B	24	$	9B			DB		
1C			5C	2A	*	9C			DC		
1D			5D	29	)	9D			DD		
1E			5E	3B	;	9E			DE		
1F			5F	5C		9F			DF		
20	7C		60	2D	-	A0			E0		
21	E9		61	2F	/	A1			E1		
22	7F		62			A2			E2	53	S
23	C0		63			A3			E3	54	T
24	D1		64			A4			E4	55	U
25	6C		65			A5			E5	56	V
26	D7		66			A6			E6	57	W
27	5D		67			A7			E7	58	X
28	D4		68			A8			E8	59	Y
29	D0		69			A9			E9	5A	Z
2A	D6		6A			AA			EA		
2B	D5		6B	2C	,	AB			EB		
2C	6B		6C	25	%	AC			EC		
2D	60		6D	5F	_	AD			ED		
2E	D2		6E	3E	>	AE			EE		
2F	61		6F	3F	?	AF			EF		
30	F0		70			B0			F0	30	0
31	F1		71			B1			F1	31	1
32	F2		72			B2			F2	32	2
33	F3		73			B3			F3	33	3
34	F4		74			B4			F4	34	4
35	F5		75			B5			F5	35	5
36	F6		76			B6			F6	36	6
37	F7		77			B7			F7	37	7
38	F8		78			B8			F8	38	8
39	F9		79			B9			F9	39	9
3A	7A		7A	3A	:	BA			FA		
3B	6D		7B	23	#	BB			FB		
3C	D3		7C	40	@	BC			FC		
3D	7E		7D	27	'	BD			FD		
3E	6E		7E	3D	=	BE			FE		
3F	6F		7F	22	"	BF			FF		

NOTE: Blank entries indicate that the code is not changed (EBCDIC 01 is ASCII 01).

H - Format Specifications

- 7 - Size to Compile
- 15 - Debug
- 21 - Inverted Print
- 26 - Alternate Collating Sequence
- 41 - 1P Forms Positioning
- 43 - File Translation
- 48 - Shared I/O
- 75 - Program Identification

F - Format Specification

- 7 - File Name
- 15 - File Type
- 19 - File Format
- 24 - Record Length
- 39 - Extension Code
- 53 - Continuation Line
- 60 - Buffer Offset Length
- 66 - File Addition/Unordered Output
- 75 - Program Identification

E - Format Specification

- 7 - Record Sequence of the Chaining file
- 11 - From File Name
- 19 - To File Name
- 27 - Table or Array Name
- 40 - Length of Entry
- 43 - Packed or Binary Field
- 46 - Alternate Table or Array Name
- 52 - Alternate Length of Entry
- 55 - Alternate Packed or Binary Field
- 58 - Comments
- 75 - Program Identification

L - Format Specification

- 7 - File Name
- 15 - Line Number - Number of Lines Per Page
- 18 - Form Length
- 20 - Line Number
- 23 - Overflow Line
- 25 - Line Number - Number of Lines Per Page
- 28 - Form Length
- 30 - Line Number Overflow
- 23 - Overflow Line
- 35 - Line Number - Number of Lines Per Page
- 38 - Form Length
- 40 - Line Number Overflow
- 43 - Overflow Line
- 45 - Line Number - Number of Lines Per Page
- 48 - Form Length
- 50 - Line Number Overflow
- 75 - Program Identification

I - Format Specification

- 7 - File Name
- 15 - Sequence
- 21 - Record Identification Code Position Field 1
- 28 - Record Identification Code Position Field 2
- 35 - Record Identification Code Position Field 3
- 42 - Stacker Select
- 52 - Decimal Positions
- 59 - Control Level
- 75 - Program Identification

C - Format Specification

7 - File Name

15 - Type

23 - Output Indicators

32 - Field Name

38 - Edit Codes

45 - Constant or Edit Word

75 - Program Identification

X - Format Specification

8 - Runtime Trace

10 - Compiles Trace

20 - Object Code Trace

27 - New Deck ID

75 - Program Identification

$$HOST FILE AND RECORD STRUCTURE

Figure H-1 is the structure of a sequential, eight-record $$HOST file and record:

Host name (Words 1 and 2)

1. Prespecified LOCL for local host
2. All others assigned to released records by HOST add command.
3. Release specified host name (blank name, zero LU) by HOST delete command if ST01-ST60 are all inactive (=0).

