OASIS

System Reference

Manual

First printing: February, 1978
Revised: April, 1978
June, 1978
October, 1978
March, 1979
October, 1979

Written by: C. P. Williams

PROPRIETARY INFORMATION

The OASIS Operating System is a proprietary product developed by Timothy S. Williams and distributed by Phase One Systems, Inc., Oakland, California.

The software described in this document is furnished to the user under a license for use on a single computer system and may be copied (with inclusion of the copyright notice) only in accordance with the terms of such license.

BASIC is a registered trademark of the Trustees of Dartmouth College.

P R E F A C E

This manual describes the OASIS Operating System, version 5.4 and provides the detailed information necessary to use the operating system, utility programs, file management, user accounting, etc.

Note: This manual describes both single and multi-user versions of the operating system. For documentation purposes, any information that pertains to multi-user only will be denoted by the vertical bar character in both margins (like this paragraph).

Referenced or Related Material

The following manuals provide information describing other programs available to the OASIS user:

OASIS BASIC Language Reference Manual

OASIS SCRIPT Processor Reference Manual

OASIS Text Editor Reference Manual

OASIS EXEC Language Reference Manual

OASIS MACRO Assembler Language Reference Manual

OASIS LINK Editor Reference Manual *

OASIS DEBUG Reference Manual

OASIS Diagnostic & Conversion Utility Program Reference Manual

OASIS Implementation Guide *

* Manual not available at this time.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

CHAPTER 1

OASIS INTRODUCTION

This version of OASIS is single-user or multi-user, programming and operating system designed for Z80 based micro computers. It permits a wide range of peripherals (printers, terminals, disk drives, etc.) and up to 64K of memory.

1.1 OASIS Design Philosophy

The OASIS Operating System was designed with four major ideas in mind: 1) OASIS was to be implemented on micro-computers but have features traditionally found on mini and mainframe computers; 2) OASIS would be implemented on machines from several manufacturers; 3) the micro-computer market consists of mainly uninitiated computer users; 4) programers and systems designers are accustomed to having an operating system perform all of the tedius tasks required in program development and execution.

The first idea was easily solved as the background of the designers of OASIS is almost exclusively in mini and mainframe machines and both in OS design and applications design and programming. Users of OASIS will find many programs, concepts and syntax familiar if they have used mainframe computers or minicomputers.

The second idea was more difficult to implement because it implies machine independence. Although OASIS is not totally machine independent there are only a few sections of OASIS that require information specific to a machine. OASIS is designed very modularly, allowing for small changes to be made for implementation on another machine. All programs written by the user on one machine with OASIS can be executed on another with no changes to the user program.

The third idea of uninitiated users is the most difficult. Partial solutions to this problem were implemented by making all functions of the operating system consistent; by providing a "help" feature in each command; by using English words for command names; and by providing a means for system's developers to create pre-stored command sequences for specific functions to be carried out by the end user.

The fourth concept was fulfilled by making all of the various functions of the commands intrinsic functions of the operating system. Some operating systems require a language product to be used for file management, peripheral input and output, etc. In OASIS all of these functions and more are part of the operating system. Language products merely access the operating system to perform specific functions. This implies easy programming and program versatility.

1.2 Components of OASIS

OASIS is composed of many programs but these programs can be divided into a few major categories:

Nucleus The SYSTEM.NUCLEUS is a large section of instructions consisting of general purpose subroutines that provide the "intelligence" and integration of the system as a whole. It is the SYSTEM.NUCLEUS that is "booted" into memory when the system is first turned on. ("Boot" is a term referring to the process of "picking yourself up by your bootlaces". In computers it indicates that there are a few instructions that provide the means of accessing more instructions, that in turn provide the means for more, etc.)

With the multi-user version of the OASIS operating system it is the NUCLEUS that controls the scheduling and system resource sharing of the multiple users.

Associated with the NUCLEUS (but physically separate) are device driver programs. These programs contain instructions designed to handle one specific peripheral device. The device driver programs are responsible for the actual acceptance and transmittal (input and output) of data to a device, error detection and error handling, and the initialization of the device.

CSI The Command String Interpreter is the hub of OASIS communications; it provides access to system and user programs, and performs input and output functions. The combination of the nucleus, device drivers and the CSI is the essence of the OASIS operating system. Through them all computations, input and output are performed.

Programs The various programs of the OASIS system provide the "sophisticated" means of processing information. It is sophisticated in the technical sense in that the various programs provide the intelligence to process information and data instead of just characters or "bytes". These programs consist of the system commands described in subsequent chapters of this manual and the language processors such as the MACRO assembler, Basic, EXEC, etc.

1.3 Program Development Features

Computer systems are often used extensively for program development. The programmer makes use of the programming "tools" available on his system to develop programs which will perform functions specific to his needs. The number and type of "tools" available on any given system depend on a good many factors - the size of the system, its application and its cost, to name a few. The OASIS system, however, provides several basic program development aids not commonly found on micro-computers: system editor, macro assembler, linkage editor, on-line debugger, and high level languages: BASIC and EXEC.

The editor is used to create and modify textual material. Text may be the lines of code which make up a source program written in some programming language, or it may be data; text may be reports, or memos, or in fact may consist of the text in this publication. (This manual and the other OASIS manuals were developed and printed using the editor and SCRIPT, an optional OASIS program.) In this respect using an editor is analogous to using a typewriter--the user sits at a keyboard and types text. But the advantages of an editor far exceed those of a typewriter because once text has been created, it can be modified, relocated, replaced, merged, or deleted--all by means of simple editing commands. When you are satisfied with the text you can save it in a disk file where it is available for later reference and use.

When the editor is used for the purpose of writing a source program, development does not stop with the creation of this program. Since the computer cannot understand any language but machine language (a sequence of discrete codes), an intermediary program is necessary which will convert source code into the instructions the computer can execute. This is the function of an assembler, compiler, or interpreter.

An assembler accepts alphanumeric representations of instructions, interprets the

code, and produces as output the appropriate object code. You can direct the assembler to generate a listing of both the source code and binary output, as well as more specific listings which are helpful during the program debugging process. The assembler is capable of detecting certain common coding errors and of issuing appropriate warnings.

In addition to the normal features of an assembler, the OASIS MACRO Assembler provides a wide range of directives, system calls (subroutines in the operating system accessible to user programs), and macro capability.

Source programs may be complete and functional by themselves; however, some programs are written in such a way that they must be used in conjunction with other programs in order to form a complete and logical flow of instructions. For this reason the object code produced by an assembler must be relocatable--that is, assignment of memory locations must be deferred until the code is combined with all other necessary object modules. It is the purpose of the linkage editor to perform this relocation.

Very rarely is a program created which does not contain at least one unintentional error (bug). Often it is not until execution that the user discovers that a program is not working properly. Programming errors may be extremely difficult to find, and for this reason a debugging tool is usually available to aid the programmer in determining the cause of error.

The OASIS debugging program allows the user to interactively control the execution of the program. With it, you can examine the contents of individual locations, set designated stopping points during execution (break points), change the contents of locations, continue execution, dis-assemble portions of the program, and add new instructions using mnemonics instead of machine code.

A high level language, such as the OASIS BASIC interpreter/compiler, provides an alternate means of writing a source program other than assembly language mnemonics. High level languages are easy to learn--a single command may cause the computer to perform thousands of machine language instructions. In a high level language you do not need to know about the mechanics of the computer. In addition, the OASIS BASIC compiler offers a special immediate or command mode which allows you to solve equations and formulas, or to interactively debug the program, similar to the debugger for assembly language programs.

CHAPTER 2

SYSTEM COMMUNICATION

The Command String Interpreter is the hub of OASIS system communications--it provides access to system and user programs, and contains the Executive Procedure Processor. The various system programs (utilities) accessible through the CSI are discussed in subsequent chapters of this manual. Major utilities, such as EDIT, EXEC, MACRO, etc., are described in separate manuals.

2.1 System Start Procedure

To load the OASIS operating system and start the CSI the following procedure must be followed.

1. Turn the power on. If the power is already on then use the reset button, if available.

2. Load an OASIS system disk in drive 1 and start the drive.

3. After a short time the following message will be displayed on the console:

   Single-user OASIS version v.r - nnK

   or

   Multi-user OASIS version v.r - nnK

   The v.r will be replaced by the current version and release number. The nn will be replaced by the amount of memory on your particular machine. If the message is not displayed after a reasonable period of time (5-10 seconds), unload the disk and reload it, making sure it is being loaded in the proper direction, etc.

4. The system will then display:

   Time (HH:MM:SS)

   At this point enter the correct time of day, preferrably in 24 hour format. It is not necessary to enter the seconds or any leading zeros.

   You may enter a carriage return only instead of the current time. When this is done the system will use the time of the last system start-up.

   Note: not all systems on which OASIS is implemented support this time of day feature. If your system does not support this feature this question will not be asked.

5. The system will then display:

   Date (MM/DD/YY)

   Enter the correct date. It is not necessary to enter leading zeros. The date entered must be a valid date, and must be greater than 01/01/77. You may enter the date using the month name such as:

Date (MM/DD/YY) 15 April 1978

or in some other reasonable format.

You may enter a carriage return only instead of the current date. When this is done the system will use the same date as that used at the last system start-up. It is permissable to omit the year number. When this is done OASIS will use the year number of the last start up.

6. After a valid time and date have been entered the system displays the current time and date along with month name and day of week. Then the system locates the NUCLEUS, CSI, EXECSAVE, and ERRMSG files on the system disk, loads the NUCLEUS and CSI, finds and loads all of the ATTACHed and SYSGENed device drivers, and passes control to the CSI. If any of these files cannot be found on the system disk an error message is displayed.

   The location of these modules by the "boot" process saves considerable time during processing because no searches need be made. The tradeoff to this advantage is that the system disk may not be changed without re-locating these files.

When the system is brought up in the manner described above the system attempts to automatically log onto the IPL account. In this automatic logon procedure a search is made of the file SYSTEM.ACCOUNT for the account IPL. When this account is found in the file the system logs onto the account. As discussed in the chapter on User Accounting, when an account is logged onto, a search is made of all attached disks for a file named: IPL.EXEC accessible by the account IPL (public or private access). This is the file name of the EXEC language program to be executed upon system logon. When the file is found control of the system is transferred to it. When the file cannot be found the CSI retains control and the prompt character is displayed.

If no logon start up program is desired you should create an empty file named <account name>.EXEC on the system disk. This will speed up logon start time because an exhaustive search is not made.

For details on using the EXEC language refer to the EXEC Language Reference Manual.

When the search is made of the SYSTEM.ACCOUNT file and no IPL account is found, the system will automatically execute the LOGON command. This command will ask you to enter the name of the account that you wish to log onto. For complete details of this command refer to the chapter on the LOGON Command.

To simulate an operating environment without user accounting use the KILL option of the ACCOUNT command. This will delete the SYSTEM.ACCOUNT file. When no SYSTEM.ACCOUNT file is found the system reverts to a non-accounting environment.

OASIS is distributed with no SYSTEM.ACCOUNT file defined.

An example system start up dialogue, single user system, without a SYSTEM.ACCOUNT file defined:

```
Single-user OASIS version 5.4 - 48K

Time (HH:MM:SS) 1159

Date (MM/DD/YY) 101079

11:59:02 Wednesday, October 10, 1979

>
```

2.2 Dynamic Input and Output

As mentioned earlier, the NUCLEUS contains the instructions to handle peripheral input and output devices. These groups of instructions are called device drivers. In this manual these drivers are referred to as physical device drivers. Physical device drivers are programmed to meet the specific requirements of a specific device. An example of a physical device driver is the program containing the instructions necessary to handle a serial input/output port (SIO). The driver for the SIO has to have instructions capable of performing input or output on a bit by bit basis at a specified rate (baud).

To make programming and operation of the system more versatile the OASIS operating system communicates with physical device drivers by means of a logical device name. The logical device name is nothing more than a linkage that points to a specific device driver. By using logical device names instead of the physical device numbers (physical device drivers are accessed by the number of the driver - refer to the appendix) it is very easy to change the input or output device of a program. All that is necessary is to change the number of the device driver that the logical name is pointing to.

Only physical device drivers are capable of communicating with the "real" world outside of the computer. The connection between the logical device driver and the physical driver is made by using the ATTACH program.

By making the connection between logical and physical device drivers dynamic, or changeable, you gain a great deal in versatility. Programs are created with specifications such as printing to the printer or console. With no changes required in the program the output can be changed from a parallel interface line printer to a serial interface line printer, or from a serial interface line printer to a video display monitor, etc.

2.3 Configuring Your System

The ability to dynamically associate physical and logical device drivers would be a nuisance if it had to be done every time the system was turned on. Also it would be inconvenient if you had to set the status of all the internal switches in OASIS every time (see the SET command).

To avoid this inconvenience OASIS provides the SYSGEN program. With it you can "permanently" set the attached devices and switches. When the SYSGEN program is executed the current linkages of attached devices are saved in the disk image of the system nucleus. The next time the system is turned on the attachments and sets will already be in effect. (See the ATTACH, SET, and SYSGEN commands.)

CHAPTER 3

USER ACCOUNTING

The OASIS Operating System supports user accounting and restricted file access between user accounts. User accounting means more than the mere record keeping of the time that each person uses the system. In OASIS it also means that a user may only access the public (common) files and programs, his own files and programs (private files), and those select files and programs of other users who have given him explicit permission to access them (shared files). Similarly a user's files and programs may not be accessed by any other user without his explicit permission. Additional accounting information is generated for system restarts.

3.1 File Ownership

User accounting utilizes the concept of file ownership. In a user accounting system each file is "owned" by one account and only one account. Files owned by a user account are referred to as "private" files. Private files may only be deleted or changed by the owner of the file. Under certain conditions a user account may have access to another user's private files. When this is true those files owned by the other account are referred to as "shared" files to the non-owner of the file. File sharing is strictly controlled by the operating system. A shared file may be changed by the non-owner but may not be deleted.

A special type of shared file is the "public" file. A public file is a file owned by the system account and all users have access to them without the necessity of specifying them as shared files. Usually the system programs are kept in the system account so that all users may utilize them.

The above ideas may best be clarified with an example. Suppose that a system has the following accounts and files:

Account	File
System	BASIC.COMMAND
	RUN.COMMAND
	EDIT.COMMAND
	ERASE.COMMAND
	RENAME.COMMAND
	FILELIST.COMMAND
RALPH	LEDGER.MASTER
	LEDGER.DETAIL
	LEDGER.HISTORY
	LEDGER.BASIC
LINDA	PAYROLL.MASTER
	PAYROLL.DETAIL
	PAYROLL.HISTORY
	PAYROLL.BASIC

If no files are designated as being shared files then RALPH can use the programs: BASIC, RUN, EDIT, ERASE, RENAME, and FILELIST (public files); he may execute the BASIC program LEDGER, and he may change or delete the data files: LEDGER.MASTER, LEDGER.DETAIL, and LEDGER.HISTORY (private files). When RALPH is logged on the system he would not even know about LINDA's files nor could he access them in any way.

LINDA can use the programs: BASIC, RUN, EDIT, ERASE, RENAME, and FILELIST too. However she may only execute the BASIC program PAYROLL and she may only change or delete the data files: PAYROLL.MASTER, PAYROL.DETAIL, and PAYROLL.HISTORY. Similarly when LINDA is logged on the system she would not know about RALPH's files nor could she access them in any way.

When the system account is logged onto (usually only done for system maintenance purposes) neither RALPH's files nor LINDA's files could be accessed in any way. In this case there are no public files and the system account cannot have shared files.

Assuming that RALPH's program and files perform General Ledger accounting functions and that LINDA's program and files perform Payroll processing it would probably be best if RALPH specified that his files: LEDGER.MASTER and LEDGER.DETAIL could be shared by LINDA. If this were done (using the SHARE command) then LINDA could create General Ledger entries reflecting her Payroll processing in RALPH's files. (LINDA could not delete these files--only the owner of a file can delete it.)

Say that Ralph gets fired and Linda is promoted to do both Payroll processing and General Ledger accounting. Before removing the account RALPH from the system (using the ACCOUNT command) his files should be transferred to the account LINDA. When this is done (using the OWNERCHG command) the LINDA account would own her original four files plus the four LEDGER files previously owned by the RALPH account.

IMPORTANT! Extreme care should be taken when deleting an account name from the system. Before deleting the account all shared access to any of the account's files should be removed (using SHARE command) and then any of the account's files to be kept should be transferred to another account. Any files that are not to be used again should be deleted.

If the above procedure is not done before an account name is deleted YOU MAY NOT EVER BE ABLE TO ACCESS THOSE FILES AGAIN. Re-adding a deleted account does not necessarily make the deleted account's files accessible.

3.2 Adding Accounts

As mentioned in the above example there are special programs that perform the user accounting features. When an OASIS system is first received it has no accounts. To start using the accounting feature you will have to add accounts with the ACCOUNT command. The ACCOUNT command allows new account names to be added and allows maintenance (changing and deleting) of existing accounts.

Before any new accounts are added to the system the user should determine whether or not the accounting feature is to be kept as a part of the system (it can be disabled but not easily re-enabled).

3.3 Privelege Levels

When setting up new accounts (or changing existing ones) the question of privilege level is asked. The privilege level of an account determines which commands may be executed by the account. Each system command has its own privilege value associated with it. These privilege values may be changed by the user with the CHANGE command to conform to his own requirements.

In the previous example assume that RALPH has a privilege level of 0 and that LINDA

has a privilege level of 1 and that the privilege values of the commands are unchanged from the distribution values (see Appendix "Privilege Levels"). In this case RALPH could only execute the commands RUN and FILELIST. His privilege level is not high enough to execute the other commands. This essentially restricts him from changing or adding any files in his account except under the control of his BASIC program (he has to use RUN to execute that program and RUN does not allow the operator to make changes in the program.)

LINDA, on the other hand, can execute all of the commands that are in the system account in the example. Therefore she, unlike the RALPH account, can erase files, rename files, edit files and use BASIC to debug and/or make changes in her program. She still would not be allowed to erase or rename any of RALPH's files even if she had shared access to them (also the EDIT command would not allow her to change any of RALPH's files because it erases and renames a file in the process of updating it).

It is best to leave the system account (or a synonym of it) at the highest privilege level available. This would allow anybody that had permission to log onto the system account to execute all of the commands in the system.

3.4 Account Synonyms

The word synonym was used in the last section. A synonym of an account is an account that is identical to the account it is a synonym to but has a different name, and possibly a different password and/or privilege level. In the example where RALPH has a privilege level of 0 the question of file maintenance arises: How can RALPH's program be changed? One way would be to set up another account that has shared access with RALPH's program file, copy the file from RALPH's account into the new account (giving it a unique name), performing the required maintenance, establishing it as a shared file with the RALPH account. Now the RALPH account can use the new version of the program. Obviously this is rather tedious.

A better way would be to establish a new account that is a synonym to the RALPH account that also has a privilege level of at least 1. When this new account is logged onto (say it was named MICHELLE) the account will have the the same file access as the RALPH account but the privilege level will be 1. This privilege level allows the execution of the EDIT command. Any changes made while MICHELLE is logged on will be made to the RALPH files.

Synonyms have another important use. Even if the password and privilege level are the same between an account and its synonym, the account name will be different. This means that any accounting history generated by the two accounts will be distinguishable.

3.5 Account Passwords

Another question asked when an account is added or changed is the account password. All accounts have a potential password associated with them. When a password is defined for an account it must be used each time an operator attempts to log onto the account. If the operator does not know the password he or she will not be able to log onto that account. This implies that the operator will not be able to access any of the files or programs that are owned by that account and not shared with other accounts.

When the owner of an account (or anyone else who knows the password) logs onto an

account and enters the password the password is not displayed as it is entered. This prevents others who don't already know the password from inadvertantly finding out. The password provides a reasonable level of file and account security.

3.6 Accounting History

As stated at the beginning of this chapter, user accounting implies the collection of user account accounting information. In the OASIS system a history record is generated every time a user logs onto or off of an account. This history record contains the information answering the questions: Who logged onto the system? When did they log onto the system? and How long were they logged onto the system? Additional information is recorded showing when the system was booted, when a disk was backed up and to where.

This account history information is kept in the special file named SYSTEM.HISTORY. This file is special in the sense that no account owns it. This history file, similar to the account name file SYSTEM.ACCOUNT, cannot be accessed by any user. The SYSTEM.HISTORY file (and later the ACCOUNT.HISTORY file) contains records accounting for each system start up, each user LOGON and LOGOFF.

In order to access the accounting information kept in the history file you must use the option CLEAR in the ACCOUNT command. This option will transfer the data in the SYSTEM.HISTORY file to the system account's file ACCOUNT.HISTORY. Once this has been done the information is accessible to all users as the file is public. This procedure should be done periodically anyway as the SYSTEM.HISTORY file can grow to use up all the space on your system disk.

When the SYSTEM.HISTORY file does grow to use all of the space available on the system disk the program that was trying to add a record to the history file will inform you that it has used all the space available and that you should use the ACCOUNT command to clear the file. When this is done there still won't be any space available but you will have access to the ACCOUNT.HISTORY file and can either erase it or copy it to another disk and erase the file on the system disk. If you do not make space available for the SYSTEM.HISTORY file no more accounting records will be added but you will be allowed to log on and off accounts.

This accounting history feature is optional and may be enabled or disabled by the user (see the SET command). OASIS, as distributed, has this history feature disabled.

CHAPTER 4

MULTI-USER OASIS

OASIS, versions 5.4 and above, is available in single and multi-user versions. A multi-user operating system is a system that allows more than one user (console) to be controlling the computer at one time. Each user in a multi-user system operates as if he had the complete attention of the computer to perform his task.

In multi-user OASIS each user is given a portion of memory to have exclusively. Additionally, each user has access to the single copy of the operating system in memory and any re-entrant programs that may be available. A re-entrant program is a program that can be utilized by more than one user at a time, without having multiple copies of the program in memory. OASIS BASIC is a program that is available as a re-entrant command.

Although a computer can only perform one single operation at a time it operates a such a high speed that it can service several users at one time by switching between users, giving a portion of time (a time slice) to each user. When the time slice is used up the system suspends the operation of that user and allows another user to have a time slice. When that user has exhausted his time slice the next user is selected and so on. Because a computer operates at very high speeds this sharing of time is generally transparent to the user.

A user may lose his time slice if, for any reason, he is waiting for an input or output process to complete.

Certain problems may arise when using a multi-user operating system. Specifically, the situation may arise that two users try to update a disk file. Both users may have read in the same copy of a record, made changes to the record without the knowledge of the other user, and then written the changed record back. The record change written back last will be the change that is actually made to the file. This could be disastrous if the record was a balance record for example.

In multi-user OASIS this situation can be easily prevented in one of two ways: the first user may put a lock on the entire file, thus preventing the second user from making any changes to the file; or both users may put a lock on the record--the user who puts the lock on the record first will prevent the second user from reading the record until the lock is released, thus insuring that the second change to the record will be made to the correct copy of the record.

Multi-user OASIS requires that the user accounting feature be used and that each user be logged onto a different account.

Multi-user OASIS also provides other convenient features: transmitting messages from user to user (MSG), forcing a user to execute a command (FORCE), peeking at another user's console output through your own console (PEEK), scheduling jobs for future execution (SCHEDULE), display status of other users (SHOW MEMORY). Of course all of the single user programs and features are available too, including: user accounting; private, public, and shared file ownership; optional printer spooler; complete program development package--MACRO Assembler, debugger, linkage editor, BASIC compiler and interpreter; system diagnostic programs--disk verify, memory test; etc.

4.1 Invoking Multi-user OASIS

When multi-user OASIS is first booted into memory it is set up in single-user mode--only one user is defined. At this time the system's manager should define

and initialize the users to be utilized. This is accomplished with the SET MEMORY command and the START command. As each user is started the respective consoles will request that the operators logon to an account. That is all that is necessary to use multi-user OASIS. This process of invoking multi-user mode may be automatically performed if the system's manager defines an account named IPL with an IPL.EXEC program that performs the necessary commands to set up the system.

After some usage of the system the system's manager may decide that the time slice value (the amount of time allocated to each user partition) is not proper for the applications being used. This is changeable with the SET SLICE command.

A typical multi-user start up would look like:

```
Multi-user OASIS version 5.4 - 64K

Time (HH:MM:SS) 830

Date (MM/DD/YY) 1030

08:30:00 Tuesday, October 30, 1979

Logon Please: IPL

>ATTACH A DISK2 (PUBLIC

>LOAD BASIC

>SET MEMORY 2 10240

>START 2 (SIO2 B19200 C4 FF6

>SHOW MEMORY

Port MEMLO MEMMX MEMHI SP--- PC--- Size  Routine Username
     0000H 4CFFH                   19712 NUCLEUS
     4D00H A886H                   23431 BASIC
  1* D087H FDFFH FB21H FAF3H D087H 11641 SHOW    IPL
  2  A887H D086H CDA9H CD87H CDCFH 10240 LOGON

>LOGOFF

Logon Please:
```

The above start up procedure will result in disk S and disk A being public (all users may access), two users, and re-entrant BASIC available to both users. As can be seen, this start up procedure is kept in the IPL.EXEC file in the account IPL. All entries after the date and before the LOGOFF were made by the system or the EXEC, nothing special had to be done by the operator.

Your start up procedure might be different in the number of users, public disks and whether or not the re-entrant BASIC is loaded.

4.2 Single-user Mode

When only one user is defined and active in the system the system is said to be in single-user mode. This mode is important because some tasks may only be performed

in single-user mode, such as full disk backups, loading and unloading re-entrant programs and detaching public disk volumes.

In the above start up procedure, note that the public disk and the re-entrant BASIC are both attached or loaded before establishing a second user to the system. This was done because these operations are required to be performed in single-user mode.

CHAPTER 5

OASIS FILE MANAGEMENT

The OASIS Operating System is essentially a file management system. The operating system consists of programs that manipulate or maintain files on the random access, mass storage device (disk drives). Each disk may contain a mixture of data and program files.

A file is a logical group of records, records being the individual elements. An example is a customer name and address file. The file is a group of customers; the records are the individual customer names and addresses.

The operating system has the ability to handle up to eight disk drives, if available. However, each disk is maintained independently of others. This means that a file may not start on one disk and continue on another. A file must be completely contained on one disk.

Each OASIS disk must be initialized before any files may be written to it. The initialization process is accomplished by using the program INITDISK with the FORMAT option. This program need only be executed once for a disk. However, there is no restriction on re-initializing a disk. The initialization process is necessary because a new disk, as received from the manufacturer, has no or wrong information written on it. The initialization process writes information on the disk necessary for OASIS to add files to the disk. The information written is a blank directory, a disk label, an allocation map indicating that the entire disk is available for new files, and data blocks. Data blocks are 1K blocks or sections of storage. 1K is a symbol representing 1024 bytes or characters.

The directory is an index which contains the contents of the disk and pointers to the data areas which constitute the files on the disk.

After a disk has been initialized, the other programs in the operating system can use it for adding files, changing files, erasing files, etc.

5.1 Directory Naming

The files on a disk are accessed by specifying the file name, file type and the disk. The disk is specified by the disk label (established in the INITDISK program) or a one character directory label (established by the ATTACH program). This directory label is really just an indication of which disk drive the disk is in and only has meaning while the disk is in that drive.

The directory label is one of the following characters: S,A,B,C,D,E,F,G. The S label is reserved to indicate the system disk and may not be attached to any other disk. (A system disk is the disk that contains the operating system programs and is the disk that the system was "booted" from at system start up time.)

A directory label may be attached to only one drive at a time, and a drive may have only one label attached to it.