Type (Word 3)

1. Prespecified zero for local host
2. All others assigned by HOST add command

   1 200UT
   2 HASP

Input LU (Word 3)

1. Prespecified for local host
2. All others assigned by SET command for valid LU. LU is valid if the LU physical device table, word 8 equipment type is for the batch LU, and if the LU is not specified in another $$HOST file record.

JA (Word 3)

Batch driver job activity:

0 No activity (in other words, all queued jobs are done, or premature termination is specified by the SET command LU=0).

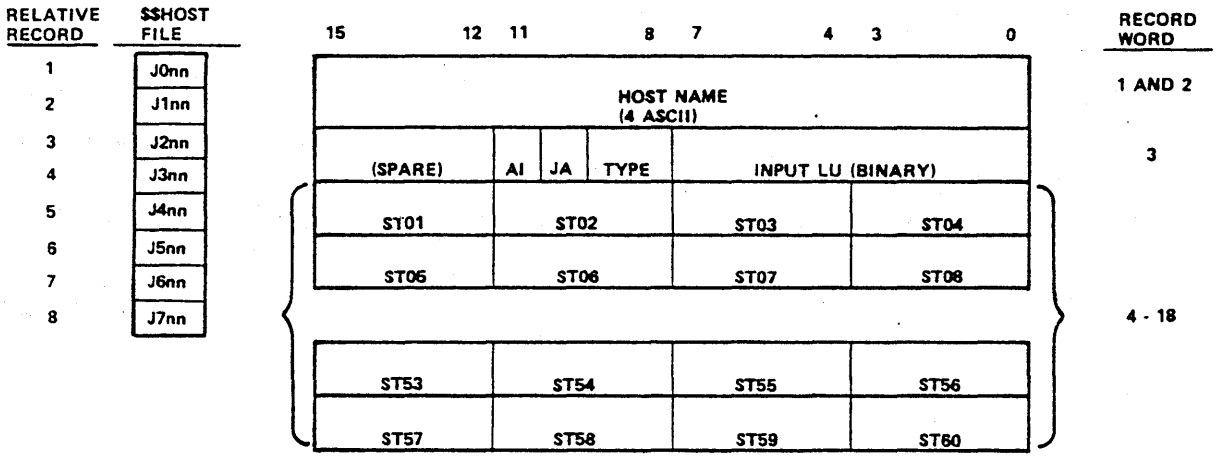
1 Driver activity (in other words, processing all queued jobs)

AI (Word 3)

Abort input driver processing

0 Continue normal processing

1 Terminate processing at completion of current job (set by SET command).



NOTES:

Jmnn = KEY-INDEX (ASCII) TO $$BATCH FILE = JOB NAME

WHERE: m = ARBITRARY HOST ID ($0 \leq m \leq 7$)

nn = ARBITRARY JOB ID FOR HOST M ($01 \leq nn \leq 60$)

Figure H-1. $$HOST File and Record Structure

STnn (Words 4-18) Job status code (0 - 16) for job name Jmnn

0 Inactive (in other words, no job is queued)

1 Not sent (in other words, job is queued for transmission)

2 Being sent (in other words, job transmission is in progress)

3 Sent (in other words, job transmission completed, awaiting output)

4 Output received (in other words, output transmitted, awaiting disposition)

5 Request print output (in other words, output awaiting printing)

6 Spare code

7 Job aborted (local host only)

8 Discard pending, job being sent (discard requested following start of transmission; is discarded upon reception).

9 Discard pending, job has been sent (is discarded upon reception of job from host)

A - F Spare codes

$$BATCH FILE AND RECORD STRUCTURE

Figure H-2 shows the structure of an indexed, 480-record $$BATCH file and record.

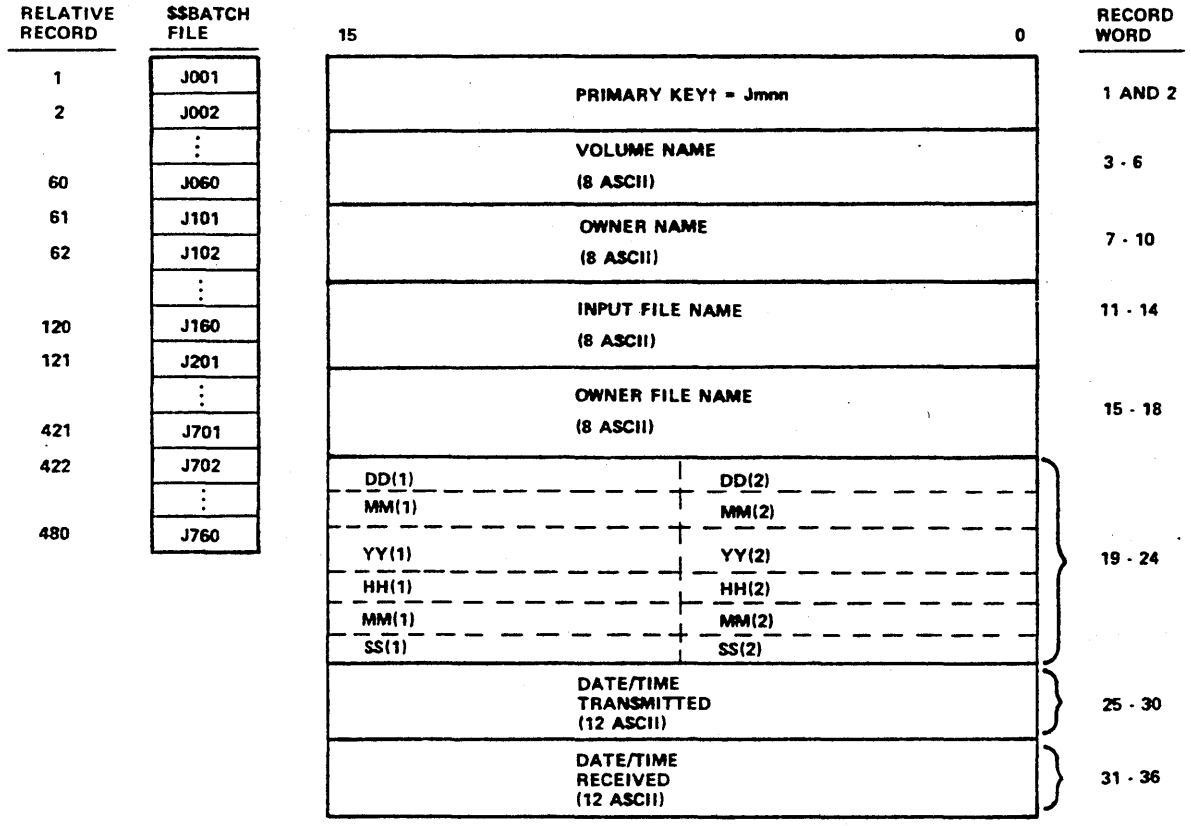


Figure H-2. $$BATCH File and Record Structure

Primary key (Words 1 and 2)	1. Prespecified Jmnn by ITOS system file creation 2. Structure allows positioning to record Jm01 by READR; thereafter Jm02 through Jm60 are retrieved sequentially by GETS.
Volume name (Words 3 - 6) Owner name (Words 7 - 10) Input file name (Words 11 - 14) Output file name (Words 15 - 18) Date/time requested (Words 19 - 24)	Filled by BATCH command processor after parameter verification
Date/time transmitted (Words 25 - 30)	Filled by batch input driver on commencing transmission to host.
Date/time Received (Words 31 - 36)	Filled by batch output driver on completion of data transmission from host.

$$PRINT FILE AND RECORD STRUCTURE

Figure H-3 shows the structure of an indexed, 50-record $$PRINT file and record.