5.2 Data and File Formats

The OASIS system makes use of five file formats: ASCII sequential, indexed, direct, absolute, and relocatable.

5.2.1 ASCII Sequential Files

Files in ASCII format conform to the American National Standard Code for Informaton Interchange, in which each character is represented by an 8-bit code. Files in ASCII format include program source files created by the Editor, listing files from the MACRO assembler, object files from the MACRO assembler, and data files from user programs.

An ASCII file, when accessed by some of the system programs, such as the Editor or BASIC, is read into memory as a whole. Other system programs and user programs usually access this type of a file one record at a time. The reason it is called an ASCII sequential file is that a record cannot be accessed without accessing all records prior to it in a sequential manner.

This type of a file is the only format that supports variable length records. The length of a record is determined by the contents of that record, not by the access method.

A sequential file cannot be updated in place. The only form of update that may be performed on a sequential file is appending records to the end of the file. Of course the file can be erased as a whole.

The size of a sequential file is determined dynamically. When the file is first created it is allocated the minimum size: 1K bytes. When a record is added to the file that will not fit in the current allocation one more block of storage is allocated to the file and the record is written to the new area. This is the only file format that does not require contiguous data space. Each block of data may be anywhere on the disk with each block of data having a pointer to the block that logically follows.

5.2.2 Direct File Format

A direct file contains binary data with fixed length records and fixed number of records. This format is used by user programs, generally for master type data.

A direct file is accessed randomly by relative record number. This means that any record may be accessed independently of all other records in the file. To access a specific record the only information required is its record number, relative to the start of the file.

The size and record length of a direct file must be known when the file is first created. A direct file is created by using the CREATE program.

5.2.3 Indexed File Format

OASIS supports a single key, hashed, sequential access file format called indexed. Similar to the direct file, an indexed file contains fixed length records, fixed allocation file size.

An indexed file is accessed randomly or sequentially. The only information necessary to access a specific record is the key that it was written with. Normally this key is logically associated with the record such as a customer name for a customer record, part name for an inventory record, etc. The key may be up to 128 characters in length but the maximum length must be known when the file is created with the CREATE program.

The combination of keyed access and sequential access allows you to simulate "approximate or generic key match" access. Approximate match means that the key field of the record accessed may be equal to or greater than the specified key. Using an index, it is often desirable to access the record that contains the key value requested, or the record with the next highest key value if the requested key value doesn't exist in the file. This allows a user program to retrieve records without knowing an exact key and without repeated access operations when the system does not have the record specified by the key.

An indexed file can be read sequentially in key sequence. This makes this file format very useful for many applications in which the information is updated randomly but is required to be listed in sequence.

An indexed file is always maintained by the system in sequence. There is no "overflow" area to be sorted periodically.

5.2.4 Absolute File Format

Absolute files are generated by the system linking loader and are restricted to machine code programs. These files contain programs in a "memory image" format which is a "picture" of what memory will look like when a program is loaded. The file itself requires the same amount of disk storage as the corresponding number of 256-character memory blocks.

5.2.5 Relocatable File Format

Relocatable and self-relocatable files are also generated by the system linking loader and are restricted to machine code programs. A relocatable file is origined at address zero. When the program is run (by entering the program name at the CSI level), the file is relocated as it is loaded into memory. (An absolute file requires no such relocation.) A self-relocating program is also origined at zero. The programmer has the responsibility of using a special system subroutine before each reference to an absolute address. This system subroutine (referred to as a system call) adjusts the address reference.

5.3 File and Disk Protection

In many applications for computers it is desirable that there be some means of preventing the inadvertent alteration of stored information. The user may wish to prevent programming errors from changing master files or he may wish to prevent operator errors. The OASIS file management system provides this means.

Any specific data or program file may be protected from erasure, alteration, or both. This protection is specified to the system by using the RENAME program. By using this program you can delete protect, write protect or read protect a file or group of files.

When a file has delete protection the system will not allow any program to erase the file.

When a file has write protection the system will not allow any program to make changes to existing records in the file or to add new records to the file. A file that is write protected is not necessarily delete protected.

When a file has read protection the system will not allow any program to access the file (writing to a file requires that part of the file first be read), with the

exception of the RUN command. This protection is intended to be used to protect a BASIC interpretive program file.

In addition to this specific protection the system provides a means of protecting an entire disk of information from being altered in any way. To write protect an entire disk the INITDISK program is used.

When a disk is write protected no file on the disk may be deleted or changed by any program.

5.4 Program Protection

System programs may be protected from unauthorized execution by means of a privileged access method. All of the system programs have a privilege value associated with them. The system has a privilege level associated with it. Any attempt to execute a program whose privilege value is greater than the current system privilege level will be ignored. The operating system is distributed with the highest privilege level available (5), thus allowing access to all of the distributed programs. The privilege level of the system may be lowered, thus restricting access to programs with a privilege value higher than the value that is set.

The privilege values for the system and the system programs may be changed by using the CHANGE command. This allows you to set your own levels of access. The privilege level of the system is a SYSGENable function, allowing you to "permanently" set the level of access. The appendix on Privilege Levels defines the methods of changing the privilege levels and values and defines the values of the programs, as distributed.

5.5 Multi-user File and Record Locking

A feature available with multi-user OASIS is entire file locking or individual record locking. When a file or record is locked by one user other users are denied access to that file or record until the user that has locked the file or record unlocks it.

This feature is very necessary for the prevention of bad data. Without file or record locking user one would be able to read a record, user two could also read that same record, change it and write it back without user one knowing that the record he has is no longer current.

File and record locking is easy for the user to specify--see the OASIS BASIC Language Reference Manual and the OASIS MACRO Assembler Language Reference Manual.

Caution: Deadlocking may occur if care is not taken when programming an application in a multi-user environment. A deadlock is when a user's processing is suspended for an excessive time due to the careless locking of a record or file by another user partition.

CHAPTER 6

SYSTEM CONVENTIONS

OASIS system conventions such as file naming procedures, prompting characters, etc. are described in this chapter. You should be familiar with these conventions before running the system.

6.1 Directory Search Criteria

When OASIS is capable of using two or more file directories, a problem of search order arises. If, in a specific program, you specify which directory to examine for the file, there is no doubt as to where and how to search directories. In some cases you do not or cannot designate a directory. In this situation there must be a set search sequence for scanning directories to find a specified data file or available disk space.

Standard Search Order

To find a data file or user program (not a user written command), user disks are searched in the system-defined alphabetical order: A, B, C, D, E, F, G, S.

To find a command or EXEC program the system disk is searched first and then the user disks: A, B, C, D, E, F, G.

To find space for creating a new data file the system disk is searched first and then any user disks.

6.2 Prompting Characters

The OASIS operating system uses different prompting characters for each of the programs that accept input from the keyboard. This provides an easy means of determining which program is asking for input. The following table summarizes the characters displayed by each program to indicate to the user that the system is awaiting user response.

> The Command String Interpreter is waiting for a command.

* TEXTEDIT or EDIT is waiting for a command.

\- BASIC is waiting for a command.

? A BASIC program is waiting for input. This prompt can be changed by the user program.

: An EXEC program is waiting for input.

= The system DEBUG program is waiting for a command.

^ A system program has displayed a full screen of data and is awaiting a response to indicate that it may proceed to the next screen.

\ The PATCH program is waiting for a command.

) The COPYFILE program is waiting for input to the translation list.

\# The LINK program is waiting for input.

6.3 File Naming Conventions

Each file created by the system or by a user program is referenced by a file description. The description consists of three elements in the following format:

filename.filetype:filedisk

filename Is a one to eight character alphameric name assigned (in most cases) by the user. The first character of a file name must be a letter. The name may only contain letters, digits, or a dollar sign.

filetype Is a one to eight character alphameric name used as a descriptor or qualifier of the filename. Certain filetypes have special meaning to OASIS and should be used accordingly. The first character of the file type must be a letter and only letters, digits, or dollar signs may be used for the other characters. Imbedded spaces are not allowed in file names or types.

filedisk This may be either the directory label of the disk or the disk label of the disk on which the file resides.

Most of the programs in OASIS allow the file disk specification to be omitted. In this situation the system will search the disks for the file, stopping on the first occurrence of the file name and type.

The OASIS operating system has several reserved file descriptions:

IPL.EXEC	- system start-up procedure.
SYSTEM.NUCLEUS:S	- the "core" of the operating system.
SYSTEM.CSI:S	- the Command String Interpreter.
SYSTEM.EXECSAVE:S	- working storage for executive procedures.
SYSTEM.DEVNAMES:S	- physical device names and attributes.
SYSTEM.ACCOUNT:S	- user accounting name, passwords, and privilege levels.
SYSTEM.HISTORY:S	- user accounting history, etc.
SYSTEM.ERRMSG:S	- error numbers and messages.
SYSTEM.EXEC:S	- EXEC language processor.
SYSTEM.SPOOLER:S	- print spooler program.
SYSTEM.SPOOL$$n	- spooled print files.
SYSTEM.DEVnn:S	- Device drivers.
SYSTEM.CLASSnn:S	- Terminal class code translators.

These file desciptions should not be used for any function other than as provided when the operating system is distributed.

Additionally, there are several file types that have special or default uses by various OASIS commands:

COMMAND	- Required by all commands that are executed by CSI.
OBJECT	- Required by the LINK command for its input files.
FORTRAN	- Default file type for a FORTRAN language processor.
COBOL	- Default file type for a COBOL language processor.
PASCAL	- Default file type for a PASCAL language processor.
BASIC	- Default file type for a ASCII, BASIC program.
BASICOBJ	- Required file type for the BASIC interpreter.
BASICCOM	- Required file type for the BASIC compiler.
BASICUSR	- Required file type for a BASIC assembly language

	subroutine.
LOADFILE	- Default file type for a re-entrant, loaded program.
SCRIPT	- Default file type for the SCRIPT processor.
ASSEMBLE	- Default file type for the MACRO assembler
SYNONYM	- Required file type for a command synonym table.
EXEC	- Required file type for the EXEC language processor.
LINK	- Default file type for a LINK control file.
COPY	- Default file type for the MACRO assembler, copy files.
MACRO	- Required file type for the MACRO assembler, macro definition.
BACKUP	- Required by various commands as a copy of its input file.

These file types must be used when indicated and should only be used for the above purposes.

The file description SYSTEM.WORK$$n is used by the system for work files. Avoid using this file type as your file may be erased by the system when it wishes to use a work file by the same name.

The file name of HELP should be not be used for data files or high level language program names as this name is reserved to indicate that a "help" message is to be displayed about the program.

6.4 Wildcard Specifications

Some of the commands allow you to specify more than one file at a time. For instance, the FILELIST command allows you to specify a list of all files on one disk, or all files with a specific name or type. In addition to these general features the system allows you to specify all files starting with a specific character or string of characters, etc.

The following definitions and examples illustrate the various ways that a file name or type may be specified. These features are only permissible when the command allows you to specify more than one file. This way of specifying a file description is referred to as "Wildcards".

Definition

*	Indicates that any characters from this position through the end of the field are acceptable.
?	Indicates that any character in this position is acceptable.
@	Indicates that any alphabetic character in this position is acceptable.
#	Indicates that any numeric character in this position is acceptable.
=	Indicates that the source file description for this element is to be used for this character position through the end of the field. This wildcard is only available in the COPYFILE and RENAME programs.

Examples:

```
* BASIC S
```

This file description indicates that all files whose type is BASIC on the S disk are to be included.

```
GLUP* B??????# S
```

This file description indicates that all files whose name starts with the characters GLUP, whose type starts with the character B, has a number in the eighth position and is on the S drive are to be included.

```
@@@@??#* ??  *
```

This file description indicates that all files should be included whose name is at least seven characters long, of which the first four characters are alphabetic, the next two may be any character, the seventh character is numeric and the eighth character may be anything, including a space. The type must be two alphameric characters only and the file may be on any disk.

6.5 Numeric Parameters

Some of the commands discussed in this manual allow certain parameters or options to be numeric. The system allows these elements to be entered using either numeric base ten (decimal) or base sixteen (hexadecimal). Decimal numbers are entered using only the digits 0-9. Hexadecimal numbers are entered using the digits 0-9 and characters A-F with the first character numeric and the number is terminated with an H. Examples of hexadecimal number entry are 14AFH and 0FFFFH.

6.6 HELP Function

When you first start using the system you will be unfamiliar with the command syntax of the various commands in the system. As an aid to the beginning user (and for those commands that are infrequently used) most of the OASIS commands have a HELP function available to them.

If, for whatever reason, you are unsure of the command syntax or options available enter the command name followed by a space and the word HELP. The command will display the following information:

- General function of the command.
- Syntax of the command.
- Options and functions available with the command.

HELP Example:

```
>ATTACH HELP

Function: To map a logical device to a
          physical device, or to un-
          attach a logical device, or
          list current attachments.

Syntax:  ATTACH log [phy [(options)]]

Where:
  log  -  CONSOLE,PRINTERn,READER
          PUNCH,COMMn,TAPEn, drive
  phy  -  Are the names defined in
          SYSTEM.DEVNAMES:S

Options:
  Lnnn -  line width for con,prt
  Pnnn -  page depth  "   "   "
  Bnnn -  baud rate for SIO
  LFnnn & FFnnn - delays for SIO
  PE & PO Parity Even or Odd for SIO
  ENAB    Auto enable (CTS/RTS) for SIO
  PP      maintain Page Parity for PRT
  Onnn    size of Overflow margin for PRT
  ALF     Auto Line Feed for PRT
  Hnnn    Head load delay (msec)
  Rnnn    Disk retry count
  STPnnn  Track step time (msec)
  STSnnn  Step settle time (msec)
```

6.7 Forms Alignment

The OASIS operating system always assumes that the forms in a listing device are at top of form and that the printing mechanism is at the left hand side of the page.

Whenever an OASIS program uses a listing device the program will always close the device with the forms at the top of form position and the printing mechanism at the left hand side. Any user written programs should also follow this convention for consistency.

6.8 Program Return Code

The system return code (displayed when RDYMSG set on) is a numeric value that, in general, indicates whether the previous program was executed to completion and whether it was successful. This code is very useful when programs are executed from an EXEC program or during testing of a new program..

There are seven specific return codes used by OASIS system programs. User written commands should conform to these return codes for consistency. (The return code is set by loading register A with the value immediately before returning control to the operating system.)

Program Return Codes

RC	Meaning
0	Normal job completion.
4	Warning errors encountered.
8	Syntax errors detected in command line.
12	Severe error detected (file not found, etc.)
16	Potentially disastrous error detected (disk/system integrity may be suspect).
253	Fatal, run-time error during RUN command.
254	Program canceled by operator using System Cancel-key.

CHAPTER 7

SYSTEM CONTROL KEYS

OASIS has twelve control keys defined for special use in controlling the system. The functions of the keys are such that you may wish to use them while a program is in operation. These keys may only be entered from the device attached as the console keyboard. The first seven of these keys will clear the type-ahead buffer.

7.1 System Restart-key

This key will cause the operating system to quit the current program in execution and cause the system to "re-boot". This key takes action immediately upon entry. Caution should be exercised in the use of this key as any open disk files that are being updated will not be closed. Any open files may then be logically invalid. The message "OK to IPL?" is displayed following the entry of this key. You must respond with a Y to actually perform the restart.

The System Restart-key is not available on multi-user OASIS as it could cause disastrous results.

7.2 System Cancel-key

This key will cause the operating system to quit the current program in execution and return control to the next higher level of program. Normally the next higher level is the operating system itself but in some cases a program may be executed from the Executive language. In this case control would return to the Executive procedure processor. When this key is entered a "clean" abort is performed: open file buffers are written to the disk file and files are closed.

Whenever this key is entered three of the toggle type keys will be reset: Printer Echo-key is set off; Console Echo-key is set on; Program Pause-key is set off.

7.3 Printer Echo-key

This key is a toggle type key, that is, entry of the key once causes the function to start, entry of the key again causes the function to stop. The function of this key is to cause all characters output to the console to be displayed on the primary listing device. When the system is first started the status of this key is "off".

7.4 Console Echo-key

This key is a toggle type key, similar to the Printer Echo-key. The function of this key is to cause all characters output to the console to be displayed or not displayed on the console. When the system is first started the status of this key is "on".

7.5 Program Pause-key

This key is a toggle type key, similar to the Printer Echo-key. The function of this key causes any program currently executing to "pause". Unless the program is accessing the disk, entry of this key will cause the program to stop whatever it is doing. When the Program Pause-key is entered again the program will continue from the point at which it was interrupted.

7.6 Program Cancel-key

The function of this key is only effective when an interactive language program is in execution. The interactive language programs include BASIC and EDIT (not RUN). When this key is entered the program will complete the execution of the current statement or sub-command and return to the command mode of that language.

7.7 Debug Break-key

Entry of this key causes a "break-point" immediately. It is only effective if the system debugger has been loaded. If the debugger is loaded the program counter will be displayed and control will be transferred to the DEBUG environment.

7.8 Line Cancel-key

Entry of this key causes the current line of input to be ignored. Most programs, upon recognizing this key, will erase the characters being ignored.

7.9 Character Delete-key

Entry of this key causes the last character entered from the keyboard to be deleted. For consoles whose class code is greater than zero this deletion is performed by physically erasing the character from the screen with a backspace, space, backspace sequence.

7.10 Console Display-fast

Entry of this key causes the display to the console ouanguage programs include BASIC and EDIT (not RUN). When this key is entered the program will complete the execution of the cuch as that distributed) is operating at the highest speed.

7.11 Console Display-slow

Entry of this key causes the display to the console output device to be slowed down

Entry of this key the first nine times will add one millisecond between the display of each character. Entry of this key the next ten times will add one hundreth of a second between the display of each charcter, etc.

The console output rate set by these two control keys is a SYSGENable function.

7.12 Console Screen Wait

This control key is a toggle command. Toggling this key on will cause the output to the console device to "wait" after each page is displayed by positioning to the bottom left-hand corner of the screen, displaying an up-arrow (^), and waiting for the operator to type a key. When the operator has typed a key the output to the Console continues with the next page.

Toggling this key off will disable this "wait" feature.

This is a SYSGENable function.

System Control Keys

Function	Key sequence
System Restart-key *	ESC,I *
System Cancel-key	ESC,Q
Printer Echo-key	ESC,P
Console Echo-key	ESC,O
Program Pause-key	ESC,S
Program Cancel-key	ESC,C
Debug Break-key	ESC,D
Line Cancel-key	CTRL/X
Character Delete-key	left-arrow
or	CTRL/H
or	DEL
or	RUB
Console Display-fast	ESC,A
Console Display-slow	ESC,B
Console Screen Wait	ESC,W

* Single-user OASIS only.

CHAPTER 8

DOCUMENTATION CONVENTIONS

The notations used to define the command syntax in this and associated manuals are defined in the following paragraphs.

Truncation and Abbreviation of Commands

When abbreviation of a command is permitted, the shortest acceptable version of the command is underscored. To determine the acceptable abbreviations refer to the chapter "SHOW Command", SYNONYM function. The example below shows the format specification for the COPYFILE command.

COPYFILE

This representation means that C, CO, COP, COPY, COPYF, COPYFI, COPYFIL, and COPYFILE are all valid specifications for this command name.

Command options are specified in the same manner. Where abbreviation is permitted, the shortest acceptable form of the option is underscored. If no minimum abbreviation is indicated, the entire word must be entered.

Special Documentation Characters

The following symbols are used to define the command format and should never be typed when the actual command is entered.

vertical bar	¦
brackets	[ ]
ellipsis	...

Additionally, angle brackets (<>) are used to enclose a term denoting that that term is to be replaced with one or more specific terms. For example: <fd> is replaced with the specific directory label or disk label such as--A, B, S, DATA1, etc.

Special Characters for OASIS

Uppercase letters and words, and the symbols listed below, should be typed as specified in the command format.

asterisk	*
comma	,
equal sign	=
parentheses	( )
period	.
colon	:

Other characters

Lowercase letters, words, and symbols that appear in the command format represent variables for which specific information should be substituted. For example, "file-desc" indicates that file identifiers such as "MYFILE.EXEC:A" should be entered.

Commands with Choices

Choices are represented in the command formats by stacking or separation with the vertical bar.

```
A                    ON|OFF
B
C
```

Brackets

The use of brackets denotes optional choices, one of which may be selected.

For example, the representation

```
[A]
[B]
[C]
```

indicates that the user may code A, B, or C, or he may omit the field entirely.

Ellipses

An ellipsis indicates that the preceding item or group of items may be repeated more than once in succession.

For example, the representation

```
(<option>...)
```

indicates that more than one option may be coded within the parenthesis.

Abbreviations Used

The following abbreviations are used throughout this manual and when used will always have the meanings defined here.

Abbreviation-Meaning

file-desc File description. Indicates that the file name, file type and file disk are to be inserted here.

fn File name.

ft File type.

fd File disk label. Indicates which disk contains the file. The disk may be identified by either the label that is ATTACHED to the directory or the label written on the disk by the INITDISK command. This parameter is normally optional.

ch Input/Output channel.

drv Disk drive number such as 1, 2, 3, ETC.

pid Partition identification number. Indicates the number of the memory partition associated with a user.

fr From. Usually used in a command that allows a from - to range to be specified.

n A lowercase n or group of n's indicates that the operator may type a number. This number may be entered in decimal or hexadecimal (see section on Numeric Parameters).

Naming Conventions

All command names, file names, file types and options must meet the following requirements:

a. All characters must be alphabetic, numeric or the dollar sign $.

b. The first character must be a letter character.

c. The length of each field may not exceed eight characters. If the field exceeds eight characters, the extra characters are truncated without any indication to the user.

Control Keys

When a reference is made to a control key or key sequence the key will always be defined in a consistent manner:

1. Keys that need to be entered in sequence will be separated by commas; for example: ESC,Q means that the escape key is typed followed by the 'Q' key.

2. Keys that need to be entered simultaneously will be separated by slashes; for example: CTRL/SHIFT/A means that the three keys control, shift, and 'A' must be typed simultaneously. To accomplish this it is easiest to type and hold the control key, and while the control key is still held down type and hold the shift key, and while both of these keys are held down type the 'A' key.

(This page intentionally left blank)

CHAPTER 9

THE COMMAND STRING INTERPRETER

9.1 The CSI Prompting Character

The Command String Interpreter (CSI) accepts input from the system keyboard and transfers control to the various command programs. The CSI is in control immediately after a system boot and its control is indicated by a prompting greater than character (>) at the left side of the screen.

9.2 CSI Format

An OASIS command, as seen by the CSI, consists of a command name, usually followed by one or more positional operands and, in some cases, by an option list. The general form for the command is:

```
command name  [operand ...] [(option ...[)]]
```

You must use the proper delimiters to separate the entries in the command line. These delimiters are - a period or space before a file type; a colon or space before a file disk; a left parenthesis before the first option; a comma or space before all other entries.

9.3 The Command Name

The command name is an alphabetic symbol of not more than eight characters. In general, the names are English verbs which describe the function you want the system to perform. For example you may wish to make a backup of a disk. In this case you would use the BACKUP command. Or you may want to list a file on the system printer. To do this, you would use the LIST command with the option PRINT.

9.4 The Command Operands

The command operands are keyword and variable symbols no longer than eight alphameric characters. These keywords and variables specify the information on which the system operates when it performs the command function. In the LIST command, you would use the file name and file type operands to specify the file to be listed.

Some commands require no operands, others require several. The command syntax and all operand requirements are specified in the following chapters.

You must enter the operands in the order in which they appear in the command formats, unless otherwise indicated.

9.5 The Command Options

The command options are keywords used to control the execution of the command. The command formats in the following chapters show all the options for each command.

The option list must be preceded by a left parenthesis - a closing parenthesis is not necessary.

If conflicting or duplicate options are specified, the last option entered is the option in effect for the command. Exceptions to this rule exist in the COPYFILE and INITDISK commands.

9.5.1 Default Options

To specify that you want the default options for a command, enter the command name and operand. It is not necessary to enter the left parenthesis unless you wish to include some of the non-default options.

9.5.2 Comments in the CSI

You may enter comments with OASIS commands by preceding the comment with the semi-colon character (;). The Command String Interpreter ignores all characters after the semi-colon, including the semi-colon.

9.6 CSI Program Search Sequence

The Command String Interpreter has a predetermined search sequence when it is given a command to search for. When you type a command the CSI will search for the command program in the following sequence:

1. Search the system disk for a program with the same name as that entered, and file type COMMAND.

2. If ABBREV is set ON then search the system abbreviation table for a match between the spelling of the command typed and the full command name. When a match is found step 1 is repeated.

3. If a user SYNONYM table has been defined then search the user synonym table for a match with the user synonym and, if ABBREV is also ON, the abbreviation specified in the table. If a match is found step 1 is repeated.

4. If IMPEXEC is ON then the system disk is searched for a file with the name as typed and a file type of EXEC. If a file is found the EXECutive procedure processor is invoked to interpret the commands in the EXEC file.

5. If ABBREV is set ON and a user SYNONYM table has been defined then step four is repeated using any matching abbreviations found in the file.

6. The above five steps are repeated for each of the attached disk drives, in ascending, alphabetic sequence.

7. If no command program or EXEC has been found the CSI will display an error message.

Once a program file has been identified and loaded into memory, control of the system is transferred to this program.

9.7 CSI Functional Summary

The following tables present a functional summary of the commands available to the OASIS user. They are described in detail in the following chapters.

Commands for Disk Maintenance

Command	Function
BACKUP	Copy an entire disk.
INITDISK	Initialize a disk for use by OASIS.
RECOVER	(Optional) Find and restore a file to it's directory.
REPAIR	(Optional) Detect and correct disk mis-allocation.

Commands for File Maintenance and Control

Command	Function
ASSIGN	Make I/O channel assignments.
ATTACH	Logically connect a device for future access.
COPYFILE	Copy one disk file to another disk file.
CREATE	Create a new file with direct or indexed format.
DUMPDISK	(Optional) Display the physical contents of file is hexadecimal and ASCII.
EDIT	(Optional) Create or update a file from terminal input.
ERASE	Delete an existing file.
FILELIST	List the names and attributes of the files in a directory.
GETFILE	(Optional) Transfer a sequential file from a CP/M compatible disk. (CP/M is a registered trade mark of Digital Research.)
KILL	(Optional) Remove a file from the directory without deallocating its space.
LIST	Type the contents of a file on a console or printer.
MOUNT	Allow a disk to be changed.
PEEK	Peek at another users console output.
RENAME	Change the name, type, or protection status of a file.
SECTOR	(Optional) Display sector numbers used of a sequential file.
STATE	Determine the existence of a file.
TEXTEDIT	Create or update a file from terminal input.

Commands to Develop and Maintain Programs

Command	Function
BASIC	(Optional) Create or update a BASIC source program.
DEBUG	(Optional) Dynamic debugger for machine language programs.
EDIT	(Optional) Create or update a source program.
FILT8080	(Optional) Translate Intel assembly mnemonics to Zilog mnemonics, compatible with the OASIS MACRO Assembler.
INTELHEX	(Optional) Convert Intel object code to OASIS object code.
LINK	(Optional) Translate with editing, an object program file to a load image (COMMAND) file.
MACRO	(Optional) Translate an assembly language source program to an object program.
PATCH	Correct minor problem in a load image file (COMMAND) program.
RELOCATE	(Optional) Create a relocatable object file from two absolute object files.
TEXTEDIT	Create or update a text file.

Commands to Change OASIS Parameters

Command	Function
ACCOUNT	Maintain user account names and attributes.
CHANGE	Change the privelege value of a program.
LOAD	Load a re-entrant command for subsequent usage.
OWNERCHG	Transfer ownership of a file to another account.
SET	Enable/disable various system switches.
SHARE	Enable/disable shared access to file by other accounts.
SHOW	Display the status of various system switches and parameters.
SPOOLER	(Optional) Change or display status of the printer spooler.
START	Initialize a user partition and allow user to logon.
STOP	Deinitialize a user partition.
SYSGEN	Save status of OASIS parameters on disk.
UNLOAD	Remove a re-entrant command from memory.

Commands for System Diagnostics

Command	Function
MEMTEST	(Optional) Diagnose memory.
VERIFY	(Optional) Test readability of a disk.

Commands to Execute Programs

Command	Function
BASIC	(Optional) Execute a BASIC program.
EXEC	Execute an EXEC program.
FORCE	Cause another user partition to execute a command.
RUN	(Optional) Execute a BASIC compiled program.
SCHEDULE	Execute a command at a later date and time.

Commands for System Communications

Command	Function
MAILBOX	Retrieve message sent by other users.
MSG	Send a message to another user(s).
RECEIVE	(Optional) Accept and save a file from another system.
SEND	(Optional) Send a file to another system.
TERMINAL	(Optional) Cause system to act like a terminal to another foreign system.

CHAPTER 10

ACCOUNT COMMAND

The ACCOUNT command provides the ability to maintain the SYSTEM.ACCOUNT file. This file is special in the sense that it doesn't belong to any account and can only be maintained by this command. The format of the ACCOUNT command is:

ACCOUNT [(option[)]]

As is seen, this command format is different from other command formats in that no file description is allowed.

IMPORTANT: USE OF THIS COMMAND IMPLIES CERTAIN RESPONSIBILITIES. BE SURE THAT YOU UNDERSTAND THE CONSEQUENCES OF DELETING, ADDING, OR CHANGING ACCOUNTS BEFORE USING THIS COMMAND.

ACCOUNT Options

TYPE Indicates that the current SYSTEM.ACCOUNT file is to be displayed on the console.

PRINTER[n] Indicates that the current SYSTEM.ACCOUNT file is to be displayed on the primary printer or PRINTERn if n is specified.

CLEAR Indicates that the account history file (SYSTEM.HISTORY) is to be transferred to the public file ACCOUNT.HISTORY. Any previous ACCOUNT.HISTORY file owned by the system will be erased.

KILL Indicates that the user accounting feature of the system is to be disabled. When this option is selected two questions will be asked of the operator. These questions must be answered correctly for the accounting feature to be disabled.

The user accounting feature is disabled by physically removing certain key programs and files from the system disk. Once this has been done the user accounting feature cannot be re-enabled without changing to an archive copy of the system.

This option should be used with great discretion for another reason: any accounts active in the system (files or programs owned by accounts other than the system) will be inaccessible after the accounting feature is KILLed. To avoid this situation you must use the OWNERCHG command to transfer ownership of the active account's files and programs to the system account. The SHARE command must also be used to remove any shared file access.

The two questions asked by this option are:

Do you wish to remove the accounting feature?

Do you ever wish to use the accounting feature again?

The first question must be answered with a Y and the second question must be answered with a N before the accounting feature will be KILLed.

COPY=fd Indicates that the current SYSTEM.ACCOUNT file is to be copied to the disk on the drive specified by fd. This is an easy means of transferring an

accounting structure to another system disk.

When no option is specified in the command line the account maintenance option of the command is invoked. This maintenance option allows for the addition, deletion, and changing of account names, passwords, and privilege levels.

When the ACCOUNT command is in the maintenance mode the program is interactive. The specific maintenance mode is determined by the operator's answer to the question:

Mode (Add/Change/Delete/End)?

Answer this question with the appropriate letter: A for adding an account, C for changing an existing account, D for deleting an existing account, or E for ending the maintenance session.

The add, change, and delete modes all ask for the account name to be specified. When the mode is add the account name entered must not be on file. When the mode is change or delete the account name must be on file.

Add Mode

In the add mode the program asks if the new account is to be a synonym to an existing account. If it is then enter the existing account name that it is a synonym to. Synonyms to existing accounts mean that the synonym account has access to all of the files that the primary account has access to. A synonym account has its own password and privilege level associated with it.

If the new account is not to be a synonym of an existing account then enter a carriage return only in response to the synonym question. Whether or not the new account is a synonym to an existing account the program will ask for the password to be associated with the new account. The rules for the composition of a password are the same as for file names. If the new account is not to be password protected then do not enter a password.

Next the program asks for the privilege level associated with the account. The privilege level is a number in the range of 0-5 but you are only allowed to specify privilege levels less than or equal to your current privilege level.

After the above information is entered the program returns to the Mode question.

Change Mode

The change mode of maintenance allows the operator to change the current password and privilege level associated with an account name. To delete the password type the number sign character (#) in response to the password question.

Delete Mode

The delete mode will remove the account name specified from the SYSTEM.ACCOUNT file and return to the Mode question.

Care should be exercised when deleting an account name. The system maintains accounts not by name but by a number that it assigns to new accounts. When an account is deleted any files it may have had will not be accessible. If a deleted account is later re-added it will probably be assigned a different number than it

had previously and the files will still be un-accessible. (In fact, it is probable that a completly different account name will be added at some time that is assigned the same number by the system. When this happens any files still in the system with the old deleted account's number will suddenly belong to the new, unrelated account.)

Therefore it is advised that before deleting an account a check should be made to see if that account owns any files. If so the files should be transferred to another account with the OWNERCHG command or erased from the system.

It is okay to delete an account if a synonym to that account still exists (synonym accounts are accounts with the same number but a different name).

End Mode

During the maintenance session all additions, changes, and deletions are kept internally to the program. To cause the SYSTEM.ACCOUNT file to be updated with the changes the END mode is used. If, instead, the System Cancel-key is entered none of the additions, changes, or deletions will be made to the file.

Examples

```
>ACCOUNT

Mode (Add/Change/Delete/End)? A

Account name? PRIVATE
Synonym name?
Password? SUPERMAN
Privilege? 3

Mode (Add/Change/Delete/End)? A

Account name? STARTREK
Synonym name? PRIVATE
Password? KIRK
Privilege? 0

Mode (Add/Change/Delete/End)? D

Account name? STARWARS

Mode (Add/Change/Delete/End)? C

Account name? PRIVATE
Password? #
Privilege?

Mode (Add/Change/Delete/End)? E
```

```
>ACCOUNT (TYPE)

Account    Password  Priv  Id

IPL                  0     0
PRIVATE              3     2
STARTREK   KIRK      0     2
SYSTEM               3     1

>
```

In the above example session the account PRIVATE is added with a password of SUPERMAN, and a privilege level of 3. Next a synonym is added by adding the account STARTREK as a synonym of PRIVATE. This synonym has the password KIRK and a privilege level of 0. This synonym account will be able to access all of PRIVATE's files (and vice versa) but, because of its lower privilege level, will not be able to access all of the programs in the account or in the public account.

Next the account STARWARS is removed from the system. This will have no effect on any synonyms that STARWARS may have had. Finally the password for the account PRIVATE is removed.

CHAPTER 11

ASSIGN COMMAND

The ASSIGN command provides the ability to make I/O channel assignments outside of an OASIS program. This means that a program can be written that is more general in purpose. Before the program is executed you would use the ASSIGN command to specify the files that the program is to use. The format of the ASSIGN command is:

ASSIGN [<ch>|* <file-desc>|<device>|CLEAR]

Where:

ch — Indicates one of the I/O channels numbered from 1 to 16.

* — Indicates that all I/O channels are to affected. This specification is only valid with the CLEAR function.

device — Indicates one of the I/O devices available to the system. The device names are listed in Appendix B. The device name DUMMY is a special name that means that the channel is assigned but there is no input or output device associated with it. A program accessing the device DUMMY will act as though the access was performed but nothing will have been output or input.

CLEAR — Indicates that the I/O channel is to be unassigned. If an asterisk was used for ch then all channels will be cleared of assignments.

You may use the ASSIGN command with no channels or devices specified. In this situation the system will display on the console all of the current channel assignments.

ASSIGN Examples:

```
>ASSIGN 1 FILE1.TYPE1:A

>ASSIGN 2 FILE2.TYPE2:A

>ASSIGN 3 PRINTER

>ASSIGN 4 CONSOLE

>ASSIGN
 1 FILE1.TYPE1:A
 2 FILE2.TYPE2:A
 3 PRINTER
 4 CONSOLE

>ASSIGN 2 OASIS.DOCUMENT:B

>ASSIGN
 1 FILE1.TYPE1:A
 2 OASIS.DOCUMENT:B
 3 PRINTER
 4 CONSOLE

>ASSIGN 3 CLEAR
```

```
>ASSIGN
 1 FILE1.TYPE1:A
 2 OASIS.DOCUMENT:B
 4 CONSOLE

>ASSIGN * CLEAR

>ASSIGN

>
```

CHAPTER 12

ATTACH COMMAND

The ATTACH command is used to associate physical device drivers with logical device names. This allows the operating system to be configured to the specific peripherals available. Additionally, this command allows you to dynamically change the actual device of I/O without making changes to any programs. The format of the ATTACH command is:

ATTACH [<logdev> [<phydev> [(<options>...[)]]]]

Where:

logdev Is the logical device name of the device to be attached to the physical device.

phydev Indicates the physical device name of the device, such as CRT or LP (line printer) etc.

The ATTACH command attaches physical device handlers to logical device names. These logical device names must be one of the names known to the system. These names are listed in the following tables.

A logical device is not really a device or peripheral. It is merely a common communication link from an application program to a specific physical device. The logical device is only a pointer to the specific device's driver program and is accessed by using the logical device name. In this manner programs gain a great deal of flexibility: an application program need only reference the logical device name, such as PRINTER1, to perform input/output; no programming is required by the application program to accomodate the requirements of the physical device. In this case the 'PRINTER' may actually be a paper tape punch; tape drive, etc.

Another attribute of a logical device is the general device class. For example: a disk drive is a random access, file structured, read/write device; a printer is a sequential, record oriented, write only device; a tape is a sequential, record oriented, read/write device; a console is a sequential, character oriented, read/write, 'softcopy' (CRT) device with cursor addressing abilities; etc.

Logical devices may be detached by using the ATTACH command with no physical device specified. Logical devices may be re-attached to another device without first detaching the device. Physical devices may be attached to more than one logical device name at any one time (except for disks).

Logical Device Names

Name	Synonym	Restrictions-Definition	Options
CONIN		Console input (keyboard)	
CONOUT		Console output (screen, etc)	P,L,C
CONSOLE		Combines both CONI & CONO	P,L,C
S		System disk	H,STP,STS,R
A thru G		Disks	H,STP,STS,R,PUBLIC
PRINTER1	PRT		P,L,O,ALF,PP
PRINTER2	PRT2		P,L,O,ALF,PP
PRINTER3	PRT3		P,L,O,ALF,PP
PRINTER4	PRT4		P,L,O,ALF,PP
READER	RDR	Character oriented, input device	
PUNCH	PCH	Character oriented, output device	
COMM1	COM	Communications	
COMM2	COM2	Communications	
COMM3	COM3	Communications	
COMM4	COM4	Communications	
TAPE1	TAP	Record oriented device	
TAPE2	TAP2	Record oriented device	
TAPE3	TAP3	Record oriented device	
TAPE4	TAP4	Record oriented device	

Physical Device Names

Name*	Synonym*	Restrictions-Definition	Options
CRT	VDM	Video Display Monitor	
KEY	KB	Keyboard	
SIO1	SIO	Serial I/O port	B,LF,FF,PO,PE,ENAB
SIO2 thru SIO8		Serial I/O ports	B,LF,FF,PO,PE,ENAB
PIO1	LP or PIO	Parallel I/O port	
PIO2 thru PIO8		Parallel I/O ports	
DISK1 thru DISK8		Disk drives	H,STP,STS,R,PUBLIC
nn		Physical device driver numbers	Any that may apply.

* Specific names and synonyms are determined by the SYSTEM.DEVNAMES file. Also see OASIS Implementation Guide for your specific configuration.

The system disk (S) may be re-attached to a different disk drive. When this is done the attach program will request that you mount the new system disk on that drive. When this has been done the system will find the location of the various programs on the new system disk. The new system disk must have the have a NUCLEUS of the same version as the old system disk.

When the serial port is attached the number of stop bits generated for each byte transmitted is dependent upon the baud specified: a baud of 110 causes two (2) stops bits; a baud greater than 110 causes one (1) stop bit. This is generally consistent within the data processing industry.

When a physical device is attached for the first time the device driver program (a SYSTEM file) is loaded into high memory, removing that memory from general use.

Detaching a device will unload the device driver (thus freeing up memory) if, and only if, the device is not attached to another logical name and the device driver is the last driver loaded in memory (driver at top of stack).

The ATTACH command cannot be used from BASIC or EDIT to attach a device for the first time (except disk drives).

The ATTACH command allows you to attach disk drives out of sequence. This means that you may have drives S, D, and G attached without having drives A, B, C, E, and F attached. In addition the drives may be detached with no regard for other disk attachments.

All device attachments are SYSGENable. If the following disk attachments have been made and SYSGENed:

```
S DISK1
A DISK2
D DISK4
G DISK3
```

and the system is re-booted with the system disk in a drive other than DISK1, the system will automatically detach all drives equal to the system disk. For example, if the above system were re-booted with the system disk in drive 3 the attachments would be:

```
S DISK3
A DISK2
D DISK4
```

When a device is attached in the multi-user OASIS it is a private device, i.e., only the user attaching the device may access it. An exception to this is a disk device which may have an option of PUBLIC specified.

ATTACH Options

The ATTACH command allows various options to be specified when certain devices are attached. These options are available depending upon the physical or logical device being attached. The previous tables listed the options available for each logical or physical device.

Bnnnn Indicates the BAUD (bits per second) of the serial device being attached. When this option is not specified for a serial device the default BAUD of 9600 is used. The available BAUDs that may be specified include: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400.

Note: Not all system support programable baud rate generators. This option is only effective when such a generator is used. For other systems a hardware change (a switch setting or solder dip header) must be made to change the baud.

PO Indicates that the serial device requires odd parity communication.

PE Indicates that the serial device requires even parity communication.

When PO or PE is not specified then no parity is used.

The options B, PO, and PE only apply when a device is attached to one of the serial ports (SIOn).

Cn Indicates the class of terminal or printer. Many terminals have special codes to control the cursor or forms. The class code number specifies which SYSTEM.CLASSnn file to use for translation of the codes used by the specific terminal. The class code may only be specified for the CONO device.

If you need to use a terminal as a console and it is not one of the terminals listed below then use class code 0 or refer to the Appendix on Terminal Class Codes and the OASIS MACRO Assembly Language Reference Manual for details on how to write your own SYSTEM.CLASSnn file to control your terminal.

The class codes currently available for terminals include:

0 No cursor control available.
1 BEEHIVE type cursor control.
2 ADDS 580 type cursor control.
3 Lear Siegler terminal type cursor control.
4 SOROQ IQ type cursor control.
5 Polymorphic VDM (and equivalent) cursor control.
6 Hazeltine 1500 type cursor control.
8 Perkin-Elmer Fox type cursor control.
9 Perkin-Elmer Bantom type cursor control.
10 ANSI standard type cursor control.
11 INFOTON 100 type cursor control
16 QUME Sprint 5 type printer.
17 DIABLO 16xx series type printer.
18 DTC 302 type printer.
19 NEC Spimwriter type printer.

Class codes 16 through 19 are used only by the SCRIPT processor to determine how to do enhanced printing, etc.

ENAB[n] Indicates that transmission on the serial port is to use a specific handshake protocall in the output direction only. There are three protocalls known to OASIS serial drivers:

ENAB1 Monitor the DTR line (data terminal ready). When this line has a true status a character will be output to the device. This is the most efficient handshake protocall used by OASIS.

ENAB2 Each line of data sent to the device is terminated with a DC1 character. Transmission of another line of data is not begun until a DC3 character is received from the device.

ENAB3 Each line of data sent to the device is terminated with an ETX character. Transmission of another line of data is not begun until an ACK character is received from the device.

ENAB4 Monitor the CTS line (clear to send) from the device. When this line has a true status a character will be output to the device.

Note: Enable options 1 and 4 may not be supported by your system.

FFnnn Indicates the number of milliseconds (one thousandths of one second) required by the device to execute a form feed. The range of this value may be between 0 and 63999 (up to 64 seconds). The default value is zero. This option may be specified in addition to the ENABLE option--both will be in effect.

Lnnn Indicates the line length for the listing device being attached. When this option is not specified the default value for the specific device is used. Normally the default is 80.

LFnnn Indicates the number of milliseconds required by the device to execute a line advance (line feed). The range of this value may be between 0 and 63999 (up to 64 seconds). The default value is zero. This option may be specified in addition to the ENABLE option--both will be in effect.

Pnnn Indicates the page size of the device being attached. This value is used when the system is outputting data to the device. The value indicates the number of lines of data that are to be transmitted before a page eject command is given. When this option is not specified the default value for the specific device is used. Normally the default is 56.

This option and the L option have no effect when a user program is performing the output.

Onn Indicates the overflow size of a listing device. The overflow size is the number of lines between the last print line of a page and the first print line of the next page. The specification of this option implies that the listing device does not have a form feed capability. Most listing devices do have form feed capabilities. The default for this option is zero, implying that the device does have form feed capability.

ALF This option specifies that the listing device has an automatic line feed upon carriage return. When this option is not specified OASIS will append a line feed character to a carriage return character on output (unless ANSI forms control is specified and the no line advance control is used). Most listing devices can optionally set the Auto Line Feed off.

PP This option specifies that you wish page parity to be maintained on the listing device. Page parity means that a report or listing to the device is to contain an even number of pages (an extra form feed is performed, if necessary). This option is useful in causing reports to start on an "infold". When page parity is in effect the listing device is assumed to be at the correct page when a file is opened to it.

PUBLIC Indicates that the disk is to be attached with public access, i.e., all users may utilize the disk. When this option is not specified the disk may only be used by the user attaching it. This option may only be used when in single user mode.

Rnn Indicates the number of disk errors that may occur consecutively before a message is displayed on the console.

Hnnn Indicates the number of milliseconds required for disk head load time. This option is used to fine tune the disk driver to your specific set of disks.

STPnnn Indicates the number of milliseconds required for track-to-track disk head step time. This option is used for fine tuning the disk driver to your specific set of disks.

STSnnn Indicates the number of milliseconds required for disk head settle time. This option is used for fine tuning the disk driver to your specific set of disks.

Since there is actually only one physical device driver for all of the disks an option specified for one disk applies to the others.

Physical Device Synonyms

Physical device names, synonyms, and general attributes are contained in the system file SYSTEM.DEVNAMES:S. The OASIS system is distributed with the standard device names listed in the table above. You may change, delete or add device names and synonyms by editing this file.

Refer to the Appendix on "System Files" for details about this file.

ATTACH Examples

```
>ATTACH READER SIO1

>ATTACH PRINTER PIO (L66,PP

>ATTACH A DISK3

>ATTACH

Name      Device          Options
S         DISK1           "SYSTEM$ ",avail=10K,dirsize=80,Public
A         DISK3           ** Not Mounted **
B         DISK2           "DATA1   ",avail=247K,dirsize=16,WP
CONIN     KEYBOARD
CONOUT    CRT             L63,P15,C5
PRINTER1  LINE-PRINTER    L66,P56,PP
READER    SIO1            B9600,LF0,FF0
```

The above example illustrates the use of the ATTACH command to show the current attachments. For attached disks additional information is given about each disk, under the options column. The first field in this column is the label of the disk, next is the available capacity of the disk, the capacity of the directory, and whether the disk is write protected.

Note that the phrase "Not mounted" may appear. This has no relationship to whether or not a disk is mounted in the drive. It only means that the system has not accessed the disk since the last mount was performed and therefore does not know the label information, etc. The disk is attached!

CHAPTER 13

BACKUP COMMAND

The BACKUP command allows you to copy the entire contents of a disk to another disk. This is usually done on a periodic basis for protection or archive purposes. The format of the BACKUP command is:

BACKUP [<fd1> <fd2>]

Where:

fd1 Indicates the directory label of the source disk.

fd2 Indicates the directory label of the destination disk.

If you omit fd1 and fd2 then OASIS will perform the backup from the drive attached as 'S' to the drive attached as 'A'.

When the BACKUP program is entered the source and destination directory labels are displayed for confirmation. If the displayed directory labels are correct enter a carriage return to indicate that you have loaded your disks and are ready to continue. If the drive labels are incorrect then enter an 'N', carriage return. The program will then ask you to enter the new source and destination drive codes.

The BACKUP program next informs you of the disk labels of your source and destination disks and waits for confirmation. If the disk labels are of the disks that you intended to backup enter a carriage return and the backup process will proceed. If the disk labels are incorrect enter an 'N', carriage return, and the program will redisplay the drive labels.

After the disks have been copied the BACKUP program asks if you wish to backup another set of disks. If you do then enter a 'Y', carriage return and the cycle is repeated. If you do not wish to copy another disk then enter a carriage return.

BACKUP performs an automatic MOUNT of the source and destination drives before and after a disk copy.

Note: If, after the backup operation is complete, a SHOW DISK of the destination reports a +8 mis-allocation, then the backup operation did not go to completion. Retry the BACKUP procedure, making sure that it goes to completion.

The BACKUP command may only be used to copy a privately attached disk to another privately attached disk or to copy from or to a public disk while in single user mode.

The following example illustrates the display of the BACKUP command.

```
>BACKUP

Source on drive S(1)
Destination on drive A(2)
Mount disk now (Y/N) - N

Enter new source drive code - A
Enter new destination drive code - B

Source disk is labeled "DISK1234"
Destination disk is labeled "DISK4321"
OK to start copy (Y/N) - Y

End of copy, again (Y/N)? - Y

Source on drive A(2)
Destination on drive B(3)
Mount disks now (Y/N) - Y

Source disk is labeled "LABEL1   "
Destination disk is labeled "LABEL2   "
OK to start copy (Y/N) - Y

End of copy, again (Y/N) - N

>
```

CHAPTER 14

CHANGE COMMAND

The CHANGE program provides a means of changing the privilege value of OASIS system programs. The format of the command is:

CHANGE <file-desc> (PRIV <nnn>[)]

Where:

nnn Indicates the value that the programs privilege value is to be set to. The range of values that a privilege value may be set to include 0 to 5.

Note that this command requires the option PRIV to be used, unlike other commands where options are optional.

The file-desc must be of an absolute or relocatable program file.

When this command is executed the privilege value of the program specified by file-desc is changed to the value indicated. Afterwards that program may only be executed if the current system privilege level is equal to or greater than the privilege value of the program.

You may not change the privilege value of a program if the current privilege value of that program is greater than your current privilege level.

Refer to the appendix on Privilege Levels for further information.

Example:

```
>CHANGE EDIT COMMAND S (PRIV 2
```

CHAPTER 15

COPYFILE COMMAND

The COPYFILE command allows you to copy one file to another, copy groups of files, copy a file or group of files from one disk to another, or copy selected records of one file to another. The format of the COYPFILE command is:

COPYFILE <file-desc1> [<file-desc2>] [(<options>[])]

or

COPYFILE <dev1> <file-desc2> [(<options>[])]

or

COPYFILE <file-desc1> <dev2> [(<options>[])]

or

COPYFILE <dev1> <dev2> [(<options>[])]

Where:

<file-desc1> Indicates the file description of the source file. Wildcards are acceptable. The file disk must be specified explicitly or with an asterisk as the syntax of this command is position dependent. All files that you have access to (private, public, and shared) may be source files.

<file-desc2> Indicates the file description of the destination file. An equal sign is acceptable for each element, indicating that the source file's description is to be used. Destination files are always private files. If the source file is a shared file then the destination file description must be different than the source or an error will occur. (A shared file has a directory entry in your account name. The copy process detects the existing file entry but cannot erase it as it is not trully yours.)

<dev1> Indicates the logical device name or the I/O channel number of the source file.

<dev2> Indicates the logical device name or the I/O channel number of the destination file.

Optionally you may specify only one file description. This is interpreted by the COPYFILE command as indicating an in-place file translation.

COPYFILE Options

Options for the COPYFILE command are as follows:

TYPE Indicates that the results of each file copy is to be displayed on the console. This is a default option.

NOTYPE Indicates that the results of each file copy are not to be displayed on the system terminal. When this option is not specified the source and destination file descriptions are displayed on the console along with the copy type ("copied to", "replaces", "appended to", etc.).

NEWDATE Indicates that the destination file(s) is to use the current system date for its directory entry.

OLDDATE Indicates that the destination file(s) is to use the date from the directory entry of the first source file for its directory entry. This is a default option.

FROM nnn Indicates the first record to be copied from the source file. nnn is the decimal record number and must follow the option FROM.

FRLABEL xxxxxxxx Indicates the first record to be copied from the source file. xxxxxxxx is a character string which appears at the beginning of the first input record to be copied. This string may be up to eight characters in length and may not include imbedded blanks. If the source file does not contain a record that starts with this character string then no records will be copied.

FOR nnnn Indicates the number of records that are to be copied including the first record qualifying.

TOLABEL xxxxxxxx Indicates the last record to be copied from the source file. xxxxxxxx is a character string which appears at the beginning of a record. The record containing this string is not copied but indicates that no more records from this source file are to be copied. This string may be up to eight characters in length and may not include imbedded blanks. If the source file does not contain a record starting with this string after the first qualifying record then the remainder of the file is copied.

TRUNC Indicates that trailing blanks from each input record are to be removed before the record is copied.

NOTRUNC Indicates that trailing blanks from each input record are not to be removed before the record is copied. This is a default option.

PACK Indicates that repetitive characters in the source file are to be coded and packed before copying. All consecutive occurrences of two or more blanks in the input records are encoded as a two character sequence. All consecutive occurrences of four or more other characters are encoded as a three character sequence.

UNPACK Indicates that encoded repetitive characters are to be expanded into the normal format.

When neither PACK nor UNPACK is specified the copy will be performed without any compression or expansion of repetitive characters.

LOWCASE Indicates that all uppercase characters in the input records are to be translated into their lower case equivalents before being copied.

UPCASE Indicates that all lower case characters in the input records are to be translated into their upper case equivalents before being copied.

NEWFILE Indicates that destination files are not to replace any pre-existing files. If a destination file specification is identical to an existing file specification then the copy is not performed. This is a default option.

REPLACE Indicates that the destination file may replace a pre-existing file of the same specification. This is a default option only when an in-place file copy is specified. Note: DELETE protected files may not be replaced (see the RENAME command).

APPEND Indicates that the output is to be appended to a pre-existing file of the same specification. If no file exists with the same specification then one is created. Note: WRITE protected files may not be appended to (see the RENAME command).

MOUNT Indicates disks must be changed before the copy can be performed. When this option is specified COPYFILE will pause before beginning the copy, display the message "Change disks now" and wait for the operator to type a carriage return before performing the copy.

PUBLIC Indicates that COPYFILE is to allow public files to be included in the source files.

VERIFY Indicates that COPYFILE is to verify its copy by performing a read after write.

NOVERIFY Indicates that COPYFILE is not to verify its copy. This is a default option.

QUERY Indicates that COPYFILE is to ask you on a file-by-file basis, if the file is to be copied. A reply of Y is required for the file to be copied. This is a default option if the source file description includes wildcards.

NOQUERY Indicates that COPYFILE is to copy each file that matches the specifications without asking the operator for permission. This is a default option when the source file description is explicit. When this option is not used and the input file description is not explicit (uses wildcards) the COPYFILE program will display each file description that matches the specification and ask (query) the operator if the file is to be copied.

TRANS Indicates that a character translation is to be performed on the source file as it is copied. When this option is specified COPYFILE will display the message "Enter Translation List" on the console terminal and accept input from the operator. The prompting character in this mode is the right parenthesis.

The translation list consists of pairs of characters entered from the console, separated by one or more blanks. Each character may be specified by typing the character itself or by typing its two digit hexadecimal equivalent.

The translation list may be continued onto additional lines by typing '++' as the last two characters of the line. This indicates that the translation list is continued on the next line. All characters after '++' on the line are ignored--may enter a comment after the ++. A line may not be continued in the middle of a character pair.

Note 1: When file-desc1 is a file that has READ protection (see RENAME command), the file must be copied in its entirety. This means that the following options are not valid when a READ protected file is being copied: FROM, FRLABEL, FOR, TOLABEL, TRUNC, PACK, UNPACK, LOWCASE, UPCASE, APPEND, and TRANS.

Note 2: When a file is copied in its entirety the destination file will have the same protection status as the source file. When a file is not copied in its entirety (any of the options listed in Note 1) the destination file will have no protection status set (except option APPEND which will not affect the protection status of the file being appended to).

Examples

```
>COPYFILE DAVID.LETTER:S = = A
DAVID.LETTER:S copied to DAVID.LETTER:A

>COPYFILE * BASIC:S = = A
PROG1.BASIC:S Okay to copy (Y/N)? N
PROG2.BASIC:S Okay to copy (Y/N)? Y
PROG2.BASIC:S copied to PROG2.BASIC:A
PROGRAM3.BASIC:S Okay to copy (Y/N)? N

>COPYFILE PROG1.BASIC:S PROG2.BASIC:S (APPEND
PROG1.BASIC:S appended to PROG2.BASIC:S

>COPYFILE PROGRAM3.BASIC:S PROG2.BASIC:S
PROGRAM3.BASIC:S not copied to PROG2.BASIC:S
       because file exists

>COPY PROGRAM3.BASIC:S PROG2.=:S (REPLACE NOTYPE

>COPY FILE1 TEXT S NEWFILE1 = A (FRL THE FOR 99 APPEND
FILE1.TEXT:S appended to NEWFILE1.TEXT:A

>C EXAMPLE FILE S = = A (TRANS
Enter Translation List:
)^ \ ++ TRANSLATE UP-ARROW TO BACKSLANT
). , ++ TRANSLATE PERIOD TO COMMA
)5F 2E ++ TRANSLATE UNDERSCORE TO PERIOD
)

>COPYFILE COMM1 PRINTER1 (TRUNC

>COPYFILE COMM4 DATA.FILE:A

>
```

CHAPTER 16

CREATE COMMAND

The CREATE command allows you to create new indexed or direct files. It is the only means of doing so. The format of the CREATE command is:

CREATE <file-desc> (<option>...[])

Where:

file-desc Indicates the file description of the file to be created. Wildcards are not allowed, and the file disk must be specified.

CREATE Options

INDEXED Indicates that the file is to be created as an indexed access file.

DIRECT Indicates that the file is to be created as a direct access file.

FILESIZE nn Indicates that the decimal number nn is the number of records the file is to contain. The maximum number of records per file is 65535.

If the file is a direct file then this number is the actual filesize. If the file is an indexed file then the CREATE program will create a file with at least this many records. The actual filesize will be determined by an algorithm required by the system's access method.

RECLEN nn Indicates that the decimal number nn is the physical length of each record in the file. The maximum length of a record in the system is 512.

This length is exclusive of any key length specified.

KEYLEN nn Indicates that the decimal number nn is the length of the index key. KEYLEN may only be specified for INDEXED files. The maximum length of a key is 128.

The CREATE command requires contiguous available space on the disk to create a file. If the largest contiguous available space is not large enough to hold the file being created the message "Disk full" is displayed.

The CREATE command sometimes appears slow to execute. This is due to the fact that CREATE not only creates the file but also initializes the entire file space to be null or empty records.

The number of blocks allocated to the file by the CREATE command is dependent upon the FILESIZE, RECLEN, and KEYLEN options.

When the file is a direct file then the allocation will be:

Allocation = RECLEN * FILESIZE / 1024 , rounded up.

When the file is an indexed file then the allocation will be:

Allocation =

2 * FILESIZE / 1024 , rounded up

+

(2 + RECLEN + KEYLEN) * FILESIZE / 1024 , rounded up

To compute the FILESIZE for a restricted allocation area the following formula should help:

$$\text{FILESIZE} = \frac{(\text{avail}) * 1024}{\text{RECLEN} + \text{KEYLEN} + 4}$$

CREATE Examples:

```
>CREATE CUSTOMER MASTER A (INDEXED FILE 2000 REC 64 KEY 30

>CREATE TRANSACT.DETAIL:ACCOUNTS (DIRECT FILE 9760 REC 32)
```

The first example will create an indexed file on drive A with space for 2003 records; length of key is 30 characters; length of record is 64. The number of disk blocks required for this file will be 192.

The second example will create a direct file on the disk labeled ACCOUNTS with space for 9760 records of length 32. The number of disk blocks required for this file will be 305.

CHAPTER 17

ERASE COMMAND

The ERASE command allows you to remove your own private files from the system. The format of the ERASE command is:

ERASE <file-desc> [(<option>...[])]

Where:

file-desc Indicates the file description of the file or files to be erased. Wildcards are acceptable.

ERASE Options

TYPE Indicates that the file description of each file erased is to be displayed on the console. This is a default option.

NOTYPE Indicates that the file description of each file erased is not to be displayed.

QUERY Indicates that the operator will be asked (queried) on a file by file basis if the file is to be erased. This is the default option when the file description is not explicit.

NOQUERY Indicates that the operator is not to be asked on a file by file basis if the file is to be erased. This is a default option when the file specification is explicit. If this option is not specified and the file description is not explicit (wildcards used) the file description of each file that matches the specification is displayed on the console and the operator is asked (queried) if the file is to be erased. Only a response of Y will cause the file to be erased.

The system will not allow this command to erase a file that has DELETE protect status. If a file has DELETE protect status you must first use the RENAME command to change the protect status.

Care should be taken if the file is shared by another account. If the file is to be erased and not re-added in the near future the operator should first use the SHARE command to verify if any other users have shared access to the file. If another account does have shared access the SHARE command should be used to remove this access before the file is erased. Having shared access to a deleted file may cause problems in the other account.

Examples

```
>ERASE F* * A
FILE1.BASIC:A Okay to Erase (Y/N)? N
FILELIST.COMMAND:A Okay to Erase (Y/N)? Y
FILELIST.COMMAND:A Protected

>ERASE X* * A
No files erased
```

```
>ERASE F* * A (NOT
FILE1.BASIC:A Okay to Erase (Y/N)? Y
FILELIST.COMMAND:A Okay to Erase (Y/N)? Y
FILELIST.COMMAND:A Protected

>
```

CHAPTER 18

FILELIST COMMAND

The FILELIST command allows you to display the directory entries for files in your account. The format of the FILELIST command is:

FILELIST [<file-desc>|<fd>] [(<option>...[])]

Where:

<file-desc> Indicates the file or files to be included in the directory search and list. Only files in your account will be included. To include files in the public account you must use the asterisk option. Wildcards are acceptable.

<fd> Indicates the directory label or disk label of the disk to be searched. When fd is omitted all of the ATTACHed directories will be searched. An fd without a file name or file type indicates that all of the files on the specified directory are to be listed. Only files in your account will be included, unless the asterisk option is used and then only files in your account and the public account will be included.

<blank> When the file-desc or fd is not specified all directories, all file names, and all file types are included in the listing.

Directory Fields

The directory contains all the information necessary for the system to maintain the files. This information is contained in a directory entry for each file. Most of the information is displayable by the FILELIST command. This information includes the following columns in the order listed here.

Fname This is the file name as defined earlier.

Ftype This is the file type as defined earlier.

Drv This is the directory label as assigned by the ATTACH command.

Date This field is updated with the current system date whenever the file is changed, created, or appended to by a user program or by one of the OASIS commands.

Time Similar to the date field.

Note: Your system may not support the time of day feature.

Recs This field indicates the current number of records in the file. For direct files it is the total file allocation as specified at creation time.

Blks This field indicates the number of 1024 byte blocks in use by the file.

Format This column contains two parts:

A character specifying the physical format of the records in the file. The character indicates its format is:

S Sequential
I Indexed access
D Direct access
R Relocatable command
A Absolute command

This character is followed by any protection status set for the file:

D Delete protected
W Write protected
R Read protected

The second part indicates the length of the records in the file. For sequential files it is the length of the longest record in the file as the file may contain records of variable length. For direct and indexed files it is the length as specified at creation time.

Label This is the disk label as assigned by the INITDISK command.

Sect This field is the starting sector number of the file and is only displayed when the asterisk option is specified.

Other This field is only displayed when the asterisk option is specified and contains two columns: the first column is a numeric value and the second column is a letter code indicating what the first column is. The letter codes are as follows:

E - Indicates that the preceding number is the ending sector number of the file (only for sequential files).
L - Indicates that the preceding number is the length of the program file (only for relocatable program files).
O - Indicates that the preceding number is the origin address of the program file (only for absolute program files).
F - Indicates that the number is the allocated filesize of the file (only for indexed data files).

FILELIST Options

FNAME Indicates that the list is to include only the file name column.

FTYPE Indicates that the list is to include the first two columns: Fname, and Ftype.

FDISK Indicates that the list is to include the first three columns: Fname, Ftype, and Drv.

DATE Indicates that the list is to include the columns: Fname, Ftype, Drv, Date, and Time.

Note: Your system may not support the time of day feature.

ALLOC Indicates that the list is to include the columns: Fname, Ftype, Drv, Date, Time, Recs, and Blks.

FORMAT Indicates that the list is to include the columns: Fname, Ftype, Drv, Date, Time, Recs, Blks, and Format.

LABEL Indicates that the list is to include the columns: Fname, Ftype, Drv, Date, Time, Recs, Blks, Format, and Label.

When none of the above seven options is specified the FILELIST command will display as many columns as can fit in a line on the device. This is determined by the attached line length.

* (Asterisk) Indicates that the listing is to include all information contained in the directory for both the public files and the files owned or shared by this account. Unless this option is specified only the files owned or shared by this account will be included in the listing.

This option must be used if you wish to have the Sect and Other columns displayed.

TYPE Indicates that the listing is to be displayed on the system console. This is a default option.

PRINTER[n] Indicates that the listing is to be displayed on the printer. If n is not specified then PRINTER1 is used.

EXEC Indicates that the listing is to be written to disk and given the name 'SELECTED.EXEC:S'. When EXEC is specified the default for fields included is FDISK. The records created in this file contain 9 variables for use when the file is executed by the EXECutive procedure processor. This option and the change in default allows you to create a file of file descriptions to be used by the EXEC command as explained in the manual describing the EXEC language. The file created by this option can also be used by a user written program, of course.

FILE Indicates that the listing is to be output to the system disk and given the name: SELECTED.FILES:S. This option is very useful for creating a data file of file names to be used by an application program. Similar to the EXEC option, the records created contain the filename, filetype, and filedisk unless another option specifies more or less information to be included.

APPEND Indicates that the listing is to be APPENDED to a previously created file named SELECTED.EXEC:S or SELECTED.FILES:S. This option is used in conjunction with the EXEC or FILE option.

HEADER Indicates that the listing is to include a heading at the top of each page. This is the default except when the EXEC option is specified or when less than the date column is listed.

NOHEADER Indicates that the listing is not to include a heading at the top of each page. This is the default when the EXEC option is specified or when less than the date column is listed.

SORT Indicates that the list is to be sorted by file name, file type, and file directory label. This is a default option.

NOSORT Indicates that the list is to be in the same sequence as in the directory.

mm/dd/yy Indicates that only files whose update date is greater than or equal to this specified date are to be included. The date must be in normalized format: mm/dd/yy.

FILELIST Examples

The following examples illustrate some of the options of the FILELIST command. The Date, Time, Recs, and Blks used in these examples do not necessarily reflect the current release of the operating system.

Note: Your system may not support the time of day feature.

```
>FILELIST S

Fname--- Ftype-- Drv -Date-- Time-  Recs Blks Format- Label---

ASSIGN   COMMAND  S 10/04/79 22:33     5    2 RDW 256 SYSTEM$C
ATTACH   COMMAND  S 10/05/79 20:53    13    4 RDW 256 SYSTEM$C
BACKUP   COMMAND  S 10/06/79 21:31     5    2 RDW 256 SYSTEM$C
BASIC    COMMAND  S 10/12/79 22:44    76   19 RDW 256 SYSTEM$C
COPYFILE COMMAND  S 10/05/79 22:04    20    5 RDW 256 SYSTEM$C
CREATE   COMMAND  S 10/04/79 12:32     6    2 RDW 256 SYSTEM$C
DUMPDISK COMMAND  S 10/06/79 22:27     7    2 RDW 256 SYSTEM$C
EDIT     COMMAND  S 10/02/79 10:34    54   14 RDW 256 SYSTEM$C
ERASE    COMMAND  S 10/12/79 15:11     5    2 RDW 256 SYSTEM$C
FILELIST COMMAND  S 10/29/79 22:16    13    4 RDW 256 SYSTEM$C
INITDISK COMMAND  S 10/04/79 12:25     8    2 ADW 256 SYSTEM$C
IPL      EXEC     S 10/24/79 23:25     2    1 S    12 SYSTEM$C
LIST     COMMAND  S 10/08/79 23:50    15    4 RDW 256 SYSTEM$C
LINK     COMMAND  S 10/08/79 19:03    10    3 RDW 256 SYSTEM$C
MEMTEST  COMMAND  S 10/11/79 20:53     4    1 RDW 256 SYSTEM$C
MOUNT    COMMAND  S 10/04/79 15:48     2    1 RDW 256 SYSTEM$C
PATCH    COMMAND  S 10/30/79 14:05     9    3 RDW 256 SYSTEM$C
RENAME   COMMAND  S 10/12/79 16:43     8    2 RDW 256 SYSTEM$C
RUN      COMMAND  S 10/18/79 22:55    68   17 RDW 256 SYSTEM$C
SET      COMMAND  S 10/11/79 14:58     6    2 RDW 256 SYSTEM$C
SHOW     COMMAND  S 10/12/79 22:49    13    4 RDW 256 SYSTEM$C
SYSGEN   COMMAND  S 10/12/79 21:41     2    1 RDW 256 SYSTEM$C
SYSTEM   CSI      S 10/11/79 13:12    22    6 ADW 256 SYSTEM$C
SYSTEM   DEV10    S 10/21/79 13:22     2    1 RDW 256 SYSTEM$C
SYSTEM   DEV11    S 10/21/79 13:25     4    1 RDW 256 SYSTEM$C
SYSTEM   DEV12    S 10/21/79 13:32     1    1 RDW 256 SYSTEM$C
SYSTEM   DEV13    S 10/21/79 13:07     3    1 RDW 256 SYSTEM$C
SYSTEM   DEV9     S 10/21/79 20:04     8    2 RDW 256 SYSTEM$C
SYSTEM   EXECSAVE S 10/22/79 15:57     8    4 DD  512 SYSTEM$C
SYSTEM   NUCLEUS  S 10/11/79 22:50    55   14 AD  256 SYSTEM$C
SYSTEM   UPDATES  S 10/12/79 15:44   133    4 S    44 SYSTEM$C
31 file(s), 132 block(s).
```

```
>FILELIST * EXEC S (FD
IPL      EXEC     S
1 file(s), 1 block(s).

>FILELIST S* * S (DATE)

Fname--- Ftype-- Drv -Date-- Time-

SET      COMMAND  S 10/11/79 22:55
SHOW     COMMAND  S 10/12/79 14:58
SYSGEN   COMMAND  S 10/12/79 22:49
SYSTEM   CSI      S 10/11/79 21:41
SYSTEM   DEV10    S 10/21/79 13:12
SYSTEM   DEV11    S 10/21/79 13:22
SYSTEM   DEV12    S 10/21/79 13:25
SYSTEM   DEV13    S 10/21/79 13:32
SYSTEM   DEV9     S 10/21/79 13:07
SYSTEM   EXECSAVE S 10/22/79 20:04
SYSTEM   NUCLEUS  S 10/11/79 15:57
SYSTEM   UPDATES  S 10/12/79 22:50
12 file(s), 41 block(s).

>FILELIST S* * S (EXEC

>LIST SELECTED EXEC S

SELECTED.EXEC:S

&1 SET      COMMAND  S &2 &3 &4 &5 &6 &7 &8 &9
&1 SHOW     COMMAND  S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSGEN   COMMAND  S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   CSI      S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   DEV10    S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   DEV11    S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   DEV12    S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   DEV13    S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   DEV9     S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   EXECSAVE S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   NUCLEUS  S &2 &3 &4 &5 &6 &7 &8 &9
&1 SYSTEM   UPDATES  S &2 &3 &4 &5 &6 &7 &8 &9

>FILELIST D* * S (FNAME
DUMPDISK
1 file(s), 2 block(s).
```

```
    >FILELIST ??S??? * S (FN
    ASSIGN
    BASIC
    LIST
    SYSGEN
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    SYSTEM
    13 file(s), 28 block(s).

    >FILELIST S (10/22/79 FD

    Fname--- Ftype-- Drv

    FILELIST COMMAND  S
    IPL      EXEC     S
    PATCH    COMMAND  S
    3 file(s), 8 block(s).

>FILELIST SYSTEM DEV#* A (*

Fname--- Ftype-- Dr --Date-- Time- -Recs JlkC VlBMad, Label--- -Sect Other--
SYSTEM   DEV13    S 10/09/79 14:25     3    1 RWD 256 BACKUP    1364   260 L
SYSTEM   DEV17    S 10/09/79 14:26     5    2 RWD 256 BACKUP    1364   801 L
SYSTEM   DEV18    S 07/13/79 18:18     5    2 RWD 256 BACKUP    1376   804 L
3 file(s), 5 block(s).

>
```

CHAPTER 19

FORCE COMMAND

The FORCE command provides a means of forcing another user to perform a task. The format of the command is:

FORCE <pid> [<command>]

Where:

<pid> Indicates the partition identificatin number of the user to be forced.

<command> Is the command that the user is to be forced to do. When the command is not specified the user will be forced to exit from the program he is doing.

The FORCE command provides a useful, but dangerous, capability. Before using this command to force another user to perform some task you should use all other means to make sure that data or labor will not be lost. At the minimum, you should first send a message to the user asking him to perform the task voluntarily. If the user does not respond to this (away from console) then you should first check the status of his partition (SHOW MEMORY) to determine what he is doing and whether it would be okay to force him to perform your task.

When the FORCE command is executed the user specified is forced to execute his quit error trap (system cancel-key) and, if a command was specified, execute the command.

Normally, the FORCE command would only be used to LOGOFF users to get in a mode that would allow single-user operation.

It is possible that the user may have a quit error trap set such that his program is not aborted (TEXTEDIT does this). In this situation the user will not be forced to perform the command until he returns to the CSI level.

CHAPTER 20

INITDISK COMMAND

The INITDISK command allows you to initialize a disk for use or re-use by the OASIS operating system. The format of the INITDISK command is:

INITDISK [<fd>] [(<option>...[])]

Where:

fd Indicates the disk drive containing the disk to be initialized. When fd is omitted A is assumed.

INITDISK Options

Options for the INITDISK command include the following:

LABEL Indicates that you only wish to re-initialize the disk label. You may change the label of a disk that is write protected.

This is the only option which may be specified when the system disk (S) is being initialized. If the label of the system disk is changed, the system will respond with an error message: "Need System Disk". Enter a system restart to reboot.

CLEAR Indicates that you wish to initialize the directory space, erasing all existing files from that disk. This option is not valid when the system disk is initialized.

FORMAT Indicates that you wish to initialize the format of the entire disk. The formatting process entails the writing of track and sector addresses and a data area consisting of E5H values. You may use this option on a disk that is write protected. This option is not valid when the system disk is initialized.

WP Indicates that the specified disk is to be write protected. This option must be used by itself. When a disk is write protected the operating system will not allow any changes to the disk by any of the commands or user programs. (INITDISK with option FORMAT ignores any write protection on a disk for obvious reasons.)

NOWP Indicates that any disk write protection on the specified disk is to be removed. This option, when specified, must be used by itself.

SIZE Indicates the number of directory entries to be allocated in the new directory. This option requires the option CLEAR or FORMAT to be in effect. The SIZE option is followed by a number which is the desired number of directory entries. When this option is not specified a default directory size is used which is determined by the capacity of the disk. This option is not valid when the system disk is initialized.

When the size option is used the INITDISK program will allocate at least the specified number of entries. More may actually be allocated due to requirements of the system. The minumum directory size is determined by the format of the disk.

INCR Indicates the logical sector increment value. This option requires the option FORMAT and is followed by the number of the sector increment. Normally the default increment value is used (1) which causes a one-to-one mapping of physical sectors in a track to logical sectors. Changing this value might improve the disk I/O throughput (or degrade) by causing the sectors to be interleaved on a track which, when the rotational speed of the disk is taken into account, might optimize the consecutive sector read/write time.

SKEW Indicates the offset between logical sector zero of one track and logical sector zero of the next track. This option requires the option FORMAT and is followed by the number of the track skew. Normally the default skew (0) is used which indicates that tracks are not skewed. Changing this value might improve the disk I/O throughput (or degrade) when the track step-time is taken into account.

HEAD Indicates the number of disk surfaces available on the disk. This option requires the option FORMAT and is followed by the number of surfaces.

TRACKS Indicates the number of tracks per surface available on the disk. This option requires the option FORMAT and is followed by the number of tracks.

SECTOR Indicates the number of sectors per track available on the disk. This option requires the option FORMAT and is followd by the number of sectors.

It is necessary that a new disk be FORMATted before attempting to write data upon it.

When a disk is FORMATed the options SIZE, INCR, SKEW, HEAD, TRACK and SECTOR do not have to be specified unless the default values are not correct. These values will be displayed before the initialization process begins. If the values are not correct respond with a "N" and the program will be exited, allowing you to restart and to specify the correct values.

The actual default values is dependant upon the manufacturer of your computer system.

The option WP for write protection is different from the hardware write protect. The hardware write protect is a function of the disk controller and the removal of the "write enable" tab on the disk itself.

While a disk is being initialized the current track number of initialization is displayed.

INITDISK Examples

```
>INITDISK A (FORMAT

INITDISK will erase all files on drive A(2)
Do you wish to continue (Y/N)? N
Disk remains unchanged

>INITDISK A (LABEL)

Enter disk label: DISKIDX

>INITDISK A (CL

INITDISK will erase all files on drive A(2)
Do you wish to continue (Y/N)? Y

>INIT B (CLEAR SIZE 32

INITDISK will erase all files on drive B(3)
Do you wish to continue (Y/N)? Y

>INIT C (FORMAT SIZE 8

INITDISK will erase all files on drive C(4)
Do you wish to continue (Y/N)? Y

Enter disk label: TEST

Number of directory entries = 32
Increment between adjacent sectors = 1
Offset between adjacent tracks = 0
Number of surfaces = 1
Number of tracks/surface = 77
Number of sectors/track = 13

Is above information correct (Y/N)? Y

Track: 0

>
```

CHAPTER 21

LIST COMMAND

The LIST command allows you to display data files on the console or the system printer. The format of the LIST command is:

LIST <file-desc> [(<option>...[)]

Where:

file-desc Indicates the file description of the file to be listed. Although the file disk may be omitted, wildcards may not be used.

The file may not be a binary program file.

LIST Options

Options for the LIST command include the following:

rec1 Indicates that the list of the file is to start with this record number. This field must be numeric. When rec1 is greater than the number of records in the file no records will be listed. When this field is omitted 1 is assumed and recn cannot be specified.

recn Indicates that the list of the file is to end with this record number, inclusive. This field must be numeric or '*' and, when specified, must follow the specification of rec1. When this option is not specified or is specified as an asterisk the list will continue until the end of the file is reached.

TYPE Indicates that the list is to be output to the console. This is a default option.

PRINTER[n] Indicates that the list is to be output to the printer specified. If n is not specified PRINTER1 is used.

HEADING Indicates that the listing is to include a heading. This is a default option. The standard heading, when specified, includes the file name, file type, file disk label, current system time and date, and the page number.

NOHEAD Indicates that the listing is not to include a heading.

INDENT n Indicates that the listing is to be indented nn columns from the left margin. If this option is omitted then the listing will be displayed starting with column 1.

UPCASE Indicates that all lower case characters are to be "folded" to their upper case equivalents.

LOWCASE Indicates that all lower case characters are to be displayed "as is" without any translation. This is a default option.

TRUNC nn Indicates that each line of data is to be truncated at column nn. The data truncated will not be listed on the next line.

FORMAT Indicates that the file contains ANSI forms control characters. Refer to

the appendix on ANSI forms control.

PAGE nn Indicates that the output of the listing is to be suppressed until page number nn is started.

TITLE text Indicates that the listing is to include an alternate heading consisting of the text followed by the current system time, date and the page number. This option, when used, must be the last option specified as all characters following 'TITLE' will be treated as the text to the title. Any words in text will be truncated to eight characters by the CSI.

OBJ Indicates that the file to be listed is an object file. The LIST command will interpret the file and display it in a readable format: record length, record type, pab number, record starting address, record data.

Sequential format files are listed, one record per line (unless FORMAT option is specified). Direct and indexed format files are listed in a special format.

For these files the key (ASCII key for indexed, record number for direct) is listed followed by a colon, space. The remainder of the record is then listed, taking as many lines as needed. The record contents will always be indented from the key.

Files that have READ protection cannot be listed with this program (see RENAME command).

LIST Examples:

```
>LIST TEST OBJECT (OBJ NOHEAD

10 P 00 005C TEST     01 0100
15 T 00 0100 2146013610121470136007EFE09D21901
15 T 00 0110 2146017EB7C20001FF5F160021480119
15 T 00 0120 194E792346239657782 39EDDA3F01B2CA
 .
 .
 .
02 Z 00 1A1A

>LIST DATA FILE (NOHEAD

This is record 1:          ONE TWO THREE FOUR
This is record 2:          ONE TWO  FOUR
This is record 3:          ONE TWO THREE FOUR

>
```

CHAPTER 22

LOAD COMMAND

The LOAD command loads a re-entrant program for subsequent usage. The format of the command is:

LOAD <file-desc>

Where:

<file-desc> Indicates the re-entrant program to be loaded. When no file type is specified the default type of LOADFILE is used. When no file disk is specified the system disk will be searched first.

This command may only be used in a single user environment (no user partions defined--see SET command).

Before a re-entrant program may be used by multiple users the program must first be loaded into memory with this command.

Up to eight re-entrant programs may be loaded at one time.

After the re-entrant program is loaded with this command another program could be executed that checks to see if there is a re-entrant copy, and acts accordingly. An example of this is the BASIC.LOADFILE and the RUN.COMMAND. The RUN command, when invoked by the CSI, is loaded and executed. One of the first operations that it performs is to ask the operating system if BASIC has been loaded. If it has then it transfers control to it and releases its own memory for use by the re-entrant BASIC. If BASIC was not loaded then it continues operation as a non-re-entrant program.

Do not load a re-entrant program unless it is to be used--all user's available memory will be reduced un-necessarily.

CHAPTER 23

LOGOFF - LOGON COMMANDS

LOGOFF Command

The LOGOFF command provides the primary means of changing accounts in the system. The format of the command is:

LOGOFF

When the LOGOFF command is executed the current account is logged off, a history record is written to the SYSTEM.HISTORY file recording the log off time and elapsed logon time of the account. At this time control is transferred to the LOGON command described next.

The LOGOFF command detaches all of your private devices (except the console) and re-attaches the public disks.

LOGON Command

The LOGON command provides the only means of changing accounts in the system. The format of the command is:

LOGON <acct-name>

Where:

acct-name Indicates the user account name to be logged on to.

When the LOGON command is executed an automatic LOGOFF is performed on the current account. This automatic LOGOFF is different from your execution of the LOGOFF command in only one area: no history record is written to the SYSTEM.HISTORY file the previous account's elapsed time is transferred to the new account being logged onto.

The program searches the SYSTEM.ACCOUNT file for the account name specified. If the account name cannot be found the program will return to the "Logon please:" question.

When the account name is found in the SYSTEM.ACCOUNT file and the account requires a password (not all accounts do) but the operator has not entered one yet, the program will ask for it:

```
Password?
```

Again, the password entered will not be displayed on the console, asterisks will be used for each character typed.

If the password is incorrect the program will return to the "Logon please:" question.

When the account name and password are correct the program sets the privilege level of the system to the privilege level of the account (see ACCOUNT command) and resets all of the SETable switchs to the SYSGENed condition. At this time the user is logged on to the account specified and may only access public files and programs, his own private files and programs, and any files and programs that he has shared access to (see User Accounting and SHARE command).

A search is made of all attached directories for an EXEC program with the file description <acct-name>.EXEC. When this file is found it is executed. If the file cannot be found then control of the system returns to the CSI level.

The LOGON command will not allow you to log onto an account that is already in use by another user. This applies to the account number, not just the account name, therefore you may not log onto a synonym of an account in use by another user.

The LOGON command resets the switches RDYMSG, ABBREV, IMPEXEC, and VERSION to the status saved by the SYSGEN program.

Examples:

```
>LOGOFF

Logoff at 12:23:15, on 02/07/79, ET = 122.68

Logon please: NAMES

Password? ******

Logon at 12:23:20, on 02/07/79.

>LOGON USER

Logon at 12:24:10, on 02/07/79

>LOG

Logoff at 13:12:10, on 02/07/79, ET = 15.24

Logon please: NAMES,******

Logon at 13:12:15, on 02/07/79.

>
```

Note that the logon procedure is slightly different when performed following a LOGOFF. This is due to the fact that after the LOGOFF is executed there is no account logged on and the system will force the operator to log onto a valid account. When the LOGON command is executed and there is an error in the account or password the previous account is still logged onto the system.

CHAPTER 24

MAILBOX COMMAND

The MAILBOX command allows you to retrieve messages sent to you while you were unable to receive messages. The format of the command is:

MAILBOX [<file-desc>] [(<options>[])]

Where:

<file-desc> Is the file description of the file to be used by MAILBOX to copy your messages. When this parameter is not specified your messages will be displayed on the console or printer.

MAILBOX Options

PRINTER[n] Indicates that your messages are to be displayed on the printer.

APPEND May only be used when the <file-desc> parameter is specified. Indicates that the messages are to be appended to the end of the file specified by <file-desc>, if that file already exists. When this option is not used the file will first be erased before your current messages are copied to it.

MAILBOX Example

```
You have mail waiting.

>MAILBOX

From: TODD, 09:58 10/19/79
Sorry I missed you. I copied all of the programs you requested
to your account. Hope that they will do the job.

>MAILBOX SAVE.MAIL (APPEND

>LIST SAVE.MAIL (NOHEAD

From: SYSTEM, 10:15 10/19/79
Please turn in all of your log sheets, to date.

From: MANAGER, 10:30 10/19/79
The system will not be available tomorrow, the 20th due to installation
of additional disk capacity.

>
```

CHAPTER 25

MOUNT COMMAND

The MOUNT command allows you to inform the system that a disk is to be changed. The format of the MOUNT command is:

MOUNT <fd>|*

Where:

fd — Indicates the label of the directory that is to be changed.

* — Indicates that all attached disks may have been changed. (System disk (S) excepted).

This command informs the system that a disk has been changed. The system maintains a copy of each disk ID that is attached. This provides a safeguard to the user in preventing disk changes during updates, etc. If a disk is changed and this command has not been executed a disk error will occur: "Disk Changed". When this message is displayed and if is acceptable for the disk to have been changed then you may respond with an 'M', which will cause a mount to be performed on the drive.

The system disk (fd = S) may not be changed at any time without re-booting the system. This is due to the fact that once a system disk has been booted several internal pointers in the operating system have been set. Resetting these pointers is essentially what occurs when the system is re-booted.

Caution: The MOUNT command must be used when a change is made from a single density disk to a double density disk or from a double density disk to a single density disk. When such a change is made and the MOUNT command is not used the first access to the changed disk may cause the system to "hang".

Note: Not all systems have the ability to detect a changed disk. This is a function of the disk drive and controller. On systems that lack this capability the MOUNT command will have no effect.

CHAPTER 26

MSG COMMAND

The MSG command provides a means of sending a message to another user. The format of the command is:

MSG <pid> <text>

Where:

<pid> Indicates the partition identification number or the user name that the message is to be sent to. A pid of * indicates that the message is to be sent to all logged on users, including the sender. You may only broadcast a message to all user if you have a privilege level of five or greater.

<text> Is the message to be sent. When <text> is omitted the MSG command will allow you to enter multiple lines of text with the message terminated by a blank line.

When the MSG command is executed the message specified by <text> will be displayed on the user's console specified by <pid>. It is possible that this cannot be done. In this situation one of several messages may be displayed on your console:

Can't send message to your self.

User is not receiving messages.

User is not logged on.

In the latter situations where the receiving user is not logged on or has his message switch set off (see SET MSG command) then the MSG command will ask you if you wish to put the message in the user's mailbox. (See MAILBOX command.)

MSG Example

```
>MSG 2 Please load the invoice forms in the printer. (Send msg when done.)
User is not logged on
Do you wish to put this message in the mailbox (Y/N)? N

>MSG TOM Would you please send me a message when you log on?

>MSG *
Enter message text, terminate input by empty line.
?System reverting to single user mode at
?5:30 for backup purposes. Please be logged
?off by that time.

From: SYSTEM
System reverting to single user mode at
5:30 for backup purposes. Please be logged
off by that time.

>
```

CHAPTER 27

OWNERCHG COMMAND

The OWNERCHG command provides a means of transferring account ownership of a file or group of files. The format of the OWNERCHG command is:

OWNERCHG <file-desc> (<acct-name> [<option> ...][)]

Where:

file-desc Indicates the file description of the file or group of files whose ownership is to be transferred. Only the current account's private files may be changed with this command. Wildcards are acceptable.

<acct-name> Indicates the account name that the files are to be transferred to. Account names are validated against the SYSTEM.ACCOUNT file. Invalid names are rejected with no changes made to any files.

OWNERCHG Options

TYPE Indicates that the results of the operation are to be displayed on the console.

NOTYPE Indicates that the results of the operation are not to be displayed on the console. When this option is not specified the message "<file-desc> changed" is displayed for each file whose ownership is changed.

QUERY Indicates that the operator is to be asked (queried) for confirmation before each file qualifying is changed. This is a default option unless the file description is explicit (no wildcards).

NOQUERY Indicates that the operator is not to be queried on a file by file basis. This is a default option when the file description is explicit. When this option is not specified and the file description is non-explicit (uses wildcards) the file description of each file matching the description will be displayed on the console and the operator is asked (queried) if the file is to have its ownership changed. A response of "Y" indicates yes and a response of "N" indicates no.

Examples:

```
>OWNERCHG MYFILE.BASIC:S (MINE

MYFILE.BASIC:S changed

>OWNERCHG * BASIC A (YOURS

AARDVARK.BASIC:A Okay to change owner (Y/N)? Y
AARDVARK.BASIC:A is now owned by account "YOURS"
BINARY.BASIC:A Okay to change owner (Y/N)? Y
BINARY.BASIC:A is now owned by account "YOURS"
SATURDAY.BASIC:A Okay to change owner (Y/N)? Y
SATURDAY.BASIC:A is now owned by account "YOURS"
APPLE.BASIC:A Okay to change owner (Y/N)? N

>
```

CHAPTER 28

PATCH COMMAND

The PATCH command provides a means of implementing minor corrections to errors in object programs, such as the system programs. The format of the PATCH command is:

PATCH <file-desc>

The file specified by file-desc cannot be write protected and it must be a program file (absolute or relocatable in format). When the PATCH command is executed the specified program file is read into memory the PATCH prompt character (\) is displayed. At this time there are five subcommands available to the operator in addition to a HELP command.

All addresses and numeric values specified in the following subcommands must be hexadecimal in format.

28.1 Dump Subcommand

The Dump subcommand displays portions of the program on the console. The format of the subcommand is:

```
\Dump addr1 [addr2]
```

Addr1 specifies the beginning address to display. Addr2, when specified, indicates the ending address to display. When addr2 is not specified sixteen addresses will be displayed.

The Dump subcommand always displays the program in lines of sixteen addresses each.

28.2 End Subcommand

The End subcommand indicates that the "patch" is complete and the program is to be written back to disk. The format of the subcommand is:

```
\End
```

or

```
\File
```

28.3 Quit Subcommand

The Quit subcommand indicates that the "patch" is to be ignored and that control is to be returned to the operating system. The format of the subcommand is:

```
\QUIT
```

28.4 Replace Subcommand

The Replace subcommand specifies that the contents of an address is to be changed. The format of the subcommand is:

```
\Replace addr value [value]...
```

Addr indicates the first address to be modified by the list of values. When more than one value is specified or if the value is a character string, the values

replace the contents of the addresses following addr.

The list of values may be of three types: 8 bit numeric, 16 bit numeric, or quoted character string. For example:

```
8 bit numeric:          5 15 A0 3F

16 bit numeric:         (3900) (9F00) (1234)

character string:       'Syntax error'
```

28.5 Verify Subcommand

The Verify subcommand allows you to verify the current contents of addresses. The format of the subcommand is:

```
\Verify addr value [value]...
```

This subcommand is similar to the Replace subcommand except that the values are used for verification, not replacement.

PATCH Example

```
>RENAME SAMPLE COMMAND S (NOWRITE
SAMPLE.COMMAND:S protection: DELETE, NOWRITE

>PATCH SAMPLE COMMAND S
\VERIFY 4900 C3 03 8F
\REPLACE 4901 05
\DUMP 4900
4900: C3058FFD 215191AF E121235A E5211C54 '....!Q[..!#Z.!.T'
\VERIFY 8FE5 C3 52 8F
\REPLACE 8FE6 (5000)
\VERIFY 8FE5 C3 00 50
\VERIFY 6000 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
mismatch
\END

>
```

CHAPTER 29

PEEK COMMAND

The PEEK command allows you to see what another user is doing by copying his console output to your console. The format of the command is:

PEEK <pid>

Where:

<pid> Indicates the partition identification number or user name of the user to be peeked at.

When the PEEK command is executed all output to the specified user's console will be echoed to your console. If the user's console uses different screen control characters than yours the output may appear "funny".

While the PEEK command is operating only the system control keys will be acknowledged from your console. The only way to exit from the PEEK command is to enter the System Cancel-key.

The PEEK command should not be used to monitor another terminal for a long period of time as the system response does degrade.

CHAPTER 30

RENAME COMMAND

The RENAME command allows you to change the filename, filetype, or protection status of your own private file or group of files. The format of the RENAME command is:

RENAME <old fd> [<new fd>] [(<option> ...[])]

Where:

old-fd Indicates the current file description of the file or group of files to be renamed. Wildcards are acceptable.

new-fd Indicates the new file description to be assigned to the file or group of files qualifying to old-fd. Equal signs are acceptable for the components of the file description. The file disk must be an equal sign or the same file disk as specified in old-fd.

When this parameter is not specified only the protection status of the file(s) may be changed. Protection statuses may not be changed when new-fd is specified.

RENAME Options

TYPE Indicates that the results of the operation are to be displayed on the console.

NOTYPE Indicates that the results of the operation are not to be displayed on the console. When this option is not specified the new-fd of each file renamed is displayed on the console.

QUERY Indicates that the operator is to be asked (queried) for confirmation before each file qualifying is renamed. This is a default option unless the file specification is explicit.

NOQUERY Indicates that the operator is not to be queried on a file by file basis. This is a default option when old-fd is explicit. When this option is not specified and the old-fd is not explicit (wildcards used) the file description of each file matching the old-fd is displayed on the console and the operator is asked (queried) if the file is to be renamed. Any response other than an "N" will be interpreted as meaning yes.

DELETE Indicates that the file is to have DELETE or erase protection. A file with DELETE protection may not be erased or renamed without first changing the protection status.

NODELETE Indicates that the file is to have any DELETE protection removed from it.

WRITE Indicates that the file is to have WRITE protection. A file with WRITE protection may not be updated without first changing the protection status. A file with WRITE protection may still be erased. To prevent this you should also specify the DELETE option.

NOWRITE Indicates that the file is to have any WRITE protection removed from it.

READ Indicates that the file is to have READ protection. A file with READ

protection can only be accessed by the RUN command. This protection mode cannot be removed from a file. Since the READ protection mode cannot be removed it is advised that the programmer maintain an archive copy of the program in case any modifications need to be made to the program. A program file with READ protection cannot be LISTed, EDITed, DUMPed, MOVEd, SCRIPTed, COPYFILEd (except in its entirety), or loaded by the BASIC interpreter.

It is not necessary to specify WRITE protection in addition to READ protection as a file with READ protection cannot be accessed by any commands or functions that might try to write to the file.

A file with READ protection may be deleted.

Example:

```
>RENAME TEST BACKUP A = BASIC
TEST.BACKUP:A renamed TEST.BASIC:A

>RENAME AARDVARK.BASIC:A (READ
AARDVARK.BASIC:A protection: NODELETE, NOWRITE, READ

>RENAME TEST * S (NODELETE NOW
TEST.COMMAND:S Okay to change protection (Y/N)? Y
TEST.COMMAND:S protection: NODELETE, NOWRITE
TEST.OBJECT:S Okay to change protection (Y/N)? N
TEST.ASSEMBLE:S Okay to change protection (Y/N)? N
TEST.BASIC:S Okay to change protection (Y/N)? Y
TEST.BASIC:S protection: NODELETE, NOWRITE

>
```

CHAPTER 31

SCHEDULE COMMAND

This documentation of the SCHEDULE command is preliminary and subject to change.

The SCHEDULE command provides a means of scheduling a job to be executed at a later date and/or time. The format of the command is:

SCHEDULE <date> <time> <command>

Where:

<date> Is the formated date that the job is to be performed on.

<time> Is the formated time that the job is to be performed at.

<command> Is the command name and options that is to be performed at the specified date and time.

The SCHEDULE command, when executed, saves the information for subsequent action and then returns control to the CSI level.

When a job has been scheduled the CSI checks to see if it is time to perform the job. This check is performed every time that you are at the CSI level. If it is time to perform the job then it automatically executes the saved command string.

If you inadvertantly schedule a job with a date and time that has already occured then the job will be executed immediately.

(This page left blank intentionally.)

CHAPTER 32

SET COMMAND

The SET command allows you to change various system parameters and options. The format of the SET command is:

SET <function> <status>

Where:

function Indicates one of the system functions or parameters that the SET command may alter.

status Indicates the status or condition that the function is to be set to.

SET Functions

The various functions that the SET command may alter are defined as follows:

SET ABBREV ON|OFF

This function allows the user to change the status of the switch that allows commands to be abbreviated. Normally commands are allowed to be abbreviated.

For a description of the effects of changing ABBREV and SYN refer to the chapter on the Command String Interpreter, CSI Program Search Sequence section.

SET CANCEL nn

This function is similar to the SET ESCAPE function except that it operates on the line cancel key.

SET CLKFRQ nn1 [nn2]

This function requires privilege level 5 or greater and provides an easy means for the user to fine tune the time of day clock. If, for whatever reason, the time of day is not accurate, this function may be used to adjust the physical interval of time between "ticks" of the clock. The two values nn1 and nn2 are used for gross and fine adjustments, respectively. Any values entered will replace the current values so it is advised that you use the SHOW CLKFRQ command to determine the current values first before adjusting them with the SET CLKFRQ command.

Decreasing the value of either or both parameters makes the time of day clock run faster--Increasing the value of either or both parameters makes the time of day clock run slower. Decreasing one parameter and increasing the other parameter will have a combined effect on the speed of the time of day clock.

SET CLOCK ON|OFF

This function causes the current system date and time to be constantly displayed on the console screen. When the CLOCK is set ON the current system date will be displayed in the upper left hand corner of the screen and the current system time will be displayed in the upper right hand corner of the screen. Each of these fields are redisplayed once per second.

The SET CLOCK function is not available in multi-user OASIS.

SET DATE mm/dd/yy

This function requires privilege level 5 or greater and allows the user to change the system date. mm/dd/yy indicates that the user types the month, day, and year that he wishes the system date set to.

Leading zeros are not required and the slash (/) delimeter may be replaced with any of the following characters ./-+*:!"#$%&'=@`<>?

SET DOWN nn

This function is similar to the SET ESCAPE function except that it operates on the down-arrow key.

SET ERRNUM ON|OFF

This function requires privilege level 5 or greater and allows the user to set the error number display ON or OFF. When a system program detects an error the error is reported to the operator in one of three ways: error number and text; error number only; error text only. This function will either turn ON or OFF the error number display. If you are turning the error number display OFF and the error text display is already OFF then the error text display will be set ON. (It is not allowed to turn both methods of displaying errors OFF.)

SET ERRTEXT ON|OFF

This function requires privilege level 5 or greater and allows the user to set the error text display ON or OFF. When a system program detects an error the error is reported to the operator in one of three ways: error number and text; error number only; error text only. This function will either turn ON or OFF the error text display. If you are turning OFF the error text display and the error number display is already OFF, then the error number display will be set ON. (It is not allowed to turn both methods of displaying errors OFF.)

Refer to the Appendix on System Errors for details on error display.

SET ESCAPE nn

This function allows the user to set the value of the character to be interpreted as the escape character from the console keyboard. Some terminals use the escape character for their own uses and do not transmit the charcter to the host computer. By using this function you can designate another character to be used as the escape character, leaving the escape key for the exclusive use of the terminal. The value nn corresponds to the ASCII value of a key. For example if a CONTROL/Y is to be used the value is 25.

SET HISTORY ON|OFF

This function requires privilege level 5 or greater and allows the user to change the status of the switch that determines whether history records are to be created for each user logon and logoff. When history record creation is suppressed by setting this switch off the LOGON and LOGOFF commands operate somewhat faster because less work is done by them.

SET IMPEXEC ON|OFF

This function allows the user to change the "switch" that allows a command to be the name of an indirect command file (EXEC).

When IMPEXEC is set OFF, and the user wishes to execute an EXECutive procedure, the user must specify that he is doing so by preceding the name of the EXEC file with the command EXEC.

When IMPEXEC is set ON the user need not explicitly specify that he is executing an EXECutive procedure unless the name of the EXEC is the same as one of the OASIS commands.

For more information regarding the EXECutive procedure processor refer to the EXEC Language Reference Manual.

SET LEFT nn

This function is similar to the SET ESCAPE function except that it operates on the left-arrow key.

SET MEMORY <pid> [<length>]

This function requires a privilege level of 5 and allows you to define a partition and its size.

Memory partitions must be allocated in ascending order and deallocated in decending order. The memory used by the partition is taken from or given to the next lowest user. When multi-user OASIS is first booted into the system there is only one partition (number 1) that owns all of available memory (single user mode).

The SET MEMORY command is used to allocate or deallocate the users. For example, if three users are desired, each with 12 thousand bytes of memory the following commands would be given:

```
>SET MEMORY 2 24000

>SET MEMORY 3 12000
```

At this time there would be three, inactive users with the second two users having 12000 bytes of memory each and the first user having the rest of memory available.

The minumum partition size is 4096.

SET PRIV nn

This function allows you to lower the privilege level, thus restricting the commands and functions that you may perform. Normally, this function would only be used in single-user OASIS when the accounting feature is not used. (The accounting feature allows you to specify a privilege level for each account name.)

SET RDYMSG ON|OFF

This function allows the user to set the ready message display either ON or OFF. When it has been set ON the system will display a "ready message" after the execution of every program. This message consists of the return code, current

system time and the elapsed time of execution.

This feature of the system can be useful if the user is testing a program that uses the return code in a decision making process.

SET RIGHT nn

This function is similar to the SET ESCAPE function except that it operates on the right-arrow key.

SET RUBOUT nn

This function is similar to the SET ESCAPE function except that it operates on the character rubout or delete key.

SET SLICE nnn

This function requires privilege level 5 or greater and allows you to change the time slice value. The time slice value is the number of "ticks" (normally 1/1000 of a second) that each user will have control of the system.

SET SYNONYM [fn] [(option[)]]

This function allows the user to change the command synonym table.

The file name refers to the user defined synonym file. This file must exist before the user synonyms may be set ON. As described in the Appendix on System Files this file must have a file type of SYNONYM and be a sequential format file on the system disk. Specifying a filename causes an automatic CLEAR option to be performed on any previously defined user synonym file.

The allowed options for the SET SYNONYM command include the following:

NOSTD Indicates that the system defined synonyms are to be turned off. Since the system defined synonyms are only abbreviations of the system commands then this option effectivly sets ABBREV OFF for the system abbreviations but not the user defined abbreviations. This option has no effect on the user defined synonyms, if any.

STD Indicates that the system defined synonyms are to be turned on. This option has no effect on the user defined synonyms, if any.

CLEAR Indicates that the user defined synonyms are to be cleared. This option does not effect the status of the system defined synonyms.

SET TIME hh:mm:ss

This function requires privilege level 5 or greater and allows you to change the system clock. hh:mm:ss indicates that you are to type the hours, minutes, and seconds that you wish to set the system clock to. Typing the seconds is not required, nor are leading zeros. The colon (:) delimiter may be replaced with any of the following characters ./-+*:!"#$%&'=@`<>?

If you type an invalid or unrecongnizable time the SET command will reply:

Invalid format for time S.B (HH:MM:SS)

SET UP nn

This function is similar to the SET ESCAPE function except that it operates on the up-arrow key.

SET VERSION ON|OFF

This function allows the user to change the status of the version switch. When the VERSION switch is set on the program name, version number, and version date will be displayed upon entry to a program.

SET Examples

```
>SET DATE 5-10.78

>SET RDYMSG ON

RC = 0, 12:15:32, ET = 15.3 minutes

>SET RDYMSG OFF

>SE ABBREV OFF

>SE VERSION ON ; THIS SHOULD NOT ALLOW THE ABBREVIATION
SE.COMMAND not found.

>SET ABBREV ON

>SET ESCAPE 22

>
```

CHAPTER 33

SHARE COMMAND

The SHARE command allows you to designate that a file is to be shared by another account. The format of the SHARE command is:

SHARE <file-desc> [(<Type>[])]

Where:

file-desc Indicates the private file whose shared access is to be modified.

Type Indicates that the current shared access accounts to this file are to be displayed on the console.

When the TYPE option is specified a list of account names that have shared access to this file will be displayed on the console.

When the TYPE option is not specified the SHARE command will allow you to change the accounts that have shared access to the file specified.

The program will ask "Add user account?". If an additional account is to be allowed shared access to this file then enter the account name and the question will be asked again for additional accounts. When no account name is entered in response to this question the next question will be asked. Do not specify a system account name as an account to have shared access. When a file is shared with another account it is also shared with all of that account's synonyms.

The programs will ask "Delete user account?". If an account that currently has shared access to this file is to be denied access then enter the account name and the question will be asked again for additional accounts. When no account name is entered in response to this question the program will return control to the CSI. It is not necessary to delete shared access to synonyms of an account that is having its shared access deleted. This is done automatically by the method that the system uses for synonym accounts.

Example

```
>SHARE MYFILE.DATA:A

Add user account? ALPHA
Add user account? OMEGA
Add user account?
Delete user account? BETA
Delete user account?

>
```

CHAPTER 34

SHOW COMMAND

The SHOW command provides you with the ability to display the status of certain system parameters. The format of the SHOW command is:

SHOW <function>

SHOW Functions

The various functions available with the SHOW command include the following.

ABBREV Display the status of the abbreviation switch.

CANCEL Display the currently defined console keyboard line-cancel character.

CLKFRQ Display the current clock frequency adjustment constants.

CLOCK Display the system clock continuously. To exit from this mode you must use the system cancel-key.

Note: Your system may not support the time of day feature.

DATE Display the current system time and date.

DISK Display the status of all attached disks, or, when followed by a disk specification, the status of the specific disk.

DOWN Display the currently defined console keyboard down-arrow character.

ERRNUM Display the status of the error number switch.

ERRTEXT Display the status of the error text switch.

ESCAPE Display the currently defined console keyboard escape character.

HISTORY Dispay the status of the history switch.

IMPEXEC Display the status of the implied EXEC switch.

LEFT Display the currently defined console keyboard left-arrow character.

MEMORY Display the the current allocation and usage of memory. The information displayed includes the partition number (Port), partition start address (MEMLO), partition end address (MEMMX), current partition high memory (MEMHI), current partition stack pointer (SP), current partition execution address (PC), partition size, in decimal (Size), command name being executed by partition (Routine), user name logged onto partition (Username).

PRIVLEV Display your current maximum privilege level.

RDYMSG Display the status of the ready message switch.

RIGHT Display the currently defined console keyboard right-arrow character.

RUBOUT Display the currently defined console keyboard rubout character.

SERIAL Display the serial number of the operating system.

SLICE Display the current system time slice value.

SYNONYM Display the synonyms currently defined.

TIME Display current system time and date.

Note: Your system may not support the time of day feature.

UP Display the currently defined console keyboard up-arrow character.

USERS Display the users that are currently logged onto the system. The information displayed includes the partition numberr, account name, and logon date and time.

VERSION Display the version number, version date, and the privilege value of all command programs on the specified disk. When no disk is specified the S disk is assumed.

WHO Display the account name of the account that you are currently logged onto.

* Display the current time and date, the memory size of your partition, and the status of: ABBREV, RDYMSG, IMPEXEC, ERRNUM, ERRTEXT, HISTORY, VERSION, ESCAPE UP, DOWN, LEFT, RIGHT, RUBOUT, CANCEL, SYNONYM, SLICE, PRIVLEV, MSG, SERIAL, and CLKFRQ.

Note: Your system may not support the time of day feature.

SHOW Examples

The following examples illustrate the uses of the SHOW command.

```
>SHOW RDYMSG
RDYMSG = OFF

>SHOW ABBREV
ABBREV = ON

>SHOW IMPEX
IMPEXEC = ON

>SHOW DISK S

S(1) Label: "SYSTEM  ",WP,
     Backup onto "SYSTEM$B" on 08/09/78 13:12,
     Capacity: 308K bytes, (77-1-16),
     Available: 142K bytes (46%),
     Largest area: 140K bytes,
     48 files is use (out of 112).
```

The information displayed by the SHOW DISK function is:

Line-1 Directory label; drive number; disk label; write protect status.

Line-2 Backup disk label; backup date and time.

Line-3 Disk capacity; number of cylinders; number of heads; number of sectors per track.

Line-4 Amount of storage available; percent available.

Line-5 Largest amount of contiguous storage available.

Line-6 Number of files in directory; total number of directory entries allocated. This is a reflection of all files on the disk, not just the files accessible from the current account.

The SHOW DISK function, in addition to displaying the above information, also validates the allocation of the disk. If any misallocation is detected an error message is displayed. Negative misallocation indicates that more space has been allocated for file storage than is actually in use. Positive misallocation indicates that less space has been allocated than is actually in use. These are dangerous conditions - the disk should not be used until the misallocation is corrected. The positive misallocation indicated that at least two files are sharing the same disk space. Any misallocation may be corrected by copying all of the files to a blank disk, one file at a time. If positive misallocation existed then one or more files may contain invalid data.

The following version numbers and dates do not necessarily reflect the current version number for the system programs.

```
>SHOW VERSION

Module    Version Date     Priv
ACCOUNT   5.3     18FEB79   5
ASSIGN    5.3     04JAN79   0
ATTACH    5.3     04JAN79   0
BACKUP    5.3     04JAN79   3
BASIC     5.3     06JAN79   2
BUILD     5.3     04JAN79   2
CHANGE    5.3     21JAN79   3
COPYFILE  5.3     19FEB79   1
CREATE    5.3     05FEB79   1
DEBUG     5.3     26DEC78   4
DUMPDISK  5.3     04JAN79   2
EDIT      5.3     19FEB79   2
ERASE     5.3     05FEB79   1
FASTCOPY  5.3     05JAN79   3
^
```

```
>SHOW SYNONYM

System    User     Shortest
Command   Synonym  Form
--------  -------- --------
ASSIGN             ASS
ATTACH             AT
BACKUP             BAC
COPYFILE           C
CREATE             CR
DUMPDISK           DU
EDIT               E
ERASE              ER
EXEC               EX
^

>SHOW *

SHOW ver 5.4 220CT79

11:16:20 Friday, October 26, 1979

NAME = SYSTEM
ABBREV = ON
RDYMSG = OFF
IMPEXEC = ON
ERRNUM = OFF
ERRTEXT = ON
HISTORY = ON
VERSION = ON
MSG = ON
ESCAPE = 27
UP = 26
DOWN = 10
LEFT = 8
RIGHT = 6
RUBOUT = 127
CANCEL = 24
SYNONYM = (STD)
SLICE = 100
PRIVLEV = 5
SERIAL = 3-1
CLKFRQ = 10,100

>SHOW MEMORY

Port MEMLO MEMMX MEMHI SP--- PC--- Size  Routine Username
     0000H A886H                   19712 NUCLEUS
  1* A580H FDFFH FB21H FAF3H A580H 22656 SHOW    SYSTEM
  2  4D00H A57FH A2A2H A27AH A2C8H 22656 CSI     DEMO
```

```
>SHOW MEMORY

Port MEMLO MEMMX MEMHI SP--- PC--- Size  Routine Username
     0000H 4CFFH                   19712 NUCLEUS
     4D00H A886H                   23431 BASIC
  1  D087H FDFFH FB21H FAF3H D087H 11641 CSI     TOM
  2* A887H D086H CDA9H CD87H CDCFH 10240 SHOW    PAYROLL

>SHOW USERS

Port Username Log-Date Log-Time
  1* SYSTEM   10/23/79 08:23:15
  2  DEMO     10/23/79 10:55:12
  3  PAYROLL  10/23/79 08:30:20

>
```

CHAPTER 35

SPOOLER COMMAND

The SPOOLER command is an optional OASIS product and may not be included in your system.

The SPOOLER command allows you to control the system's print spooler. Not all systems that support the OASIS operating system can use the print spooler, only those systems that have hardware interrupt capability. An indication of hardware interrupt capability is the existence of the time of day clock in OASIS. The format of the command is:

SPOOLER <function>

Where function is one of the following:

INIT Indicates that the system print spooler is to be loaded and initialized. In order for the print spooler to be initialized the PRINTER1 device must be already attached and the file SYSTEM.SPOOL$$1 must exist, on-line. When the spooler is initialized and not suspended (see below) all output directed to the PRINTER1 device will go through the print spooler.

STATUS Indicates that the current status of the print spooler is to be reported. One of several messages may be displayed, including:

Spooler is not initialized.
Spooler is busy printing.
Spooler is waiting for work.
Spooler is waiting for printer.
Spooler is suspended.

SUSPEND Indicates that the system print spooler is to be suspended, after the current report being printed by the spooler is finished. Once the spooler has been suspended all further output directed to the PRINTER1 device will output immediately to that device. The spooler remains in memory.

RESUME Indicates that the system print spooler is to resume its function. This function is only effective when the spooler is currently suspended. When the spooler is resumed, any reports previously suspended will start to print.

ABORT Indicates that the current report being printed by the spooler is to be aborted. The message "Aborted" will be printed on the PRINTER1 device when this is done, whether or not there was a report actually aborted.

The OASIS system print spooler is a queing logical device. Once the spooler has been initialized and is not currently suspended, all output directed to the physical device that PRINTER1 is attached to will be "spooled". This means that if PRINTER1 is attached to SIO1 and if COMM1 is attachad to SIO1 all output directed to PRINTER1, COMM1, or SIO1 will be spooled. For this reason it is best to have no other logical devices attached to the physical device that PRINTER1 is attached to.

The print spooler can only be "un-initialized" by re-booting the system.

Do not use the print spooler for reports that require operator attention during the printing process as the spooler is a "background" task that can only be interrupted between reports, not in the middle of a report. This specifically excludes reports

that require a form alignment pattern to be generated, reports that print on non-continuous paper (must pause between pages), etc.

As stated previously, not all OASIS systems have a print spooler. Even if your system does have the print spooler it may be ill-advised to use it. In general, only use the print spooler in systems whose disk access is considerably faster than the printer being used. This means that if you have a slow disk or a fast printer it will be inefficient to use the spooler.

Some situations do arise where the spooler can be convenient, even though technically inefficient such as: listing several files followed by editing. Since the edit process does not use the disk (after the program and file are loaded into memory) the disk and printer are idle during the editing process. If the file listings are directed to the print spooler first then the editing can be done simultaneous to the spooler's output of the file listings.

CHAPTER 36

START COMMAND

The START command activates a user partition. The format of the command is:

START <pid> (<indev> [<outdev> [<device options>]][)]

Where:

<pid> Indicates the partition identification number of the user partition to be started. This partition must have been previously defined by a SET MEMORY command.

<indev> Indicates the physical device name to be used as the CONIN device for the user.

<outdev> Indicates the physical device name to be used as the CONOUT device for the user. This specification may be omitted which will indicate that the CONOUT device is the same as the CONIN device.

<device options> Indicates the options to be used on the console device. All of the options listed in the ATTACH command description may be used.

When the START command is executed the user on that partition is asked to logon. If the user does not have is console turned on yet he may enter a carriage return after turning his console on and the logon message will be re-displayed.

START Example:

```
Multi-user OASIS version v.r 64K

Time (HH:MM:SS) 815

Date (MM/DD/YY) 1030

08:15:00 Monday, October 30, 1979


Logon Please: IPL

>SET MEMORY 2 10240

>START 2 (SIO2 B9600 L63 P15 FF6 C4

>
```

CHAPTER 37

STATE COMMAND

The STATE command provides a quick means of determining the existence of a file or group of files. The format of the STATE command is:

STATE <file-desc> [<option>...]

Where:

<file-desc> Indicates the description of the file or files to be "stated". Wildcards are acceptable.

STATE Options

TYPE Indicates that the number of files found and their total allocation is to be displayed on the console. This is the default option.

NOTTYPE Indicates that nothing is to be displayed. This option is useful when the STATE command is executed from an EXEC and it is necessary to know whether a file exists or not. A return code of 0 indicates that the file does exist and a return code of 12 indicates that the file does not exist.

PUBLIC Indicates that the public files (owned by the system) are to be included in the search.

When the STATE command is executed one of two messages will be displayed. If no file can be found that matches the file-desc the message "No files found" will be displayed and the system return code will be set to 12. If one or more files can be found that match the file-desc then the message "nn files, nnn blocks" will be displayed. The nnn blocks indicates the number of blocks of disk storage used by the files qualifying. (Return code set to 0.)

The STATE command operates only on files accessible to the current account. When wildcards are used the public account files will be included in the file count and number of blocks of storage used. When the file description is explicit the public account files will only be included if a match cannot be found in the private and shared files. This means that an explicit file description of a file that is both in the private or shared account and also is a file in the public account will result in only the private or shared file being reported.

STATE Examples:

```
>STATE * COMMAND S
42 file(s), 219 block(s).

>STATE * BACKUP S
No files found.

>
```

CHAPTER 38

STOP COMMAND

The STOP command deactivates a user partition. The format of the command is:

STOP <pid>

Where:

pid Indicates the partition identification number of the user partition to be deactivated.

The STOP command can only be used on a partition that has logged off (see LOGOFF command).

The STOP command does not release the memory used by the partition--the SET MEMORY command does that.

CHAPTER 39

SYSGEN COMMAND

The SYSGEN command allows you to save the status of certain system parameters for future initial system loads. The format of the command is:

SYSGEN

The specific parameters that the SYSGEN command saves include all ATTACHments of logical to physical devices along with their options, and the status of the following SETable switches:

CLOCK
CLKFRQ
RDYMSG
ABBREV
SYNONYM
HISTORY
ERRNUM
ERRTEXT
IMPEXEC
VERSION
PRIVLEV
ESCAPE
UP
DOWN
LEFT
RIGHT
RUBOUT
CANCEL
Console Display-speed
Console Screen Wait

The SYSGEN command requires that the SYSTEM.NUCLEUS file not be write protected (see RENAME command).

Multi-user OASIS SYSGEN will save the SLICE value but will not save the CLOCK switch or the PRIVLEV value.

In multi-user OASIS the data saved by the SYSGEN command only applies to the partition 1 user; however, some of the data saved is global in nature and therefore is saved for all users: CLKFRQ, DATE, ERRNUM, ERRTEXT, HISTORY, SLICE, and TIME.

CHAPTER 40

TEXTEDIT COMMAND

The OASIS TEXTEDIT command allows you to create and maintain files for use by programs such as the EXEC language processor, the BASIC compiler/interpreter, the MACRO assembler, the SCRIPT processor, etc. Although the Editor is generally used to create or maintain files for these processors there is no restriction on its use: you can maintain files to be used by your own programs.

A file that is created or maintained by the TEXTEDIT program for use by another system program generally contains a mixture of commands and data to that program. These commands or data should not be confused with the commands that the TEXTEDIT program uses. For example, the operator may wish to create a file of commands and data to be used by the EXEC processor. All of these commands and data are treated as text to the TEXTEDIT program.

The OASIS TEXTEDIT command is a file editor with commands that allow you to change, add or delete text from a file. All of these commands are oriented to the user, that is, these commands are English words whose meanings indicate the function that they perform in the Editor. For instance, the command that indicates that you wish to locate the next occurence of a sequence of characters is "LOCATE".

All of the commands may be abbreviated to the first character except the FILE and QUIT commands which might cause unrecoverable results.

The OASIS TEXTEDIT is a line oriented editor. This means that after a command is executed the text pointer is positioned at the beginning of the current line and the commands reference entire lines. For instance the "DELETE" command deletes lines of text; the "INPUT" command inputs lines of text; the "PAGE" command types lines of text; etc. There is a command available that allows you to use this editor as a character oriented editor. This is the MODIFY command.

Caution: Do not allow a CTRL/Z character to be placed in a file. This character is interpreted by most programs as meaning the end-of-file and will probably cause loss of data if placed in a file by an operator or user program.

40.1 Invoking the OASIS TEXTEDITOR

To enter and use the OASIS text editor you use the CSI TEXTEDIT command in the following format:

TEXTEDIT <file-desc>

When this command is executed the TEXTEDIT program is first loaded into memory and the file description is passed to the program. The program then searches the directory (specified or default search sequence) and, when the file is found, reads the entire file into memory. If the entire file cannot fit into memory due to the amount of memory available the Edit program will display the message: "Available Memory Now Full:" followed by the last line of text that it was able to read. If this line of text is not the last line of text in the file then the operator should abort the edit session and reduce the size of the file by manipulating it with the COPYFILE command.

If memory is filled up when the file is read in, and the last line of text is the line that is displayed after the error message, the operator may continue the edit session if he first deletes some of the text lines before adding any new text.

When the file is not found on the specified directory or the default search directories then the edit program will display the message "New File" before displaying the prompt character.

Due to the fact that the disk image of a file being edited is not updated until the operator FILEs the memory image of the file the System Cancel command is redefined in the edit session. If this key were not redefined then it is possible that hours of work may be lost by the inadvertant typing of this command. To abort an edit session the operator uses the QUIT command.

TEXTEDIT Prompting Character

After the TEXTEDIT command has been executed an asterisk (*) will be displayed on the left side of the console terminal. This is the prompting character for the TEXTEDIT program and indicates that the TEXTEDIT program is waiting for a command.

TEXTEDIT Modes

The OASIS TEXTEDIT has two modes of operation: the command mode, which is indicated by the edit prompting character on the left side of the screen, and the text input mode, which is indicated by no prompting character on the left side of the screen.

TEXTEDIT Commands, General

Most of the edit commands have parameters following the command word. These parameters tell the edit command interpreter what the operator wishes to do specifically.

Parameters to the edit commands may be of two types: numeric and string. Numeric parameters are always assumed to be decimal (base 10); string parameters are always enclosed within delimiters. A delimiter is a character that indicates the beginning or end of something. For more versatility the OASIS editor allows many characters to be string delimiters. The only restriction being that the delimiting character may not be a space character or a question mark character, and the delimiter may not be part of the string. The terminating delimiter, when used, must be the same as the beginning delimiter. For documentation purposes the slash character (/) will be used for the string delimiter.

If, while typing a line of text or command, the operator should wish to cancel what he has entered, he may type either the program cancel-key or the line cancel-key. This will abort the line being typed with no change to the text file.

40.2 Bottom Command

The BOTTOM command allows the operator to position the pointer to the end of the text. The format of the BOTTOM command is:

BOTTOM

The BOTTOM command will position the pointer to the line before the end of file marker, and display the line. If there are no lines of text in the file then the BOTTOM command will display TOF: indicating that the line before the end of file marker is the top of file marker.

40.3 Case Command

The CASE command instructs the editor on whether to 'fold' the input from the keyboard to uppercase, lowercase or to accept the input as is.

The format of the CASE command is:

CASE [mode]

Where:

Mode	Meaning
U	'Fold' or change all alphabetic input from the keyboard to upper case. This is the default CASE mode.
M	Accept all input from the keyboard with no translation of case mode. This is the default case mode for all non-program file types.
L	Accept all input from the keyboard with inverse translation. This is the inverse of the CASE mode M. All alphabetic characters typed from the keyboard as text input are translated to their inverse case before display. This feature is useful when the console keyboard does not have a shift lock key.

40.4 The Carriage Return Command

The carriage return command is provided as a quick and easy means of advancing the text pointer to the next line of text in the file. It is identical in effect to the DOWN 1 command. The format of the carriage return command is simply to enter a carriage return after the edit prompt character.

40.5 Delete Command

The DELETE command allows the operator to delete whole lines of text from the file. The format of the DELETE command is:

DELETE [<nnn>]

Where:

<nnn>	Indicates the number of lines to be deleted, including the current line.
<blank>	When no parameter is specified after the command DELETE then only the current line is deleted.

40.6 Down Command

The DOWN command allows the operator to position the pointer down to the next line. The format of the command is:

DOWN

When the DOWN command is executed the next line in the text file becomes the current line and is displayed on the console.

40.7 Down Arrow Command

The down arrow command provides a quick and easy means of specifying that the text pointer is to advance one line of text. When the operator types the down arrow key the editor will interpret it as the DOWN command, display the message DOWN, and advance the text pointer one line of text. Some terminals label the down arrow key as "line feed" or "LF".

40.8 File Command

The FILE command allows the user to terminate an TEXTEDIT session normally by updating the file and returning to the CSI. The format of the FILE command is:

FILE [<file-desc>]

Where:

<file-desc> Indicates an optional file description. When the <file-desc> is specified, only as much as is necessary need be entered, the omitted parameters will default to the current file description.

When <file-desc> is specified the current file description is not changed but the <file-desc> specified is the description used by the FILE command.

When no <file-desc> is specified after the FILE command the current file description is used. In either case when the FILE command is executed the editor writes the file in memory to the disk file. If the editor finds an existing file on the specified disk with the same name and type as that being filed, the file type of that file is renamed to BACKUP, erasing any other file by that same description. When the editor has successfully "filed" the text file the file description that it was saved as is displayed on the console terminal along with the message "filed". Control returns to the CSI.

40.9 Input Command

The INPUT command allows the user to add new text to the file. The format of the INPUT command is:

INPUT

The INPUT command allows multiple lines of text to be added with each line added after the current line. As each line is added it becomes the the current line.

When the editor is accepting multiple lines of text no prompting character is displayed on the left side of the console. This is because the prompting character indicates that the editor is ready to accept a command and in this case it is not accepting commands but text.

To return to the edit command mode the user merely types a carriage return with no text preceding it.

40.10 Locate Command

The LOCATE command allows the user the ability to locate and position to the next occurrence of a sequence of characters. The format of the LOCATE command is:

LOCATE [/<string>[/]]

Where:

<string> indicates the sequence of characters that the editor is to search for. The search begins with the line following the current line. If the sequence of characters is found then the text pointer is positioned to the line that contains that string, and the line is displayed. If the sequence of characters is not found then the editor displays the message: "Not Found:" followed by the command that is was trying to execute. The text pointer is not changed when the string is not found.

<blank> Indicates that the last valid LOCATE command is to be executed again.

40.11 Modify Command

The MODIFY command is the only command that allows the editor to act like a character oriented text editor, and when the MODIFY command is being executed there is a new set of commands available to the user. The format of the MODIFY command is:

MODIFY [<nnn>|*]

Where:

<nnn> Indicates the number of lines that are to be modified.

* Indicates that all lines starting with the current line to the end of the text file are to be modified. To exit from this mode before the end of the file is reached you must use the Program Cancel-key.

<blank> When no parameter is specified then the default value of one is used indicating that the current line is to be modified.

When the MODIFY command is executed the line to be modified is displayed on the console terminal and the cursor is positioned at the first character position. The editor is now in a character oriented mode and the MODIFY command makes full use of the fact that the console terminal is normally a CRT with cursor positioning controls. The cursor indicates the current text pointer position in the line. Any changes made to the line are immediately indicated by the display of the line and the position of the cursor.

Any control characters imbedded in the line are expanded for display purposes to two characters: an up arrow character (^) followed by the ASCII representation of the control character. For example, a CTRL/I is displayed ^I. Even though the control character is displayed as two characters it actually is only one character. When a control character is deleted from the line both characters are erased from the display. When a control character is skipped over, both characters are skipped, etc.

The commands available to you when the editor is modifying a line are one character commands, but still the one character is the first character of the word that it is an abbreviation for. When possible, it is the same character that would be used in the edit command mode.

The following sub-sections discuss the commands available from the MODIFY command. When you type a command it is not displayed as that would disrupt the display of the line being modified. Any effect that a command has on the text in the line is immediatly displayed however.

Insert Characters Subcommand (I)

To insert new characters into the line the user types the insert command (I). Any characters typed after the I has been typed are added to the line before the current character. As each character is added to the line the remainder of the line is re-displayed.

To exit from the Insert character command you type a carriage return.

While in the insert character command you may backup one character position by typing the RUBOUT key. This backs the text pointer up one position, the cursor is backspaced, and that character is deleted. It is possible to backspace past the position that the insert command was given.

Delete Character Subcommand (D)

To delete a character from the line you use the delete character command (D). Every time a D is typed the current character is deleted from the line and the character is erased from the screen. This is an immediate command.

Replace Characters Subcommand (R)

To replace characters in the line you use the replace character command (R). After the R command has been typed each character that is typed replaces the current character and the text pointer is advanced one character position.

To exit from the replace character command you type a carriage return.

While in the replace character command, you may backup a character position by typing the RUBOUT key. This will back the text pointer up one position and the cursor will backspace. No characters are deleted and the editor is still in the replace character command. It is possible to backspace past the position that the replace character command was typed and still remain in the replace character mode.

Advance Character Subcommand ( )

To advance the cursor and text pointer one position you type the advance character command ( ), a space. When the space is typed the text pointer and the cursor are advanced one character position. You may not advance past the end of the line, however you may insert new characters at the end of the line or replace characters at the end of the line.

The right arrow has the same effect as the space key and is more graphic in its meaning. Both may be used interchangeably.

Find Character Subcommand (F)

To advance the cursor and text pointer to the next occurrence of a specific character you use the find character command (F) followed by the character to be positioned to. When the F is typed followed by another character the cursor is advanced to the next occurrence of that character in the line being modified. The

character specified must be entered in the same case as the character to be found. When the character can not be found the cursor will be positioned to the end of the line.

Backspace Character Subcommand (RUB)

To back the text pointer and cursor one character position you type the backspace character command. This command is any of the keys defined as the character delete keys, such as left-arrow, CTRL/H or RUB. The left arrow is more graphic in its meaning and is usually the key that is used for the RUBOUT. When this key is typed the text pointer and the cursor are backed up one character position.

Uppercase Character Subcommand (U)

To change the current character to uppercase you type the uppercase character command (U). When the U character is typed the current character is translated into uppercase, redisplayed and the text pointer and cursor are advanced one character position.

Lowercase Character Subcommand (L)

To change the current character to lowercase you type the lowercase character command (L). When the L is typed the current character is translated to lowercase, redisplayed and the text pointer and cursor are advanced one character position.

Beginning of Line Subcommand (B)

To position the cursor and text pointer to the beginning of the line being modified you use the beginning of line command (B). When the B is typed the cursor is positioned to the first character in the line.

End of Line Subcommand (E)

To advance the cursor and text pointer to the end of the line you type the end end of line command (E). When the E is typed the cursor and text pointer are advanced to the end of the line.

This command allows the user to easily add or change text at the end of the line.

Quit Subcommand (Q)

To quit the modification of the line and restore any changes made to the line you type the quit modify command (Q). When the Q is typed the line is re-displayed as it was before any changes were made and the modification of the line is terminated. If there are any iterations of the MODIFY command remaining then the text pointer advances one line and the next line is placed in the MODIFY mode.

End Modify Subcommand (RET)

To end the modification of the line and save any changes made to the line you type the end modify command (carriage return). When the carriage return is typed the line is re-displayed with all changes saved and the modification of the line is terminated. If there are any iterations of the MODIFY command remaining then the text pointer advances one line and the next line is placed in the MODIFY mode.

40.12 Page Command

The PAGE command allows the user to display a page of text and to position the text pointer to the end of the next page of text. The format of the PAGE command is:

PAGE

The PAGE command will display one screen of text.

The first line of text displayed by the PAGE command is the current line. The last line of text displayed on the screen by the PAGE command is determined by the class of terminal designated by the OASIS ATTACH command. For the CRT this would be 15 lines of text. At the end of the execution of the PAGE command the text pointer will be positioned to the beginning of the last line displayed. This means that if two PAGE commands are typed consecutively then the first line displayed by the second PAGE command is the last line displayed by the first PAGE command. In this manner it is easy to observe the flow of text in spite of the page break.

40.13 Quit Command

The QUIT command allows the user to abort an edit session without updating the file on the disk. This may be necessary for many reasons such as specifying the wrong file in the TEXTEDIT command or the operator has decided that the changes made to the file should not be saved. The format of the QUIT command is:

QUIT [<return code>]

When the QUIT command is executed with no return code following, the editor will set the return code to zero and return control to the environment that invoked the TEXTEDIT command (CSI or EXEC) without updating the file. When the return code is specified it must be a numeric value between 0 and 255. The value is assigned to the system return code and control is returned to the environment that invoked the TEXTEDIT command.

40.14 Top Command

The TOP command allows the operator to position the text pointer to the beginning of the text file. It has the inverse effect of the BOTTOM command. The format of the TOP command is:

TOP

Upon execution of the TOP command the text pointer will be positioned to the top of file marker which is before the first line of text. The editor will display the message: "TOF:". This command is the only command that will allow the operator to enter lines of text before the first line of text. Refer to the INPUT command.

40.15 Up Command

The UP command allows the operator to position the text pointer backward in the text file. The effect of the UP command is the reverse of the DOWN command. The format of the UP command is:

UP [<nnn>]

Where:

<nnn> Indicates that the editor is to position up nnn lines from the current text pointer. If the value of nnn is greater than the number of lines that precede the current line then the text pointer will be postioned to the top of file marker and the message: "TOF:" will be displayed.

<blank> When no parameter is entered after the UP command then the last UP /string/ command is used. If there have been no prior UP /string/ commands executed then UP 1 is used.

40.16 Up Arrow Command

The up arrow command allows the user a quick and easy means of specifying that the text pointer is to back up one line of text. When the operator types the up arrow key the editor will interpret it as an UP 1 command, display the message UP 1, and back the text pointer up one line of text. If the console terminal does not have an up arrow key then the user may use the CTRL/Z key instead.

40.17 TEXTEDIT Command Summary

```
BOTTOM    Locate end of file.
CASE      Set upper, lower, or mixed.
DELETE    Remove lines of text.
DOWN      Advance line pointer.
FILE      Save text file on disk and return to system.
INPUT     Accept text from keyboard.
LOCATE    Locate specified string in text.
MODIFY    Allow character-by-character change of current line.
            B     Position to the beginning of the line.
            D     Delete one character.
            E     Advance to end of line.
            F     Find and position to specific character.
            I     Insert characters.
            L     Convert to lower case.
            Q     Quit modification - no changes made.
            R     Replace characters.
            U     Convert to upper case.
            <sp>  Advance one character.
            <DEL> Backup one character.
PAGE      Display one screen of text on console.
QUIT      Abort this edit session.
TOP       Locate top of file.
UP        Position line pointer backwards in file.
<CR>      Advance line pointer one line.
<up-arrow> Position line pointer up one line.
<down-arrow> Position line pointer down one line.
```

40.18 TEXTEDIT Errors and Messages

Message Explanation

Not Sequential Error message - occurs when an attempt is made to TEXTEDIT an indexed or direct file.

Invalid command syntax Error message - occurs during edit command mode. This error message indicates that the operator has typed a valid command name but has used the wrong parameters or has made a typing error.

Unrecognized command: Error message - occurs during edit command mode. This error is usually caused by the operator typing text when the editor is expecting a command. Also occurs when a command abbreviation is used and not separated by a space from text that follows it.

Too many parameters: Error message - occurs during edit command mode.

Invalid numeric Error message - occurs when the command syntax requires a numeric parameter but the operator has entered a alpha character.

Space required following command Error message -

Required parameter missing: Error message - occurs when the operator has not entered all of the parameters required by the command syntax.

Must be U, L, or M Error message - occurs during CASE command and operator has not specified a valid CASE mode.

Invalid filename Error message - occurs when operator has specified a file description. File names must start with an alphabetic character, contain only alphbetic, numeric or the $ character and can be no more than eight characters in length.

fn.ft:fd filed Information message - displayed after the editor has successfully FILED the text file.

File Name Missing Error message - occurs when operator has specified a file description with a missing name field.

File Type Missing Error message - occurs when the operator has specified a file description with a missing type field.

NEW FILE: Information message - displayed when the TEXTEDIT program is first entered and the specified file to be edited is not found.

EDIT: Information message - occurs when the editor re-enters the edit command mode after the INPUT command is exited.

INPUT: Information message - displayed when the editor enters a multipla Life input or replace mode.

EOF: Information message - displayed when the editor encounters the end of file marker.

TOP: Information message - displayed when the editor encounters the top of file marker.

Not Found: Information message - displayed when the editor can not LOCATE the sequence of characters specified.

No Room: Error message - occurs when the editor detects that there is insufficient memory available to save the line of text just entered by the operator.

End of Memory Reached: Information message - occurs when the editor detects that there is probably insufficient memory available to add more lines of text.

DOWN Information message - displayed in response to the operator typing the down arrow in the command mode.

UP Information message - displayed in response to the operator typing the up arrow in the command mode.

Available Memory Now Full: Information message - occurs during input of text or when TEXTEDIT is loading the file into memory. Indicates that the text was accepted and saved in memory but there is not sufficient space available for any more text input.

The operator should delete some text to make space available or divide the file into multiple files (if the purpose of the file will allow multiple files).

Truncated: Error message - the operator has INPUTed, CHANGEd, or MODified a line whose new length is greater than 255. The line has been truncated to 255 characters and is displayed following the error message.

BACKUP File is Protected Error message - Indicates that the editor cannot rename the input file to BACKUP because a BACKUP file already exists and it is delete protected.

Can't save a BACKUP File Error message - An attempt was made to file or save a file with file type BACKUP.

Disk Full Error message - During an attempt to SAVE or FILE the file being edited the disk became full. When this occurs the input file has already been renamed to file type BACKUP. Use the CSI command to erase some of the unused files on the disk or file it with a differnt name designating that a different disk drive is to be used.

CHAPTER 41

UNLOAD COMMAND

The UNLOAD command unloads a re-entrant program from memory, freeing up its memory space for general usage. The format of the command is:

UNLOAD <file-desc>

Where:

<program-name> Is the name of the re-entrant program to be unloaded from memory.

The UNLOAD command may only be used in single-user mode (only one partition defined).

To make the memory used by the re-entrant program available when more than one re-entrant program is loaded the programs should be UNLOADed in the reverse sequence that they were loaded in; otherwise the memory used by a program may not be returned to the system.

(This page intentionally left blank)

APPENDIX A

CSI COMMAND SUMMARY

Command	Function
ACCOUNT	Maintain the user account names file.
ASSIGN	Assign files or devices to I/O channels.
ATTACH	Make a disk or device available to OASIS.
BACKUP	Copy the entire contents of an OASIS disk to another disk.
BASIC	(Optional) Execute or maintain an OASIS BASIC language program.
CHANGE	Change the privilege value of a program.
COPYFILE	Copy a file (or portion of a file) to another file (append, replace, translate, etc.)
CREATE	Create a new direct or indexed sequential file.
DEBUG	(Optional) Perform dynamic debugging of a machine language program.
DUMPDISK	(Optional) Display the physical contents of a disk file.
EDIT	(Optional) Create or modify an ASCII sequential file.
ERASE	Delete a file or files from a disk.
EXEC	Execute an EXECutive language program.
FILELIST	Display the files and their attributes on the terminal, printer, or to a disk file.
FILT8080	(Optional) Translate Intel mnemonic assembly language source program to Zilog mnemonic assembly language source program.
FORCE	Force another user to execute a command.
GETFILE	(Optional) Copy a file from a non-OASIS disk.
INITDISK	Initialize a disk format, directory, or label.
INTELHEX	(Optional) Convert INTEL hex object code to OASIS object code.
LINK	(Optional) Perform translation and editing of object programs to load image programs.
LIST	Display the contents of a file on the terminal or printer.
LOAD	Load a re-entrant program for subsequent usage.
LOGOFF	Sign off of the current user account.
LOGON	Sign on to a different user account.
MACRO	(Optional) Translate a Zilog mnemonic assembly language source program to an object program (assemble).
MAILBOX	Retrieve information from other users.
MEMTEST	(Optional) Perform diagnosis of memory.
MOUNT	Allow a change of disks.
MSG	Send a message to another user.
OWNERCHG	Change the ownership of a file to another account.
PATCH	Update disk image of assembled program file.
PEEK	Peek at another user's console output.
RECEIVE	(Optional) Accept a file from another OASIS system.
RECOVER	(Optional) Restore lost file to it's directory.
RELOCATE	(Optional) Create a relocatable command from an absolute command.
RENAME	Change the name or type of an existing file.
REPAIR	(Optional) Detect and correct disk mis-allocation.
RUN	(Optional) Execute an OASIS BASIC compiled program.
SCHEDULE	Schedule a task for future execution.
SCRIPT	(Optional) Process and output an OASIS SCRIPT language file.
SECTOR	(Optional) Display sector linkages of a sequential file.
SEND	(Optional) Transmit a file to another OASIS system.
SET	Change OASIS system parameters.
SHARE	Designate a file(s) to be accessible from other user accounts.
SHOW	Display OASIS system parameters.
SPOOLER	(Optional) Change or display status of the printer spooler.

START	Start another user partition.
STATE	Determine existence of a file(s).
STOP	Stop another user partition.
SYSGEN	Permanently save SET and ATTACH status.
TERMINAL	(Optional) Emulate terminal to foreign system.
TEXTEDIT	Create or modify a text file.
UNLOAD	Unload a previously loaded re-entrant command.
VERIFY	(Optional) Read an entire disk and report any errors detected.

APPENDIX B

DEVICE NUMBER ASSIGNMENTS

The following numbers, names, and synonyms are the only logical device designations that the ATTACH command recognizes. All programs and commands communicate with peripherals by means of these logical device designations. The actual device that these are associated with is normally unknown (and irrelevant) at execution time. Adding additional physical device drivers to a system does not increase the number of logical devices in the system.

Logical Device Numbers

Number		Device-Name	Synonym
		DUMMY	
0		DISK-S	S
1		DISK-A	A
2		DISK-B	B
3		DISK-C	C
4		DISK-D	D
5		DISK-E	E
6		DISK-F	F
7		DISK-G	G
8	*	CONIN	CONI
9	*	CONOUT	CONO
10		READER	RDR
11		PUNCH	PUN
12		PRINTER1	PRINTER, PRT, PRT1
13		PRINTER2	PRT2
14		PRINTER3	PRT3
15		PRINTER4	PRT4
16		COMM1	COMM, COM, COM1
17		COMM2	COM2
18		COMM3	COM3
19		COMM4	COM4
20		TAPE1	TAPE, TAP, TAP1
21		TAPE2	TAP2
22		TAPE3	TAP3
23		TAPE4	TAP4

* The devices CONIN and CONOUT may be treated as a single device by using the name CONSOLE.

APPENDIX C

INDEXED & DIRECT FILE ACCESS

As explained in the description of the CREATE command indexed files require a filesize that is dependant upon an algorithm used in accessing the file. The filesize entered in the CREATE command may be modified by the system in order to optimize access to that file. Table D-1 is a partial list of the filesizes used by the system for indexed file access. It is provided here to inform the programmer of the filesize that was used in creating an indexed file. Using this information the programmer can calculate the amount of disk space required by the file without actually creating it.

Indexed file access response time is relatively constant until the file is approximately 70-80% full. When a file exceeds this load factor the response time degrades to unacceptable times. For this reason indexed files should be over-allocated to insure that the response time will not degrade.

It should be noted that, although the average response time degrades as the file fills, the response time for a particular record remains constant. For this reason, records that will be accessed more frequently than others should be added to the file when the file is nearly empty.

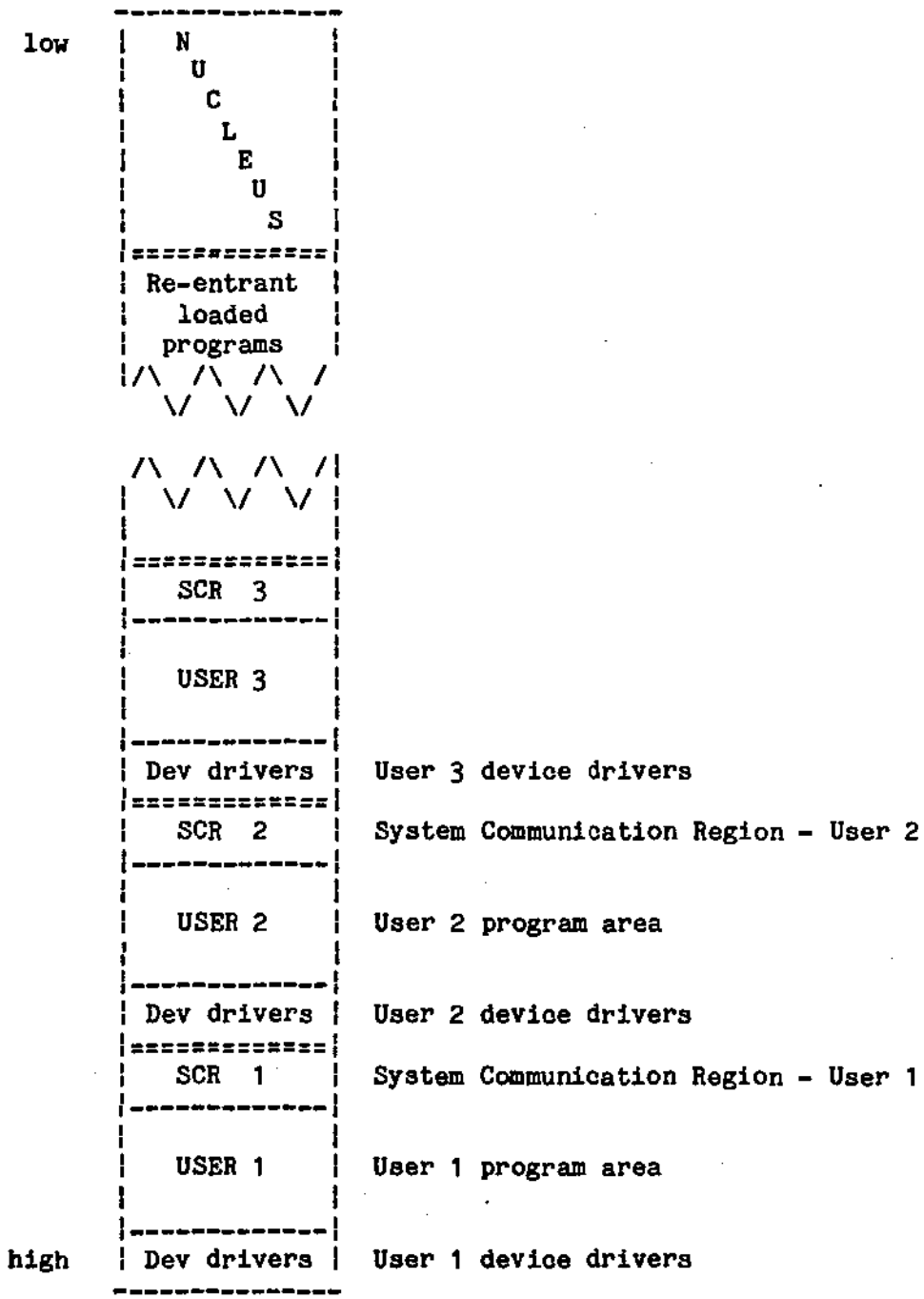
TABLE D-1

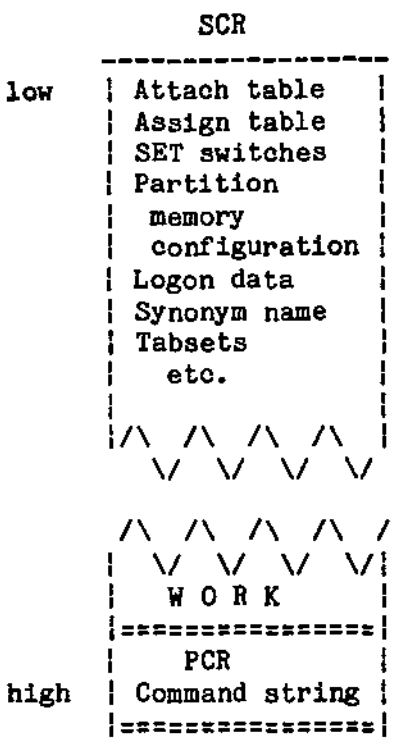
Indexed File Format Filesizes

3	7	11	19	23	31	43	47	59	67	71	79	83
103	107	127	131	139	151	163	167	179	191	199	211	223
227	239	251	263	271	283	307	311	331	347	359	367	379
383	419	431	439	443	463	467	479	487	491	499	503	523
547	563	571	587	599	607	619	631	643	647	659	683	691
719	727	739	743	751	787	811	823	827	839	859	863	883
887	907	911	919	947	967	971	983	991	1019	1031	1039	1051
1063	1087	1091	1103	1123	1151	1163	1171	1187	1223	1231	1259	1279
1283	1291	1303	1307	1319	1327	1367	1399	1423	1427	1439	1447	1451
1459	1471	1483	1487	1499	1511	1523	1531	1543	1559	1567	1571	1579
1583	1607	1619	1627	1663	1667	1699	1723	1747	1759	1783	1787	1811
1823	1831	1847	1867	1871	1879	1907	1931	1951	1979	1987	1999	2003
2011	2027	2039	2063	2083	2087	2099	2111	2131	2143	2179	2203	2207
2239	2243	2251	2267	2287	2311	2339	2347	2351	2371	2383	2399	2411
2423	2447	2459	2467	2503	2531	2539	2543	2551	2579	2591	2647	2659
2663	2671	2683	2687	2699	2707	2711	2719	2731	2767	2791	2803	2819
2843	2851	2879	2887	2903	2927	2939	2963	2971	2999	3011	3019	3023
3067	3079	3083	3119	3163	3167	3187	3191	3203	3251	3259	3271	3299
3307	3319	3323	3331	3343	3347	3359	3371	3391	3407	3463	3467	3491
3499	3511	3527	3539	3547	3559	3571	3583	3607	3623	3631	3643	3659
3671	3691	3719	3727	3739	3767	3779	3803	3823	3847	3851	3863	3907
3911	3919	3923	3931	3943	3947	3967	4003	4007	4019	4027	4051	4079
4091	4099	4111	4127	4139	4159	4211	4219	4231	4243	4259	4271	4283
4327	4339	4363	4391	4423	4447	4451	4463	4483	4507	4519	4523	4547
4567	4583	4591	4603	4639	4643	4651	4663	4679	4691	4703	4723	4751
4759	4783	4787	4799	4831	4871	4903	4919	4931	4943	4951	4967	4987
4999	5003	5011	5023	5039	5051	5059	5087	5099	5107	5119	5147	5167
5171	5179	5227	5231	5279	5303	5323	5347	5351	5387	5399	5407	5419
5431	5443	5471	5479	5483	5503	5507	5519	5527	5531	5563	5591	5623
5639	5647	5651	5659	5683	5711	5743	5779	5783	5791	5807	5827	5839
5843	5851	5867	5879	5903	5923	5927	5939	5987	6007	6011	6043	6047
6067	6079	6091	6131	6143	6151	6163	6199	6203	6211	6247	6263	6271
6287	6299	6311	6323	6343	6359	6367	6379	6427	6451	6491	6547	6551
6563	6571	6599	6607	6619	6659	6679	6691	6703	6719	6763	6779	6791
6803	6823	6827	6863	6871	6883	6899	6907	6911	6947	6959	6967	6971
6983	6991	7019	7027	7039	7043	7079	7103	7127	7151	7159	7187	7207
7211	7219	7243	7247	7283	7307	7331	7351	7411	7451	7459	7487	7499
7507	7523	7547	7559	7583	7591	7603	7607	7639	7643	7687	7691	7699
7703	7723	7727	7759	7823	7867	7879	7883	7907	7919	7927	7951	7963
8011	8039	8059	8087	8111	8123	8147	8167	8171	8179	8191	8219	8231
8243	8263	8287	8291	8311	8363	8387	8419	8423	8431	8443	8447	8467
8527	8539	8543	8563	8599	8623	8627	8647	8663	8699	8707	8719	8731
8747	8779	8783	8803	8807	8819	8831	8839	8863	8867	8887	8923	8951
8963	8971	8999	9007	9011	9043	9059	9067	9091	9103	9127	9151	9187
9199	9203	9227	9239	9283	9311	9319	9323	9343	9371	9391	9403	9419
9431	9439	9463	9467	9479	9491	9511	9539	9547	9551	9587	9619	9623

APPENDIX D

MEMORY/DISK LAYOUT





OASIS Disk Layout

Sector	Contents
0	Boot loader.
1	Disk label and allocation bit map.
2 thru m	Any additional sectors needed for bit map.
m+1 thru n	Directory entries (8 entries per sector).
n+1 thru o	1K blocks for files and programs.

m = ((o/4)-1792)/2048
n = directory size * 8 - m
o = surfaces * (tracks/surface) * (sectors/track)

The following is the definition of the contents of a file entry in the directory. Each entry is 32 decimal bytes in length. The locations and contents are expressed in hexadecimal.

Directory Entries

Byte	Contents
0	File format: 00 = Empty - never used x1 = Relocatable x2 = Absolute x4 = Sequential x8 = Direct 10 = Indexed FF = Deleted
01-08	File name.
09-10	File type.
11-12	Record count.
13-14	Block count.
15-16	Address of 1st sector.
17-18	Variable by file format: I = Key length - 1 and record length - 1 S = Record length of longest record in file D = Allocated record length A,R = Record length (always = sector length)
19-1B	Date and time of update.
1C	Owner Id.
1D	Shared from owner Id.
1E-1F	Variable by file format: I = Allocated file size S = Disk address of last sector in file D,R = Zero A = Origin address

APPENDIX E

SYSTEM FILES

E.1 The System Disk

The OASIS system requires a system disk to operate. The system disk is always the disk currently attached as the "S" disk and normally is the disk the system was booted from.

Frequently the system disk contains all of the commands to be used by the operator. However, this is not a requirement. In order for a disk to qualify as a system disk it must contain, as a minimum, the following files:

SYSTEM.NUCLEUS	
SYSTEM.CSI	
SYSTEM.EXECSAVE	
SYSTEM.EXEC1	
SYSTEM.DEVNAMES	
SYSTEM.ERRMSG	
SYSTEM.DEVnn	Device drivers
SYSTEM.CLASSnn	Terminal class codes
SYSTEM.ACCOUNT	If accounting is implemented
SYSTEM.HISTORY	If accounting is implemented
SYSTEM.SPOOLER	If print spooler is implemented
SYSTEM.SPOOL$$n	If print spooler is implemented
SYSTEM.MAILBOX	If messages are to be used
SYSTEM.SCHEDULE	If program scheduling is to be used

As can be seen, the required files all have the file name SYSTEM. For the best response time of the system these files should be placed on the disk near the beginning. (Some systems require that the SYSTEM.NUCLEUS be immediately follow the directory.) To accomplish this is relatively simple: onto a cleared disk (INITDISK command) copy the SYSTEM.NUCLEUS, then copy all other SYSTEM files (COPYFILE SYSTEM * S = = <fd>).

Because the SYSTEM.ACCOUNT file resides on the system disk, changing system disks may cause a different set of account names to be available. For this reason it is generally best to have only one system disk with different data disks for the various applications of the computer. Sometimes this is not possible. In this situation, changes to the SYSTEM.ACCOUNT file on one system disk should be made to all system disks. If this is not done, operators may become frustrated when they cannot LOGON to their account, thinking that somebody has restricted their access to the system when in fact somebody merely used a different system disk.

NOTE: The system disk received from your OASIS distributor should be kept "virgin": make no changes to account names, privilege levels, etc., without considering the effect this might have on the access and support of the operating system. The user should always be able to return to the version of the system received to determine if a problem is in the original system or in something he may have done to the system.

Multiple System Disks

When it is necessary to have multiple system disks you should make the SYSTEM.ACCOUNT file be identical on all system disks. This is easily done by using the COPY option of the ACCOUNT command.

The importance of maintaining identical SYSTEM.ACCOUNT files on the various system disks cannot be overemphasized. If account ids are different on a system disk then the accounts will have a different set of files. It is possible that a set of files will be unaccessible (and their existence will not be reported) if accounts are irresponsibly deleted or changed.

B.2 The SYSTEM.DEVNAMES File

The SYSTEM.DEVNAMES file contains the acceptable names of all physical devices and their primary modes of operation. This is a sequential file and is maintainable by the user. In fact, additional names can be added to this file to create more meaningful synonyms to existing devices. If a device name is not in this file you will not be able to ATTACH a logical name to the device (except by using the physical device number, i.e. ATTACH A 4).

The format of the file is simple and straightforward: one record per device name, each record containing the device name, device number, and mode of operation, all separated by a single space.

The device name is any alphanumeric string of characters, up to eight characters in length, with the same limitations as file names (no embedded spaces, must start with a letter, etc.). This name will be the normal method of specifying the device in the ATTACH command.

The device number is the number of the device driver program to be used to service the device. When a device is ATTACHed this number is used to find the device driver program on the system disk. All device driver programs have file descriptions in the form: SYSTEM.DEVnn where nn is the number of the device driver. This number must be in the range 1 through 32 and the numbers 1 through 8 are reserved for disk device drivers. Care should be taken to ensure that two or more device drivers are not attached that service the same peripheral. This would cause unpredictable results.

The mode of operation may be any combination of the following:

D - Indicates the device is a disk device.
C - Indicates that the device may be a console.
P - Indicates that the device is a printer.
S - Indicates that the device uses serial I/O.
I - Indicates that the device supports input.
O - Indicates that the device supports output.

These codes tell the ATTACH command what options may apply to the device. For example a device that uses serial I/O uses the options: PO, PE, B, LF, FF, ENAB; printer devices use the options: L, P, O, PP.

It is permissible for more than one device name to have the same device number. These additional device names will be treated as synonyms to the first device name that uses the number.

Disk device name records in this file contain additional information: the disk devices that share the disk driver. Normally a disk controller and the disk driver program control and communicate with up to four disk drives. To indicate to the ATTACH program that only one disk driver need be loaded for all of the disks that share this driver the ATTACH program needs to know exactly which disks are shared by the driver.

To illustrate this file part of the standard, distributed file will be used. The complete DEVNAMES file varies from system to system.

```
FLOPPY1 1 D 1 2 3 4
FLOPPY2 2 D 1 2 3 4
FLOPPY3 3 D 1 2 3 4
FLOPPY4 4 D 1 2 3 4
PIO1 13 PIO
LP 13 PIO
SIO1 17 CPSIO
SIO 17 CPSIO
DISK1 1 D 1 2 3 4
DISK2 2 D 1 2 3 4
DISK3 3 D 1 2 3 4
DISK4 4 D 1 2 3 4
```

As can be seen there are four disk devices: FLOPPY1, FLOPPY2, FLOPPY3, and FLOPPY4. These four devices have synonyms of DISK1, DISK2, DISK3, and DISK4 respectively. (Only the FLOPPYn names will be used when the current attachments are displayed as they occur first in the file.) Additionally, these four disk devices all share one disk driver.

There is a serial device named SIO1 with a synonym of SIO. This serial device supports input and output and may be attached as the console.

There is also a parallel device named PIO with a synonym of LP. This parallel device supports input and output.

E.3 The SYSTEM.DEVnn Files

The files are the physical device driver programs and should be kept unless disk space is at a premium on the system disk and one or more device drivers are never used.

E.4 The SYSTEM.CLASSnn Files

These files are the terminal class code translation tables used by the CONOUT device to translate cursor control commands into the proper codes required by your specific terminal. It is only necessary to keep the specific class code files that you use--the others may be deleted to make more space available on your system disk.

For more information about these files refer to the appendix on "Terminal Class Codes" in this manual.

E.5 The SYSTEM.ERRMSG File

The SYSTEM.ERRMSG file contains all of the error, warning, and information messages that the system uses. This file is a direct file and cannot be maintained by the editor. However, a program is available to maintain this file if it is necessary to translate the messages to a foreign language.

The format of this file is simple: one record per message, each record containing the message number and the message text, separated by one space.

The message number is a number in the range of 1 through 255 used by the system to

reference the messages.

The message text contains the message itself. Some messages contain variable information determined only at display time. This variable information is denoted by inclosing the parameter number within braces such as {1}.

E.6 The SYSTEM.ACCOUNT File

The SYSTEM.ACCOUNT file is a restricted access file required by the system if user accounting is to be used. Once this file has been removed (only by using the ACCOUNT command) from a disk there can be no user accounting while that disk is the system disk.

This file contains information about the user accounts: the account names, passwords, and privilege levels for each account in the system. This file can only be accessed by certain system programs and can only be maintained by the ACCOUNT command when the system account is logged on.

E.7 The ACCOUNT.HISTORY File

The ACCOUNT.HISTORY file contains the user accounting information that was transferred from the SYSTEM.HISTORY file (see ACCOUNT command). This file is a sequential file accessible by the user for logon analysis, etc. A record is added to the SYSTEM.HISTORY file (and later transferred to the ACCOUNT.HISTORY file) every time a user logs on, logs off, or boots the system.

Each record of this file contains information dependent upon what caused it to be created. However, every record begins with the same format: time of creation, date of creation, and record type. The record type information specifies what caused the record to be created and what information follows:

1	System boot - no further data
2	User LOGON - followed by account name, id, and port.
3	User LOGOFF - followed by account name, id, port, and elapsed logon time in minutes.

All fields of information are separated from other fields by spaces. The date field is in standard date format: mm/dd/yy. The time field is in standard time format: hh:mm:ss. Account names always use eight positions with trailing spaces, if necessary. The elapsed time may have leading spaces (no leading zeros) and is expressed to the nearest hundredth of a minute.

Because the ACCOUNT.HISTORY file is always replaced (not appended to) when the SYSTEM.HISTORY information is transferred you should probably append the file to one of your own if you wish to maintain any archive information.

Note 1: If your system does not support the time of day feature the time of creation and the elapsed time fields will contain zeros. By placing zeros in the record the file will have the same format as a file created on a system that does have the time of day feature and can be analyzed by a program that is identical on both systems.

Note 2: The history record creation can be turned on or off with the SET command.

E.8 User Defined Synonyms File

The user defined synonym file, referred to in the SET and SHOW commands, is a sequential file(s) created and maintained by the user containing the user defined synonyms to commands. There may be more than one user defined synonym file in the system but the SET command only allows one of the files to be designated as the user synonym file at any one time.

This file may have any valid name but must reside on the system disk and must have a file type of SYNONYM. This option of the system is not normally used but is quite convenient in some situations. The most frequent use of this option occurs when a user has access to, and utilizes, several different operating systems. In order to avoid confusion this user would probably create a synonym file that reflects the command names used by the other system(s). (The other system probably doesn't allow the user to define his own synonyms.) For example, some operating systems use the name DIRECTRY, DIR or VTOC for the function performed by OASIS's FILELIST command. In this case the user could create a synonym for the FILELIST command of DIRECTRY with an acceptable abbreviation of three characters (DIR). Other commands could be treated similarly.

The format of the user defined synonyms file is relatively simple: one record per synonym definition, each record containing three fields specifying the true command name, the synonym to the command name, and the minimum number of characters allowed for abbreviation of the synonym, each separated by a single space from the other fields. The minimum number of characters allowed for abbreviation is optional and, when omitted, implies that no abbreviation is allowed.

In the above situation the synonym for the FILELIST command would look like:

```
FILELIST DIRECTRY 3
FILELIST VTOC
```

E.9 The SYSTEM.EXECSAVE File

The SYSTEM.EXECSAVE file is a data file used by the EXECutive language processor to save the status and variables of one or more EXEC programs while another EXEC program or command is executing.

In multi-user OASIS the EXECSAVE file is named EXEC1, EXEC2, etc., one for each user partition.

This file is a direct access file with each record 512 characters in length. The number of records in this file determine the level of EXEC call nesting allowed by the system. As distributed this file contains two records which allow one EXEC program to call another but does not allow that second program to call another EXEC program.

To expand (or decrease) the level of EXEC nesting log on to the system account, erase the SYSTEM.EXECSAVE file (be conscious of the fact that while this file is removed no EXECs may be executed and the system may not be booted), and create a new file with the filesize corresponding to the level of nesting desired.

E.10 The SYSTEM.SPOOL$$1 File

The SYSTEM.SPOOL$$1 file is a data file used by the printer spooler and must be created by the user before the spooler is invoked (see chapter: "SPOOLER COMMAND").

This file may reside on any attached disk. Any file format is acceptable but a direct file is the easiest for you to create. The record length of the file is ignored by the spooler, only the total allocated file size is used.

A usable file size for the SYSTEM.SPOOL$$1 file is about 100K bytes. This size file will be allow the system to keep about 25 pages ahead of the programs generating output to the printer (a full page of printing is about 70 characters x 60 lines = 4200 characters).

This size of a file can be created by entering the command:

```
>CREATE SYSTEM SPOOL$$1 fd (DIRECT REC 512 FILE 200)
```

APPENDIX F

ANSI FORMS CONTROL

The OASIS Operating System utilizes the American National Standards Institute (ANSI) standard for printer forms control for output directly to a device attached as a printer.

When ANSI forms control is used (optional in the LIST command) the first character of each record transmitted to a printer is a forms control command. This command character is not printed on the printer but is translated to the codes required to effect the forms control desired. A record is always terminated with a carriage return (in BASIC this is accomplished by using no trailing semicolon in the output list).

Every record output to a printer is printed with a trailing carriage return but no trailing line feed. The line advance is performed by the next record's forms control, if desired.

The following is a list of the control characters and their functions.

Char	Function
1	Forms eject.
+	No forms advance. This allows for overprinting.
0	Double line advance. Creates one blank line.
-	Triple line advance. Creates two blank lines.
other	Single line advance. This is the normal mode of printing and is equivalent to printing without ANSI forms control except the first character is not printed.

APPENDIX G

TERMINAL CLASS CODES

The terminal class code, set with the ATTACH command, informs the operating system what display features a console has and how to invoke them. Unfortunately terminal manufacturers have no standards imposed on them in this area. Therefore each manufacturer uses the method that he thinks is best or is easiest for him to implement.

In most systems, programs that try to utilize some of the special features that terminals have must be coded for a specific terminal(s). This is not necessary when using the OASIS system. In OASIS programs are coded using an internal set of codes defined below. When outputing to the console OASIS translates these codes to the codes required by the attached console. This provides greater flexibility and transportability for the programs.

The tables on the following pages describe the classes implemented and distributed with the OASIS operating system. Each class code is defined in its respective SYSTEM.CLASSnn file. Additional classes may be added by you--refer to the end of this appendix.

Internal Terminal Codes

Function	Code	Function	Code
Cursor HOME	SOH	Keyboard ON	EM
CLEAR Screen	FF	Keyboard OFF	NAK
End Of Screen clear	CAN	Protect ON	EOT
End Of Line clear	ETB	Protect OFF	ENQ
Cursor LEFT	BS	Format ON	STX
Cursor RIGHT	ACK	Format OFF	ETX
Cursor UP	SUB	Blink ON	GS
Cursor DOWN	LF	Blink OFF	RS
Cursor address	DLE,c,l	Erase Unprotected	FS
Address coding	Abs	Insert Line	DC1
		Insert Character	DC3
		Delete Line	DC2
		Delete Charcter	DC4

Terminal Class Codes 1 - 4

FILE Func	CLASS1 BEEHIVE 100	CLASS2 ADDS 580	CLASS3 ADM 3A	CLASS4 SOROC IQ
HOME	ESC,"H"	DLE,b,VT,b	RS	RS
CLEAR	ESC,"E"	FF	SUB	ESC,"*"
EOS	ESC,"J"	Simulated	Simulated	ESC,"Y"
EOL	ESC,"K"	Simulated	Simulated	ESC,"T"
LEFT	BS	NAK	BS	BS
RIGHT	ESC,"C"	ACK	FF	FF
UP	ESC,"A"	SUB	VT	VT
DOWN	LF	LF	LF	LF
X,Y Coding	ESC,"F",l,c Standard	DLE,l,VT,c Standard	ESC,"=",l,c Standard	ESC,"=",l,c Standard
IL	N	N	N	N/A
IC	O	O	O	N/A
DL	T	T	T	N/A
DC				N/A
EU				ESC,"+"
KON *	I	I	I	ESC,"'"
KOFF *	M	M	M	ESC,"#"
PON	P	P	P	ESC,")"
POFF	L	L	L	ESC,"("
FON	E	E	E	ESC,"&"
FOFF	M	M	M	ESC,"'"
BON	E	E	E	N/A
BOFF	N	N	N	N/A
	T	T	T	
	E	E	E	
	D	D	D	

The standard cursor control is the ASCII displayable characters. The space character has a value of 1, the exclamation character has a value of 2, etc.

N/A indicates that the function is not available on the terminal.

* The OASIS system maintaines its own keyboard on/off independant of the terminal itself. When the KOFF feature is enabled any characters typed on the keyboard are ignored by the system and the BELL code is sent to the CONOUT device.

Terminal Class Codes 5 - 8

FILE Func	CLASS5 V D M	CLASS6 HAZELTINE 1500	CLASS7	CLASS8 Perkin-Elmer Fox
HOME	SOH	"~",DC2		ESC,"H"
CLEAR	FF	"~",FS		ESC,"K"
EOS	CAN	"~",CAN		ESC,"J"
EOL	ETB	"~",SI		ESC,"I"
LEFT	BS	BS		BS
RIGHT	ACK	DLE		ESC,"C"
UP	SUB	"~",FF		ESC,"A"
DOWN	LF	LF		LF
X,Y Coding	DLE,c,l Abs	"~",DC1,c,l Standard		ESC,"X",l,c Standard
IL	N	"~",SUB		N
IC	O	N/A		O
DL	T	"~",DC3		T
DC		N/A		
EU		"~",GS		
KON *	I	"~",ACK		I
KOFF *	M	"~",NAK		M
PON	P	"~",EM		P
POFF	L	"~",US		L
FON	E	N/A		E
FOFF	M	N/A		M
BON	E	N/A		E
BOFF	N	N/A		N
	T			T
	E			E
	D			D

Abs indicates that the line and column number is column number is coded using absolute values for the line and column number, base 0. (Column 1 is 0, column 2 is 1, etc.)

Standard indicates that the line and column number is coded using the ASCII displayable character set: the space character is 1, the exclamation character is 2, etc.

N/A indicates that the function is not available on the terminal or that it is not implemented in OASIS.

* The OASIS system maintaines its own keyboard on/off independant of the terminal itself. When the KOFF feature is enabled any characters typed on the keyboard are ignored by the system and the BELL code is sent to the CONOUT device.

Terminal Class Codes 9 - 12

FILE	CLASS9	CLASS10	CLASS11	CLASS12
	Perkin-Elmer	A N S I	Infoton 100	
Func	Bantom	(ANSI)		
HOME	ESC,"H"	ESC,"[H"	ESC,"H"	
CLEAR	ESC,"K"	ESC,"[H"	FF	
		ESC,"[2J"		
EOS	Simulated	ESC,"[J"	ESC,"J"	
EOL	Simulated	ESC,"[K"	ESC,"K"	
LEFT	BS	BS	BS	
RIGHT	ESC,"C"	ESC,"[C"	ESC,"C"	
UP	ESC,"A"	ESC,"[A"	ESC,"A"	
DOWN	LF	LF	LF	
X,Y	ESC,"X",l,c	ESC,"[",l, ";",c,"H"	ESC,"f",c,l	
Coding	Standard	ASCII	Standard	
IL	N	ESC,"[1L"	ESC,"L"	
IC	O	N/I	N/A	
DL	T	ESC,"[1M"	ESC,"M"	
DC		ESC,"[1P"	N/A	
EU		ESC,"[H", ESC,"[2J"		
KON *	I			
KOFF *	M			
PON	P	N/I	N/A	
POFF	L	N/I	N/A	
FON	E	N/I	ESC,"4"	
FOFF	M	N/I	ESC,"3"	
BON	E	ESC,"[m"	N/A	
BOFF	N	ESC,"[7m"	N/A	
	T			
	E			
	D			

ASCII indicates ASCII characters are used: column 1 is "1", column 22 is "22", etc.

Standard indicates that the line and column number is coded using the ASCII displayable character set: the space character is 1, the exclamation character is 2, etc.

N/A indicates that the function is not available on the terminal or that it is not implemented in OASIS. N/I indicates that the function is not implemented in the conversion file.

* The OASIS system maintaines its own keyboard on/off independant of the terminal itself. When the KOFF feature is enabled any characters typed on the keyboard are ignored by the system and the BELL code is sent to the CONOUT device.

G.1 Implementing New Class Codes

If you have a terminal that you wish to use as a console but its cursor control requirements is not meet by any of the class codes defined and distributed then you may implement your own class code by using the OASIS MACRO assembler.

For details on the requirements of the SYSTEM.CLASSnn file refer to the OASIS MACRO Assembler Language Reference Manual.

APPENDIX H

PRIVILEGE LEVELS

The OASIS operating system supports privileged access to programs. Privileged access means that each program is assigned a privilege value. Access to a specific program is allowed only if the current user has a privilege level equal to or greater than the privilege value of the program.

An attempt to access a program with a privilege value higher than your privilege level will be rejected with no error message.

The current privilege level may be lowered by using the SET PRIV command. The current privilege level may be displayed by using the SHOW PRIV command. The current privilege level is a SYSGENable function but this only has meaning when the user accounting feature is removed.

The following table lists the OASIS commands and the privilege value associated with them, as they are distributed.

5 ACCOUNT	0 FILELIST	3 MEMTEST	3 SECTOR
0 ASSIGN	4 FILT8080	0 MOUNT	1 SEND
0 ATTACH	5 FORCE	0 MSG	1 SET
3 BACKUP	2 GETFILE	3 OWNERCHG	3 SHARE
2 BASCONV	3 INITDISK	3 PATCH	0 SHOW
2 BASIC	4 INTELHEX	0 PEEK	0 SPOOLER
3 CHANGE	3 KILL	1 RECEIVE	5 START
1 COPYFILE	2 LINK	3 RECOVER	0 STATE
1 CREATE	0 LIST	4 RELOCATE	5 STOP
4 DEBUG	5 LOAD	1 RENAME	5 SYSGEN
2 DUMPDISK	0 LOGOFF	3 REPAIR	2 TEXTEDIT
2 EDIT	0 LOGON	0 RUN	1 TERMINAL
1 ERASE	4 MACRO	0 SCHEDULE	5 UNLOAD
2 FILECONV	0 MAILBOX	0 SCRIPT	3 VERIFY

The privilege level of a program (COMMAND) may be changed to a different value by using the CHANGE command.

Note 1: Because the privilege level of the system cannot be raised by any program you should take care to always maintain at least one system disk that has the highest privilege level that you will ever need. This is usually accomplished by never making any changes to the copy of the system on the distibution disk.

Note 2: Do not raise the privilege level of the LOGON and LOGOFF commands--you would make it very difficult to change from one account to another. Privilege values and levels may be in the range of 0 to 5. The following conventions were used to determine the privilege values of the system programs:

0 Non-file modifying system utilities, RUN version of BASIC.
1 File modifying system utilities.
2 High level language programming utilities.
3 Maintenance utilities (i.e. PATCH).
4 Assembly level language programming utilities.
5 System management.

APPENDIX I

SYSTEM ERRORS AND MESSAGES

I.1 Disk Errors

All disk errors are displayed on the console after the disk retry count has been exhausted. The format of the disk error message is:

Disk <fd>(<#>) <message> <cyl>,<head>,<sect>

Where:

<fd> Indicates the logical disk label (S, A, B, etc).

<#> Indicates the physical disk number that was attached to the logical disk label.

<message> Indicates the specific disk error message.

<cyl> Indicates the cylinder number of the disk that the error occured on. This number is in decimal.

<head> Indicates the head or surface number of the cylinder that the error occured on. This number is in decimal.

<sect> Indicates the physical sector number within the cylinder of the disk error. This number is displayed in decimal.

When a disk error occurs the system automatically retries the operation until it is successful or until the retry count is exhaused. If it is successful no indication of an error is given to the operator (except a slight delay in processing). When the operation is not successful and the retry count is exhausted an error message is displayed on the console and the system waits for a response from the operator.

When a disk error message is displayed on the console the operator must type a key for processing to continue. Valid responses by the operator include:

R Retry the operation again

I Ignore the operation and continue processing as if the operation was successful. This response may cause more errors than intended. In general, only use the I response for read type operations and when you are sure that the information will not be used for a write operation following.

Q Quit the operation and return control to the Command String Interpreter.

M Perform a MOUNT for the error disk. This response is only valid when the error message is 'Disk Changed'.

Caution should be exercised in the use of the I and Q responses. The actual results of this response will be dependent upon the specific operation being performed. However, the I response is invaluable when performing a disk backup or similar operation. Before using the I response you should try the R response several times to insure that the data is truly unrecoverable.

The integrity of information written to or read from a disk is verified by use of a checksum or CRC value. CRC is the initials of the term "Cyclic Redundancy Check" and refers to the process of generating a value using an algorithm that relates the value generated to the information that it was generated from. This is a reasonably accurate and efficient method of verifying read/write integrity. When information is to be written to the disk a CRC value is generated and written along with the data. When the information is read back from the disk a CRC value is generated for the information read. If the CRC of the data read is not the same as the CRC read back with the data then there is an error. This error may be related to the data or it may be that the CRC value read from the disk is in error.

The following disk error messages are described in alphabetic order. The number preceding the error message is the internal error number.

Disk Error Messages

9 Address CRC Error - The sector or track address read from the disk does not correspond to its checksum read from the disk. Enter a carriage return to retry the operation.

4 Data CRC Error - The data read from the disk does not correspond to its checksum read from the disk. Enter a carriage return to retry the operation.

6 Disk Changed - The disk ID on the disk does not match the ID saved in memory. Mount the correct disk, unload this disk, or enter an M.

5 Invalid Parameters - The sector number to be read/written is invalid (negative or greater than maximum). The data and/or program is invalid. This error usually only occurs during a read of a sequential file, indicating that the link address to the next block of the file is invalid. The only method of exiting from this error condition is the I or Q response.

3 Not Initialized - Addresses and data on the disk cannot be read; system assumes the disk has not been formatted. To recover from this error you must quit and initialize the disk. If the error is occuring on many disks that are actually initialized then the head alignment of the disk drive should be suspected.

1 Not Ready - Disk not loaded, not rotating at proper speed, or is mounted off center. Unload and reload disk slowly.

7 Sector Not Found - Sector address is invalid or doesn't match its checksum. If the operation is a write to the disk then enter a carriage return to retry the operation - usually this will recover from the error. If the operation is a read, then you will probably have to cancel the operation and either restore the data from the backup disk (initialize this disk first) or try to recover the data using the DEBUG program (if available). The DEBUG program, if used improperly, may cause more loss of data than recovered!

8 Track Not Found - Track address is invalid or doesn't match its checksum. Retry the operation. If not recoverable restore from your backup disk.

2 Write Protected - Disk is physically or logically write protected.

The following errors may occur during system start up (boot):

Disk Error: nn Indicates one of the above errors. The disk error message cannot be displayed because the system cannot be read into memory. Usually the error can be resolved by reloading the disk slowly, making sure that it is seated properly, etc.

Not Found: "SYSTEM.NUCLEUS" Indicates that a non-system disk is in the system drive.

Not Found: "SYSTEM.EXECSAVE" Indicates that the EXECSAVE file is not on the system disk. This file must exist in order to complete the system start up.

Not Found: "SYSTEM.CSI" Indicates that a non-system disk is in the system drive.

Not Found: "SYSTEM.ERRMSG" Indicates that a non-system disk is in the system drive.

I.2 System Messages

The following is a listing of the SYSTEM.ERRMSG file as distributed. The messages that contain numbers surrounded by braces {} indicate that the message contains variable information that is determined only at the time the message is displayed. For example, message number 42 is used by the STATE command to display the number of files found and the number of disk blocks used by those files.

The messages are fairly self explanatory. More detail about the specific messages cannot be given because the messages are almost totally independent of the programs. In general, if a message is displayed that you do not know how to interpret then read the section or manual describing the program that was executing when the message was displayed.

Number	Text
[1]	Syntax error.
[2]	File name missing.
[3]	File type missing.
[4]	Drive code missing.
[5]	Disk not attached.
[6]	Disk not mounted.
[7]	Invalid option.
[8]	File not found.
[9]	Disk full.
[10]	Directory full.
[11]	Invalid file name.
[12]	Invalid drive code.
[13]	File "{1}.{2}" not found.
[14]	"{1}" is not recognizable.
[15]	"{1}" is an invalid numeric.
[16]	Expecting "{1}", found "{2}".
[17]	Expecting "{1}", found end-of-line.
[18]	Protected file.

[19] Insufficient privilege.
[20] Incorrect serial number.
[21] Expecting end-of-line, found "{1}".
[22] Required paramter missing.
[23] Must be "ON" or "OFF".
[24] Incorrect format, should be "{1}".
[25] Cannot detach the CONSOLE.
[26] Invalid record in "{1}.{2}".
[27] Cannot attach to an output only device.
[28] Cannot attach to an input only device.
[29] Can only attach to a random storage device.
[30] Cannot attach to a random storage device.
[31] Cannot detach the System disk.
[32] Undefined logical device.
[33] Undefined physical device.
[34] Must be sequential organization.
[35] File already exists.
[36] Organization code is missing.
[37] Missing option: "{1}".
[38] COMM device is not attached.
[39] Device is not attached.
[40] File channel is not assigned.
[41] No files found.
[42] {1} file(s), {2} block(s).
[43] Must have at least two disks attached.
[44] Source and destination drives cannot be the same.
[45] No files erased.
[46] {1} file(s) erased, {2} block(s) recovered.
[47] Source and destination do not have the same capacity.
[48] Receiver disconnected.
[49] Receiver timed out.
[50] Too many serial numbers.
[51] Load module contains unresolved external references.
[52] Cannot mix ABS with REL or COM modules.
[53] Cannot have multiple ABS modules.
[54] Symbol table overflow.
[55] Unexpected end-of-file.
[56] Program has no PABs!
[57] Only private files may be shared.
[58] SYSTEM.ACCOUNT file is missing.
[59] System account cannot share files.
[60] Incorrect password.
[61] Account not found.
[62] File "{1}.{2}:{3}" already exists on account "{4}".
[63] Spooler is not initialized.
[64] Spooler is busy printing.
[65] Spooler is waiting for work.
[66] Spooler is waiting for printer.
[67] Spooler is suspended.
[68] Insufficient Memory.
[69] Device is attached to partition {1}.
[70] Invalid partition number.
[71] Partition not allocated.
[72] Partition is still active.
[73] Partition is not active.
[74] Can't send message to your self.

[75] User is not logged on.
[76] User is not receiving messages.
[77] Must be single user mode or private disk volume.
[78] Must be single user mode.
[79] Can only execute from System account.
[80] No files changed.
[81] File "{1}.{2}:{3}" is now owned by account "{4}".
[82] Partition size is too small.
[83] Partition size is too large.
[84] Partition must be one larger than current top partition.
[85] Can only modify top partition.
[86] Next lower parrtition is still active.
[87] Module {1} is not loaded.
[88] Module {1} is already loaded.
[89] Too many re-entrant modules.

APPENDIX J

ASCII CHARACTER SET

Dec	Hex	Oct	Binary	Display	Key	Definition	Special Use
0	00	000	0000 0000		CTRL/@	NUL	Filler
1	01	001	0000 0001	Home	CTRL/A	SOH	Start of Heading
2	02	002	0000 0010	FON	CTRL/B	STX	Start of Text
3	03	003	0000 0011	FOFF	CTRL/C	ETX	End of Text
4	04	004	0000 0100	PON	CTRL/D	EOT	End Of Transmission
5	05	005	0000 0101	POFF	CTRL/E	ENQ	Enquiry
6	06	006	0000 0110	Right	CTRL/F	ACK	Acknowledge
7	07	007	0000 0111	Bell	CTRL/G	BEL	Bell
8	08	010	0000 1000	BS	CTRL/H	BS	Backspace
9	09	011	0000 1001	TAB	CTRL/I	HT	Horizontal tabulation
10	0A	012	0000 1010	LF	CTRL/J	LF	Line Feed
11	0B	013	0000 1011		CTRL/K	VT	Vertical tabulation
12	0C	014	0000 1100	CLEAR	CTRL/L	FF	Form Feed
13	0D	015	0000 1101	CR	CTRL/M	CR	Carriage Return
14	0E	016	0000 1110		CTRL/N	SO	Shift Out
15	0F	017	0000 1111		CTRL/O	SI	Shift In
16	10	020	0001 0000		CTRL/P	DLE	Data Link Escape Cursor addre
17	11	021	0001 0001		CTRL/Q	DC1	Device Control 1
18	12	022	0001 0010		CTRL/R	DC2	Device Control 2
19	13	023	0001 0011		CTRL/S	DC3	Device Control 3
20	14	024	0001 0100		CTRL/T	DC4	Device Control 4
21	15	025	0001 0101	KOFF	CTRL/U	NAK	Negative Acknowledge
22	16	026	0001 0110		CTRL/V	SYN	Synchronous Idle
23	17	027	0001 0111	EOL	CTRL/W	ETB	End of Transmission Block
24	18	030	0001 1000	EOS	CTRL/X	CAN	Cancel
25	19	031	0001 1001	KON	CTRL/Y	EM	End of Medium
26	1A	032	0001 1010	Up	CTRL/Z	SUB	Substitute
27	1B	033	0001 1011		CTRL/[	ESC	Escape
28	1C	034	0001 1100		CTRL/\	FS	File separator
29	1D	035	0001 1101	BON	CTRL/]	GS	Group separator
30	1E	036	0001 1110	BOFF	CTRL/^	RS	Record separator
31	1F	037	0001 1111		CTRL/_	US	Unit separator

Dec	Hex	Oct	Binary	Display	Key	Definition	Special Use
32	20	040	0010 0000			space bar SP - Space	
33	21	041	0010 0001	!	!	Exclamation mark	
34	22	042	0010 0010	"	"	Double quote	
35	23	043	0010 0011	#	#	Number symbol	LINK prompt
36	24	044	0010 0100	$	$	Dollar symbol	
37	25	045	0010 0101	%	%	Percent symbol	
38	26	046	0010 0110	&	&	And - Ampersand	
39	27	047	0010 0111	'	'	Single quote	
40	28	050	0010 1000	(	(	Left parenthesis	
41	29	051	0010 1001	)	)	Right parenthesis	COPYFILE prompt
42	2A	052	0010 1010	*	*	Asterisk	EDIT prompt
43	2B	053	0010 1011	+	+	Plus symbol	
44	2C	054	0010 1100	,	,	Comma	
45	2D	055	0010 1101	-	-	Minus/Hyphen symbol	BASIC prompt
46	2E	056	0010 1110	.	.	Period/Decimal point	BUILD prompt
47	2F	057	0010 1111	/	/	Divide symbol - Slash/Slant	
48	30	060	0011 0000	0	0	Zero	
49	31	061	0011 0001	1	1	One	
50	32	062	0011 0010	2	2	Two	
51	33	063	0011 0011	3	3	Three	
52	34	064	0011 0100	4	4	Four	
53	35	065	0011 0101	5	5	Five	
54	36	066	0011 0110	6	6	Six	
55	37	067	0011 0111	7	7	Seven	
56	38	070	0011 1000	8	8	Eight	
57	39	071	0011 1001	9	9	Nine	
58	3A	072	0011 1010	:	:	Colon	EXEC prompt
59	3B	073	0011 1011	;	;	Semicolon	
60	3C	074	0011 1100	<	<	Less than symbol - Left carrot	
61	3D	075	0011 1101	=	=	Equal symbol	DEBUG prompt
62	3E	076	0011 1110	>	>	Greater than/Right carrot	CSI prompt
63	3F	077	0011 1111	?	?	Question mark	

Dec	Hex	Oct	Binary	Display	Key	Definition	Special Use
64	40	100	0100 0000	@	@	At sign - Atpersand	
65	41	101	0100 0001	A	A		
66	42	102	0100 0010	B	B		
67	43	103	0100 0011	C	C		
68	44	104	0100 0100	D	D		
69	45	105	0100 0101	E	E		
70	46	106	0100 0110	F	F		
71	47	107	0100 0111	G	G		
72	48	110	0100 1000	H	H		
73	49	111	0100 1001	I	I		
74	4A	112	0100 1010	J	J		
75	4B	113	0100 1011	K	K		
76	4C	114	0100 1100	L	L		
77	4D	115	0100 1101	M	M		
78	4E	116	0100 1110	N	N		
79	4F	117	0100 1111	O	O		
80	50	120	0101 0000	P	P		
81	51	121	0101 0001	Q	Q		
82	52	122	0101 0010	R	R		
83	53	123	0101 0011	S	S		
84	54	124	0101 0100	T	T		
85	55	125	0101 0101	U	U		
86	56	126	0101 0110	V	V		
87	57	127	0101 0111	W	W		
88	58	130	0101 1000	X	X		
89	59	131	0101 1001	Y	Y		
90	5A	132	0101 1010	Z	Z		
91	5B	133	0101 1011	[	[	Left bracket	
92	5C	134	0101 1100	\	\	Back slash/slant	PATCH prompt
93	5D	135	0101 1101	]	]	Right bracket	
94	5E	136	0101 1110	^	^	ASCII up arrow - circumflex	
95	5F	137	0101 1111	_	_	Underscore	

Dec	Hex	Oct	Binary	Display	Key	Definition	Special Use
96	60	140	0110 0000	`	`	Back quote	
97	61	141	0110 0001	a	a		
98	62	142	0110 0010	b	b		
99	63	143	0110 0011	c	c		
100	64	144	0110 0100	d	d		
101	65	145	0110 0101	e	e		
102	66	146	0110 0110	f	f		
103	67	147	0110 0111	g	g		
104	68	150	0110 1000	h	h		
105	69	151	0110 1001	i	i		
106	6A	152	0110 1010	j	j		
107	6B	153	0110 1011	k	k		
108	6C	154	0110 1100	l	l		
109	6D	155	0110 1101	m	m		
110	6E	156	0110 1110	n	n		
111	6F	157	0110 1111	o	o		
112	70	160	0111 0000	p	p		
113	71	161	0111 0001	q	q		
114	72	162	0111 0010	r	r		
115	73	163	0111 0011	s	s		
116	74	164	0111 0100	t	t		
117	75	165	0111 0101	u	u		
118	76	166	0111 0110	v	v		
119	77	167	0111 0111	w	w		
120	78	170	0111 1000	x	x		
121	79	171	0111 1001	y	y		
122	7A	172	0111 1010	z	z		
123	7B	173	0111 1011	{	{	Left brace	
124	7C	174	0111 1100	|	|	Vertical bar	
125	7D	175	0111 1101	}	}	Right brace	
126	7E	176	0111 1110	~	~	Tilde	
127	7F	177	0111 1111	DEL	DEL	RUBOUT	

Reader's Comments

Name _____ Date ___/___/___
Organization _____
Street _____
City _____ State ____ Zip _____

Name of manual: _____

Did you find errors in this manual? If so, specify with page number.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

Indicate the type of user/reader that you most nearly represent:

- ___ Assembly language programmer
- ___ Higher-level language programmer (BASIC, FORTRAN, etc.)
- ___ Occasional programmer (experienced)
- ___ User with little programming experience
- ___ Student programmer
- ___ Non-programmer interested in computer concepts and capabilities
- ___ Data entry operator

Mail to: OASIS Documentation
Phase One Systems, Inc.
7700 Edgewater Drive #830
Oakland, CA 94621