Primary key (Words 1 and 2)	Prespecified PRnn by ITOS system file creation.
Volume name (Words 4 - 7)	Specified by deferred batch output driver when an unidentified job is received and file PRnn is created.
Time/date requested (Words 10 - 14)	Specifed by deferred batch output driver when a job is submitted for printing.
Number of records (Words 16 and 17)	1. Prespecified to ASCII blanks by ITOS system file creation. 2. Specified by deferred batch output driver to the number of records in the PRnn file when submitted for printing. 3. Reset to ASCII zeros by the PRINT utility after printing of PRnn file is completed.

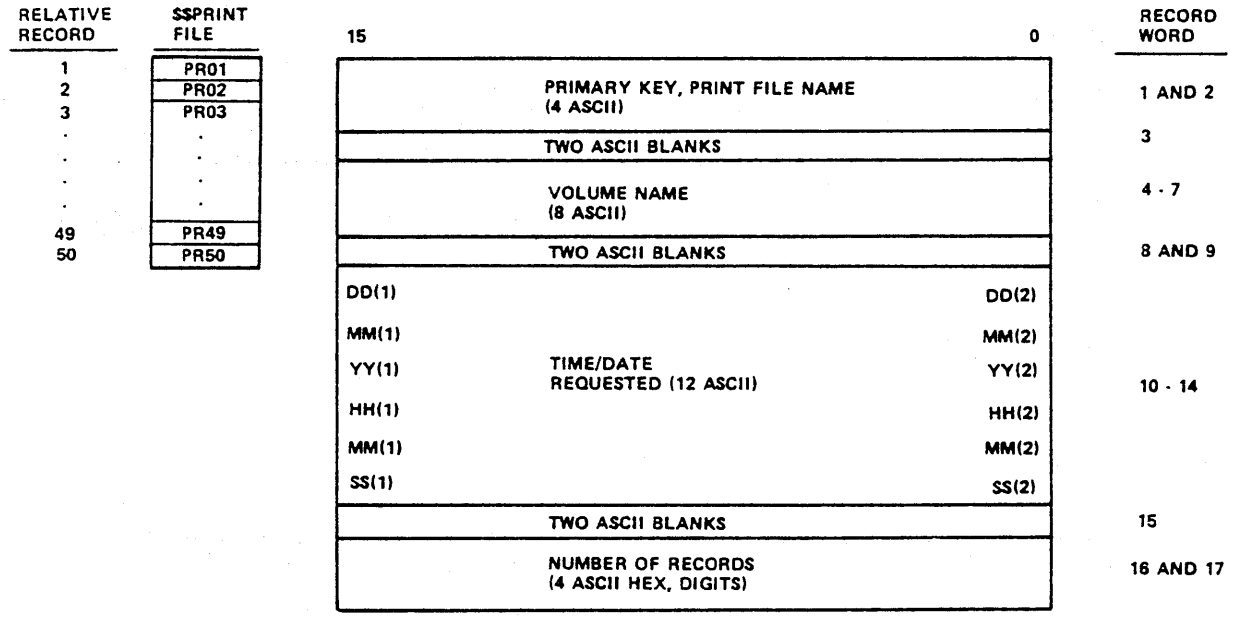


Figure H-3. $$PRINT File and Record Structure

INDEX

COMMENT SHEET

MANUAL TITLE CDC® Interactive Terminal-Oriented System (ITOS) Version 1

Reference Manual

PUBLICATION NO. 96768290 REVISION C

FROM NAME: _____

BUSINESS
ADDRESS: _____

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

CUT ALONG LINE

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

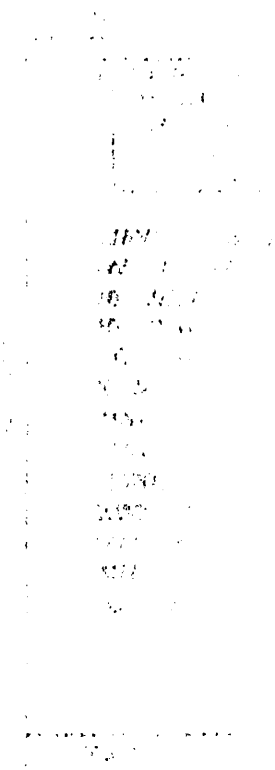
POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

CUT ALONG LINE

FOLD

FOLD



CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION