CUBIX

A SINGLE USER

DISK OPERATING SYSTEM

FOR THE 6809

SYSTEM DESCRIPTION

AND

USER GUIDE

Revised: 30-Jul-90

Dunfield Development Systems

----------------------------

High quality tools for

Embedded Development

at low prices.

http://www.dunfield.com

Copyright 1983-2005 Dave Dunfield

All rights Reserved

CUBIX User Guide

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

1.1 Getting Started 1

2. THE FILE SYSTEM 2

2.1 BLOCK files 2

2.2 CHARACTER files 2

2.3 File names 2

2.4 Wildcards 3

2.5 File Protections 4

2.6 Load Address 4

3. SERIAL DEVICES 4

4. COMMAND MODE 5

4.1 Return Codes 5

5. THE SYSTEM LIBRARY 5

6. COMMAND FILES 6

7. COMMANDS AND UTILITIES 8

8. APPENDICIES 9

8.1 APPENDIX 1 - Commonly used file types 9

8.2 APPENDIX 2 - Return codes 10

8.3 APPENDIX 3 - Error messages 11

8.4 APPENDIX 4 - Sample command files 12

CUBIX User Guide Page: 1

1. INTRODUCTION

CUBIX (also refered to as DOS) is a disk operating system which

runs on the Motorola 6809 microprocessor. It is intended to be a ROM

resident system, and occupies about 6.5 kilobytes of memory excluding

I/O drivers. A single 8K ROM provides ample space to implement an

entire CUBIX system.

CUBIX provides many facilities, including a file system, a command

line interpreter, a command file processor, and an extensive library

of system services available to applications programs.

CUBIX may be implemented on any computer system that has a 6809

central processing unit (CPU), at least one disk drive, and a console

terminal (or integrated keyboard/video display). For information on

porting CUBIX to a new hardware platform, see the CUBIX SYSTEM

PORTING GUIDE.

1.1 Getting Started

When the system is powered on, it first performs a test of the

CUBIX ROM and the system RAM. These tests will take several

seconds, and may be aborted by pressing the ESCAPE key on the

console terminal.

Following the startup tests, CUBIX outputs its startup message

and copyright notice.

If the startup tests were not aborted, the system will then

look for a file with the FILENAME 'A:[MAIN]STARTUP.\*', and if it

is found, it is processed according to its FILETYPE in the same

manner as are commands executed from the SYSTEM LIBRARY (See

SYSTEM LIBRARY).

Finally, CUBIX will enter its internal command mode processor

and output a '\*' prompt, awaiting further commands.

CUBIX User Guide Page: 2

2. THE FILE SYSTEM

The CUBIX file system consists of a set of routines within the DOS

that allow controlled access to the disk for storage and retrieval of

programs and information. Individual programs or collections of data

on the disk are called files.

A disk file is a collection of data which is contained on the disk

in an organized and accessable structure. The DOS maintains a

directory of all files present on the disk, as well as a map

indicating which disk sectors belong to which files, and which

sectors are available for use.

File access is under the control of the DOS, and has two major

types, BLOCK and CHARACTER.

2.1 BLOCK files

BLOCK files are written to and read from the disk directly from

system memory in 512 byte blocks. Block files are always a

multiple of 512 bytes long, and may contain all data bytes from

$00 to $FF ('$' indicates a hexidecimal number). Data transfer to

and from block files is the fastest form of disk file access.

2.2 CHARACTER files

CHARACTER files are written to and read from the disk a single

character at a time. (Data is actually buffered and transferred

to/from the disk in 512 byte blocks by the DOS). Character files

may be any length (But will always occupy a multiple of 512 bytes

of disk space), and may contain any data bytes except for $FF,

which is used as an END OF FILE marker.

The DOS does not make any actual distinction between BLOCK and

CHARACTER files, allowing block files to be read as characters, and

character files to be read in blocks. Note that if a file saved as a

block file contains the data byte $FF, the remainder of that file

will not be accessable when performing a character read.

2.3 File names

DOS files are always identified and accessed through a unique

FILE SPECIFICATION name. File specification's consist of four

parts, a DRIVE SPECIFIER, a DIRECTORY PREFIX, the actual FILENAME,

and a FILETYPE

The DRIVE SPECIFIER is a single character drive name, which is

in the range of 'A' to 'D', and indicates one of the four disk

drives supported by the operating system. The drive specifier is

an optional part of the file specification, and if supplied, must

be separated from the remainder of the file specification by a

colon (':'). If a drive is not specified, the DOS uses a default.

The default when the system is first brought up is 'A:', and may

be changed by the user at any time with the "SET DEFAULT" command.

CUBIX User Guide Page: 3

The DIRECTORY PREFIX is a string of up to eight characters

included in each files directory entry, which provides a means of

separating the files on the disk into logically related groups,

each group having it's own unique directory prefix. The directory

prefix is an optional part of the file specification, and if

supplied, must be separated from the remainder of the file

specification by placing it within square brackets, directly in

front of the FILENAME. If the directory prefix is not specified,

the DOS uses a default. The default when the system is first

brought up is'[MAIN]', and may be changed by the user at any time

with the "SET DEFAULT" command.

The FILENAME is a string of up to eight characters which

specifies the actual name of the file.

The FILETYPE is a string of up to three characters which

specifies the type of the file. It follows the filename, separated

from it by a dot ('.'). The filetype is used to specify the

purpose of the file. There are several filetypes recognized by the

DOS and its utility programs, and indicate files with particular

meaning to the DOS or utility. For a list of commonly used

filetypes, refer to the appendix on file types.

2.3.1 Sample Filename

A:[PREFIX]FILENAME.TYP

^ ^ ^ ^

| | | +----- FILETYPE

| | +------------ FILENAME

| +-------------------- DIRECTORY PREFIX

+------------------------- DRIVE SPECIFIER

2.4 Wildcards

Some comamnds and utilities allow the use of the WILDCARD

character '\*' in a file specification, which means 'anything' for

example, to display the names of all executable ('.EXE') files

with all directory prefixes on the disk in drive 'B', you could

enter the command:

'DIR B:[\*]\*.EXE'

The WILDCARD may also be used as the LAST part of a section of

the file specification. For example, the command:

'DIR ABC\*.\*'

would display the names of all files (On the current default

drive, and with the current default directory prefix) which have

FILENAMES beginning with 'ABC'.

Note that the WILDCARD character '\*' may NOT be used in the

DRIVE SPECIFIER portion of a file specification.

CUBIX User Guide Page: 4

2.5 File Protections

Each file on the disk has in it's directory entry, a bit field

which indicates which file system operations may be performed on

the file. The names of the bits, and the operations which may be

performed are:

(R)ead - The file may be read or loaded.

(W)rite - The file may be written to or saved.

(E)xecute - For '.EXE' files, it may be executed.

For '.@' files it may be invoked as

a command file.

(D)elete - The file may be deleted from the disk.

When a file is first created, the protection bits are set to

allow all of the above operations. The protection bits may be

altered with the 'SET FILE' command.

2.6 Load Address

Each file on the disk has in it's directory entry a 16 bit

address field called the load address. This is the address at

which the file will be loaded in response to a 'LOAD' command, if

no address is explicitly supplied.

For '.EXE' files, the load address indicates where in memory

the file will be loaded when it is executed.

3. SERIAL DEVICES

CUBIX at any one time can recognize up to eight serial devices,

which are referenced as devices 0 to 7. The device drivers for these

devices may be installed and de-installed dynamically, as the system

is running.

At any one time, one particular device is designated as the system

console, and it is via this device that the DOS and most programs

running under it communicate. The active console device may be

changed by the user at any time with the "SET CONSOLE" command.

When a CUBIX system is first brought up, the following device

assignments are in effect:

Device Description

0 NULL device, never returns a character, and throws

away any characters output to it.

1 SYSTEM CONSOLE, the console terminal or integrated

keyboard/video display appears here.

2-7 System dependant.

Any unused device driver vectors are initialized to zero, which

will cause a device error if the corresponding device is referenced.

CUBIX User Guide Page: 5

4. COMMAND MODE

Whenever CUBIX is expecting a command, it prompts with '\*' on the

console terminal, and is said to be in COMMAND MODE. While in command

mode, DOS commands can be entered from the console terminal,

terminated by the <return> key, and will be executed by the DOS. The

<backspace> or <delete> keys can be used to correct typing mistakes

when entering commands.

4.1 Return Codes

When a DOS command, utility, or application program terminates,

it passes back to its caller (usually DOS) a return code

indicating if it completed successfully or not. A return code of

ZERO indicates successful completion, and will not be displayed by

the dos. If a DOS command, utility, or application program returns

a NON-ZERO return code, the DOS will display the message 'Rc=<n>'

where <n> is the value of the return code.

The return code message is in addition to the normal error

messages, and is actually generated by the command mode

interpreter, and therefore will not be displayed unless the DOS

command, utility, or application program is returning directly to

dos command mode.

For information on the meaning of specific return codes, refer

to the appendix on return codes.

5. THE SYSTEM LIBRARY

Whenever a command is given to DOS that it does not recognize as a

command of its own, It examines the SYSTEM LIBRARY for any file with

a FILENAME which is the same as the command. If a file by that name

is found, it's FILETYPE is examined, and if it is '.EXE', the file is

loaded and just as if the user had entered 'RUN [<system

lib>]<command>'. If the FILETYPE is '.@', then the file is

interpreted as a command file, just as if the user had typed

'@[<system lib>]<command>'. In both of the above cases, the remainder

of the command line is passed to the file as parameters.

The the FILETYPE found by the above search is not '.EXE' or '.@',

then the DOS makes a second search of the system library, looking for

a '.EXE' file with a FILENAME which is the same as the FILETYPE of

the first file. If this is found, it is loaded and run, and is passed

the entire command line (including the origional command) as

parameters.

If none of the above conditions are met, the DOS will display the

message 'Unrecognized command', and reject the command line with a

return code of 255.

CUBIX User Guide Page: 6

The DOS uses a special drive specifier and directory prefix to

identify files belonging to the SYSTEM LIBRARY. When the system is

first brought up the drive and prefix are 'A:[SYSTEM]', and may be

changed by the user at any time with the "SET SYSTEM" command.

Commands entered in the above matter may be preceded by a drive

specifier and/or a directory prefix, which overrides that used for

the system library.

6. COMMAND FILES

Command files provide a method of executing DOS commands and

programs from a text file, instead of from the console keyboard as is

normally done. This is a convenient way of grouping together a number

of separate commands so that they will all be executed by typing only

one command on the console.

To run a command file, simply type '@<filename>' where

'<filename>' is the FILENAME given to the command file.

The following functions are available in command files ONLY:

?A<n>

Advances <n> lines in the command file and continues from there.

?E[<n>]

Exits (terminates) the command file. If <n> is specified, that

value will be the command file's return code. If not specified, the

return code will be that from the last command executed within the

file.

?F<comamnd>

Executes the command contained within the remainder of the line

only if the last DOS command or executed program failed (Returned a

NON-ZERO return code).

?I

Reads the command file parameter variables (?0-?9) from the

console, after prompting with '\*'.

?K

Reads a single keystroke from the console terminal, and returns

the value of the character entered as the return code.

?O<text string>

Outputs the remainder of the line to the terminal.

?P<text string>

Prints the remainder of the line on the terminal, with a line feed

and carriage return at the end.

CUBIX User Guide Page: 7

?R<n> <comamnd>

Executes the command on the remainder of the line only of the last

DOS command or executed program terminated with return code <n>.

?S<command>

Executes the command on the remainder of the line only if the last

DOS command or executed program was successful (ZERO return code).

?0-?9

Parameters to the command file may be referenced by the names from

'?0' to '?9', with '?0' being the name of the command file, and '?1'

being the first parameter.

?= <tokens>

Assigns the tokens on the remainder of the line to the '?0-?9'

command file parameter variables.

??<string1> <string2>

Returns a ZERO return code if the two strings match, 255

otherwise.

?@<file>

Provides a method of chaining command files without the stack

overhead involved with nesting them. The currently executing command

file is terminated, and the new command file is begun.

?\*

Comment, remainder of line is ignored.

Command files may be nested to any depth (Up to available space on

the system stack), HOWEVER!!!, the parameters to a command file are

NOT saved when another command file is invoked, meaning that when one

command file calls another, it's old parameter values will be lost,

and replaced by the those given to the new command file. A useful

trick is to pass any parameters that you need to retain to the new

command file is parameter positions beyond those which it uses,

allowing the origional value to be retained in the new parameter

position.

NOTE: If SSR (System Service Request) #101 is issued from within any

program executing under the command file, all commands files will be

terminated.

CUBIX User Guide Page: 8

7. COMMANDS AND UTILITIES

The following pages contain a list of the commands supplied with

the operating system.

All commands which are not INTERNAL are located in the SYSTEM

directory of your master disk.

Operands to commands which are shown in angle brackets '<>'

identify operands for which the user supplies a value.

Operands to commands which are shown in square brackets '[]'

identify operands which are optional (not required for all forms of

the command).

Commands shown with a type of INTERNAL are commands which are

built into the operating system. They do not have to be loaded from

disk every time they are executed.

APPLICATIONS are commands which reside on disk, and are loaded

every time that they are executed. Applications always return to DOS

when they terminate. When an executing application invokes another

application, the first application is terminated.

UTILITIES are special applications that follow certain rules which

allow them to be invoked from an application without terminating it.

Applications invoking utilities will resume execution when the

utility terminates.

COMMAND FILES are lists of other commands contained within a file,

which are sequentially executed as one command.

SPECIAL commands are commands of another type, which for one

reason or another do not behave exactly as indicated above. These

commands are intimate with operating system, in order to accomplish

unusual tasks.

Most applications and utilities will display a short summary of

the command syntax and available options if given "?" as a single

operand.

@ @

Format : @<command file name> [<operands>]

Type : Special Internal

Area of use : Procedure automation

Related commands : All commands

Qualifiers : NONE

The '@' command initiates the execution of a command file. The

file indicated by '<command file name>.@' is opened and commands to

be executed are read from it.

For more information, consult the COMMAND FILE section of this

manual.

ASM ASM

Format : ASM <pgmname>

Type : Application

Area of use : Software development

Related commands : ASP, BUILD, RAID

Qualifiers : /ERROR - List only lines with errors.

/FAST - Skip optimization passes.

/QUIET - Suppress progress messages.

/SYMBOL - Include symbol table in listing.

/TERM - List output to terminal, not file.

The ASM command invokes the resident 6809 assembler, to assemble a

source file into an intermediate object file. The <pgmname> operand

is the name of the source file to be assembled. The assembler always

assumes a file type of '.ASM'. The file will be assembled, and an

object file ('.OBJ') will be generated. The listing file will be

placed in <pgmname>.LST if it is not directed to the console with the

'/TERM' qualifier.

The '/ERROR' qualifier causes only those lines containing errors

to be displayed in the listing. This is very useful for debugging.

The '/SYMBOL' qualifier causes ASM to append a sorted display of

the symbol table, to the listing file.

The '/FAST' qualifier causes ASM to skip the optimization passes,

causing all offsets to be calculated as 16 bits, all unspecified

memory references will use extended addressing, and forward

references in EQUate statements will not work. This qualifier is

useful to speed up the assembly process, when initially debugging a

program.

Unless the '/QUIET' qualifier is given, ASM will output progress

messages, indicating what stage of assembly it is performing.

For more information, consult the ASM manual.

ASP ASP

Format : ASP <pgmname>

Type : Application

Area of use : Software development

Related commands : ASM, BUILD, RAID

Qualifiers : /COMMENT - Include comments for source.

/QUIET - Suppress progress messages.

/SOURCE - Include source file as comments.

ASP is an Assembly Source code Preprocessor, which accepts

structured constructs such as IF/THEN/ELSE and WHILE/END, and

converts them to standard assembler source code.

ASP accepts a file with the name '<pgmname>.ASP', and produces a

file with the name '<pgmname>.ASM', which is ready to be assembled.

Any errors discovered by the preprocessor are displayed on the

console.

The '/COMMENT' qualifier causes ASP to copy any comments from the

source file into the output file. This is useful in cases where an

ASP source file is being permanently converted to assembler source,

and will be maintained in the assembly source format.

The '/QUIET' qualifier prevents ASP from displaying its progress

messages.

The '/SOURCE' qualifier causes ASP to include each line from the

source file in the output file as a comment, beginning with '\*>>'.

This comment line will appear in the output file immediately

preceding the lines of assembler code generated by it.

For more information, consult the ASP manual.

BASIC BASIC

Format : BASIC [program]

Type : Application

Area of use : Application programming

Related commands :

Qualifiers : NONE

BASIC (Micro BASIC) is a BASIC interpreter for CUBIX and the 6809.

If executed with no operands, an interactive session will be

started. If a program name is supplied, the file program.BAS will be

automatically loaded and started.

For more information, consult the BASIC manual.

BUILD BUILD

Format : BUILD <objname>

Type : Application

Area of use : Software development

Related commands : ASM, ASP, RAID, DOWNLOAD

Qualifiers : /KEEP - Keep the '.OBJ' file.

/MHX - Build downloadable file.

/QUIET - Inhibit output file statistics.

The BUILD command takes an intermediate object file such as is

produced by the assember, and builds a complete program image.

The '/MHX' qualifier causes BUILD to make a Motorola Hexidecimal

Format download image, and place it in a '.MHX' file. If this

qualifier is NOT given, BUILD makes an executable image, and places

it in a '.EXE' file. In all cases, the FILENAME of the created file

will be the same as that of the intermediate '.OBJ' file.

Unless the '/KEEP' qualifier is given, BUILD will delete the

intermediate '.OBJ' file, before writing out the new image file.

The '/QUIET' qualifier prevents BUILD from displaying statistics

about the newly created executable image, as well as the output

filename for both types of image files.

CALC CALC

Format : CALC [expression ...]

Type : Utility

Area of use : General & Software development

Related commands :

Qualifiers : NONE

CALC is a simple Hexidecimal/Decimal desk calculator. If the

operand <expression> is supplied, CALC evaluates it, and displays the

results in both decimal and hexidecimal. If CALC is not given an

operand, it enters an interactive session, and prompts for

expressions, which it evaluates and displays. Entering a NULL line

will terminate interactive CALC.

CALC assumes input numbers are decimal, unless preceded by a

dollar sign (hexidecimal), a percent sign (binary), a '@' sign

(octal), or is enclosed within single quotes, which indicates that

the ASCII character value is to be used.

CALC evaluates expressions from left to right, with no operator

precedence.

The following operators are recognised by CALC:

Monadic (Single operand) operators:

- Negation (calculate negative value)

~ Complement (Invert all zero's and one's).

= Swap (Exchange high and low bytes)

Dyadic (two operand) operators:

+ Addition.

- Subtraction

\* Multiplication

/ Division

\ Modulus (Remainder after division)

| Logical OR

& Logical AND

^ Exclusive OR

CD CD

Format : CD [<directory>]

Type : Command File

Area of use : General

Related commands : SET DEFUALT, SHOW DEFAULT

Qualifiers : NONE

The CD command provides an easy way to change your default

directory prefix. The usual '[' & ']' square brackets surrounding a

directory name are optional with the CD command.

CD displays the full drive and directory name of your new

directory.

If no operand is given to CD, it simply displays the full name of

your current default directory.

CHGDISK CHGDISK

Format : CHGDISK [<command string>]

Type : Special Utility

Area of use : General

Related commands : All Applications & Utilities

Qualifiers : NONE

CHGDISK is a utility that allows you to load an application or

utility program from one disk, and place a new disk in the drive

before it is executed.

If CHGDISK is invoked with no operands. It first prompts for a

command string, allowing you to change disks (In case the command you

wish to execute is on a different disk from the CHGDISK utility).

Once the command is entered, CHGDISK reads the application or

utility program from the disk, and prompts for you to change disks.

After changing disks, press <RETURN>, CHGDISK will then execute

the loaded program.

If a command string is given as an operand to CHGDISK, it will

load it from disk immediately. In this case, the command must reside

on the same disk as CHGDISK.

CHKDISK CHKDISK

Format : CHKDISK <drive>

Type : Application

Area of use : Disk Maintenance

Related commands : FORMAT

Qualifiers : /NOALLOC - Supress allocation test.

/NOMEDIA - Supress disk media test.

/QUIET - Supress progress messages.

/REBUILD - Rebuild disk allocation

without prompting.

CHKDISK tests the integrity of a disk.

CHKDISK first verifies the consistency of the file system, by

ensuring that all files are linked correctly, containing no

unallocated sectors and that no two files share common sector

linkage. CHKDISK also detects at this time if any sectors are marked

as allocated, but not used in any files. The '/NOALLOC' qualifier

will cause CHKDISK to skip this test.

CHKDISK next performs a media test, verifying that all sectors on

the disk are usable. The '/NOMEDIA' qualifier will cause CHKDISK to

skip this test.

During both tests, CHKDISK keeps an internal copy of the disk

linkage table, which it updates to repair any errors it encounters in

the allocation test, or to mark as allocated any unusable sectors it

encounters in the media test.

If errors are encountered in either test, CHKDISK will prompt with

'Write updated link table? '. Entering 'Y' will cause CHKDISK to

write its updated link table to the disk, thereby releasing any

misallocated sectors, and marking as busy any unusable sectors.

The '/REBUILD' qualifier supresses the prompt, and causes CHKDISK

to update the disk link table if any errors were encountered.

CONCAT CONCAT

Format : CONCAT <dest\_file> <file1> [<file2> ...]

Type : Application

Area of use : File Maintenance

Related commands : COPY, TYPE

Qualifiers : /QUIET - Suppress size report.

CONCAT reads one or more input files, and writes the data from all

of them to one destination file.

Unless the '/QUIET' qualifier is specified, CONCAT reports the

total number of lines written.

Only ASCII text files will be correctly joined by the CONCAT

command.

CONNECT CONNECT

Format : COnnect <device>

Type : Internal

Area of use : Communications

Related commands : DOWNLOAD, INTIO, STTY

Qualifiers : NONE

The CONNECT command creates a logical connection between the

console device, and the specified device. Any characters received

from that device will be sent to the console device. All characters

except for the ESCAPE character which are entered at the console will

be transmitted from the specified device. Entering ESCAPE at the

console terminates the connect command.

On systems where the console device is an integral keyboard and

video display, CONNECT provides an easy way to emulate a dumb

terminal.

For external serial console devices, CONNECT is more often used as

a "pass through" mechanism to access a communications medium such as

a modem or switched network.

COPY COPY

Format : COPY <source spec> [<dest spec>]

Type : Application

Area of use : File Maintenance

Related commands : CONCAT, TYPE, COPYDISK

Qualifiers : /MOVE - Move file instead of copy.

/NEW - Don't overwrite existing files.

/PROT - Copy file protections.

/QUIET - Suppress progress messages.

/REPLACE - Replace files without prompt.

The COPY command copies files from one location to another.

The source file specification may contain wildcards ('\*'), in

which case all files matching that pattern will be copied. Wildcard

('\*') characters in the destination specification indicates sections

of the original name which are to be retained.

If the destination filename and type are not specified, they

default to '\*', indicating the original file name and type.

If the destination directory is not specified, it defaults to the

current default directory.

The '/MOVE' qualifier causes copy to move the file, instead of

copying it. For source and destination on the same disk, COPY simply

renames the file to the new directory and name. For locations on

different disks, COPY copies the file, then DELETE's the original.

The '/NEW' qualifier prevents COPY from copying or moving any

files if the destination file already exists. If this qualifier is

not specified, COPY will prompt for the action to take. Enter 'R' to

replace the old file, 'C' to change the name of the file being

written, or 'S' to skip the file and not copy it. The '/REPLACE'

qualifier causes COPY to always replace existing files specified by

the destination operand.

Unless the '/QUIET' qualifier is specified, COPY displays the

names of all files it is copying.

COPYDISK COPYDISK

Format : COPYDISK <source drive> <dest drive>

Type : Application

Area of use : Disk Maintenance

Related commands : FORMAT, CHKDISK, COPY

Qualifiers : /PROMPT - Prompt to insert disks.

/QUIET - Suppress progress messages.

COPYDISK copies an entire disk to another disk.

The disks must be the same size. COPYDISK will fail with the error

message 'Incompatable drive sizes' unless this is true.

The '/PROMPT' qualifier causes COPYDISK to prompt, allowing you to

change the disks before the copy operation starts.

If the source and destination are specified as the same drive,

COPYDISK will prompt you to insert the source and destination disks,

allowing a disk copy to be performed on a single drive.

Unless the '/QUIET' qualifier is specified, COPYDISK will output a

running display of the operation being performed, and the sector

number that it is operating on.

CREATE CREATE

Format : CReate <filename> [<addr>]

Type : Internal

Area of use : File maintenance

Related commands : DELETE, FILES, LOAD, SAVE

Qualifiers : NONE

The CREATE command creates a file with the specified name. If the

optional address value is specified, the file is created with that

load address. The created file is one block in size.

The load address of a file is stored in that file's directory

entry, and specifies where in memory the file will be loaded in

response to a "LOAD" command which does not specify an address. In

the case of an executable (.EXE) file, the load address determines

where in memory the file will be loaded when is is executed.

The DOS will automatically create a file if an attempt is made to

write to a file which does not exist. The CREATE command is used

mainly to create files with a particular load address. CREATE is also

useful from within command files, because it will fail (non-zero

return code) if the file it attempts to create already exists. This

provides a method for ensuring that a particular command does not

overwrite an existing file.

DELETE DELETE

Format : DELete <filespec>

Type : Internal

Area of use : File maintenance

Related commands : CREATE, FILES, LOAD, SAVE

Qualifiers : NONE

The DELETE command permanently removes a file from the disk.

Any disk sectors used by the file will be released, allowing them

to be reused by other files.

If <filespec> does not contain wildcard characters, DELETE will

simply delete the named file.

If <filespec> does contain wildcard characters, the DELETE command

will prompt with the name of each file it finds which matches that

pattern, allowing the following responses:

Y - Delete the file and advance to the next one.

N - Retain the file and advance to the next one.

Q - Retain the file and terminate the delete command.

A file will not be deleted if it does not have 'D' delete

permission set in its protection string.

DIR DIR

Format : DIR <dir spec>

Type : Application

Area of use : File maintenance

Related commands : FILES, SET FILE, SHOW FILE, DIRF

Qualifiers : /DISK - Include file disk address.

/LOAD - Include file load address.

/SIZE - Include file size.

/TOTAL - Display only directory names

and total file counts.

/NOHEADER - Suppress directory header.

/PROT - Include file protections.

The DIR command displays the names of files on the disk in a

format which is easier to read than that of the FILES command.

The operand <dir spec> may be any directory or file specification

which is valid for the FILES command.

The names to be shown are sorted and displayed as well as the

total number of files.

DIRF DIRF

Format : DIRF <file> <pattern> [<prefix> [<postfix>]]

Type : Utility

Area of use : File maintenance

Related commands : FILES, DIR, @

Qualifiers : /NODISK - Don't include drive prefix.

/NODIRECTORY - Don't include directory.

/NOTYPE - Don't include file type.

The DIRF command produces a directory listing and writes it to a

file. The directory listing is of similar format to that of the FILES

command.

The names of all files matching <pattern> are written one per line

to the specified file.

If the <prefix> is specified, it must be enclosed within unique

delimiter characters, and will be placed ahead of each filename in

the output file.

If the <postfix> is specified, it must be enclosed within unique

delimiter characters, and will be placed following each filename in

the output file.

The '/NODISK', '/NODIRECTORY' and '/NOTYPE' qualifiers may be used

to suppress the corresponding portion of the filename from the output

command.

This command is very useful for creating command files which

perform an operation on many files.

EG: "DIRF SETEXE.@ \*.EXE 'SET FILE ' ' PROT=E'"

DOWNLOAD DOWNLOAD

Format : DOwnload <device> [<filename>]

Type : Internal

Area of use : Communications

Related commands : BUILD, CONNECT, INTIO, STTY

Qualifiers : NONE

The DOWNLOAD command accepts a standard Motorola hexidecimal

format load file from the specified serial device, and loads it into

memory at the address contained in the load records of the file.

If the optional filename is specified, DOWNLOAD also writes the

memory image to that file.

If the specified file does not already exist, it will be created,

and the load address will be set to the address contained in the

first load record in the download file.

DOWNLOAD provides an easy method up updating system software over

a remote serial link.

ED ED

Format : ED <filename>

Type : Application

Area of use : File maintainance, Development

Related commands : ED, EDIT

Qualifiers : NONE

ED is a tiny screen oriented text editor, which operates using a

VT100 or compatable terminal.

Pressing the HELP key from within ED will display a short summary

of the ED commands.

This editor has been superceeded by the 'EDT' screen editor, and

is provided for backwards compatabliity only.

For more information, consult the ED manual.

EDIT EDIT

Format : EDIT <filename> [<command file>]

Type : Application

Area of use : File maintainance, Development

Related commands : ED, SDE

Qualifiers : /BATCH - Read commands from file.

/QUIET - Don't display editing session.

EDIT is a simple line oriented text editor. It is used to create

and edit disk files containing ASCII text. While editing a file, EDIT

prompts with '.', and has the following commands available:

B - Go to beginning of file.

1 C/<s1>/<s2>/ - Substitute <s2> for <s1> on current line.

1 CA/<s1>/<s2>/ - Substitute all <s2> for <s1> in entire file.

1 CD/<s1>/<s2>/ - Sub. all <s2> for <s1> in entire file, display.

1 CH/<s1>/<s2>/ - Substitute all <s2> for <s1> in current line.

2 DE <n> - Delete <n> lines, 1 if <n> not specified.

E - Goto end of file.

EX - Exit and save file.

I - Insert lines ahead of current, null to exit.

2 N <n> - Advance <n> lines, 1 if <n> not specified.

2 P <n> - Print <n> lines, 1 if <n> not specified.

QU - Quit editor, throw away changes.

3 :R - Restore position in file from last save.

3 :S - Save position in file.

2 U <n> - Backup <n> lines.

2 + <n> - Duplicate current line <n> times.

/<string> - Move to next line containing <string>.

? - Display size of file.

\*<command> - Execute DOS command.

4 $<string> - Save ESC string.

1) '/' may be any unique delimiter character not appearing within

<s1> or <s2>. Trailing '/' is optional.

2) The value of <n> defaults to one (1) if no value is given.

3) The restored position may move if lines are inserted, changed,

or deleted above the saved position.

4) The given string is inserted at the end of the input line

(Including the carriage return) whenever ESC key is pressed.

When the '/BATCH' qualifier is given, EDIT will read its input

commands (and text) from the file specified by <command file>. The

'/QUIET' qualifier can be used to prevent the editing session from

being displayed.

EDT EDT

Format : EDT <filename>

Type : Application

Area of use : File maintainance, Development

Related commands : ED, EDIT, TTYPATCH

Qualifiers : NONE

EDT is a fully featured text editor for use under CUBIX. It

supports both line and visual (screen) oriented modes of operation,

and has many powerful commands.

When operating in visual mode, EDT uses the standard "TTY"

interface, allowing the "TTYPATCH" utility to be used to define the

terminal control codes and functon keys used.

For more information, consult the EDT manual.

FILES FILES

Format : FIles [<filespec>]

Type : Internal

Area of use : File maintenance

Related commands : CREATE, DELETE, DIR

Qualifiers : NONE

The FILES command displays the names of files on the disk.

If <filespec> is a directory prefix, then all files having that

directory prefix are displayed.

If the operand is a full file specification (which may contain

wildcards), FILES displays the names of all files on the disk which

match that pattern.

If no operand is given, the names of all files in your current

default directory are displayed.

FLINK FLINK

Format : FLINK <filename> [<filename> ...]

Type : Utility

Area of use : File maintenance

Related commands : CHKDISK

Qualifiers : /UNLINK - Unlink file from directory.

The FLINK command creates one or more directory entries with the

given name(s), which are linked to the disk data of the first file.

This has the effect of making one file accessable by multiple names

(Perhaps in different directories, or with different load address or

protections).

Care MUST be taken to never DELETE a file which is linked to

another file, because the data area (used by all linked files) on the

disk will be released, and all other linked directory entries would

be left pointing to unallocated (or worse reused) file system space.

The CHKDISK utility will report an allocation error for all but

one of the linked files. This occurs because CHKDISK goes through the

motions of deleting all files (without actually deleting them) in

order to determine that all sectors that are allocated an the disk

are actually used within files.

When the '/UNLINK' qualifier is used, FLINK removes the directory

entries for all named files, WITHOUT releasing the data space

allocated to them. This allows the multiple directory entries created

by FLINK to be removed leaving the disk data intact for other file

links.

FLINK/UNLINK may also be used to get rid of a file in which a

known bad sector exists, thereby insuring that those sectors will not

be re-used.

The CHKDISK utility can be used to recover disk sectors accidently

lost by unlinking a file with no other directory entries pointing to

it.

If a linked file is accidently deleted, use CHKDISK to determine

the names of the other links, and then FLINK/UNLINK to remove those

entries.

FORMAT FORMAT

Format : FORMat <drive> [Address=<value>

Cylinders=<value>

Directory=<value>

Heads=<value>

Interleave=<value>

Sectors=<value>]

Type : Internal

Area of use : Disk maintenance

Related commands : SET DRIVE, CHKDISK

Qualifiers : NONE

The FORMAT command formats the specified disk, and creates a file

system on it. After displaying the selected parameter settings, the

FORMAT command prompts with 'Ok?'. Entering 'Y' will format and

initialize the disk. 'N' cancels the command.

All disks used on the system must be formatted before use.

The CYLINDERS= parameter specifies the number of cylinders which

are to be formatted on the disk.

The DIRECTORY= parameter specifies the number of extra disk

sectors which are to be allocated to the directory on the disk.

Although the directory is automatically extended when more directory

space is needed, creating the disk with a large enough directory

keeps that directory in one place at the start of the disk, and

reduces the amount of disk activity needed to access files. If no

DIRECTORY= parameter is specified, three extra sectors (four total)

are allocated. Each sector provides space for 16 file entries.

The ADDRESS= parameter specifies the physical device address of

the drive to be formatted.

The HEADS= parameter specifies the number of heads (sides) that

are to be formatted on the disk.

The INTERLEAVE= parameter specifies the interleave factor which is

used when formatting the disk.

The SECTORS= parameter specifies the number of sectors which are

to be formatted on each track of the disk.

FORTH FORTH

Format : FORTH

Type : Application

Area of use : Application programming

Related commands :

Qualifiers : NONE

FORTH is an implementation of the FORTH language. This version of

FORTH is unique in that it compiles to directly executable 6809

machine code, instead of the usual intermediate interpreted code.

This results in much faster execution than is possible with most

FORTH systems.

For more information, consult the FORTH manual.

HELP HELP

Format : HELP [<topic> [<subtopic...>]]

Type : Utility

Area of use : System & Application documentation

Related commands :

Qualifiers : /<library> - Select help library.

/ - Display available libraries.

The HELP command, will display the selections in the requested

help library, and will prompt for topics and subtopics, displaying

the help information associated with each item selected. If the

response to a topic or subtopic prompt contains more than one word,

HELP will assume the subsequent words to be additional subtopics.

A list of topics/subtopics may be given to HELP as parameters, and

will be processed as if they were entered in response to a 'Topic?'

prompt. Help will accept a '?' in response to a prompt, which causes

it to re-display the text associated with that prompt. A null line,

entered as a response to a prompt, causes HELP to back up one

subtopic level, and prompt again. If help is given a null line in

response to the 'Topic?' prompt, HELP will terminate.

A CONTROL-C entered in response to any prompt, will cause HELP to

terminate.

Help will accept a library name, as a qualifier, denoted by using

the slash '/' character. If this is given, HELP will look for a file

with the name [HELP]<library name>.HLP, and will use it as the source

of help information. If no help library name is specified, HELP

assumes the default help library '[HELP]SYSTEM.HLP'.

A list of the available HELP libraries can be obtained by typing

'HELP/'.

HEXED HEXED

Format : HEXED <filename>

Type : Application

Area of use : File patching

Related commands : EDIT, EDT, TTYPATCH

Qualifiers : NONE

HEXED is a screen oriented hexadecimal file patch utility. It uses

the "TTY" screen interface, which allows it to be customized with the

"TTYPATCH" utility for use on any terminal.

HEXED displays a 256 byte portion of the file in HEX/ASCII dump

format, and allows either the HEXADECIMAL or ASCII display to be

edited.

When the cursor is positioned in the HEXADECIMAL display, typing

two HEX characters will update the value under the cursor and advance

to the next byte.

When the cursor is positioned in the ASCII display, typing any

ASCII character will replace the character under the cursor and

advance to the next byte.

The following special keys are recognized by HEXED:

UP ARROW - Move cursor up.

DOWN ARROW - Move cursor down.

RIGHT ARROW - Move cursor to the right.

LEFT ARROW - Move cursor to the left.

PAGE UP - Display preceeding 256 bytes.

PAGE DOWN - Display following 256 bytes.

HOME - Go directly to the start of the file.

END - Go directly to the end of the file.

F1 - Execute line mode command.

F2 - Re-execute last line mode command.

F3 - Toggle between HEXADECIMAL and ASCII fields.

The following line mode commands are available:

G<xxxx> - Goto position <xxxx> in file.

Q - Quit editing session and return to dos.

QQ - Quit even if unsaved changes are present.

W - Write file and save changes.

?<string> - Search forward for ASCII <string>.

/<xx> <xx>..- Search forward for HEX byte values.

For more information on terminal setup, see the "TTY" manual.

INTIO INTIO

Format : INTIO [<device>]

Type : Special Utility

Area of use : Serial Communication

Related commands : CONNECT, DOWNLOAD, STTY

Qualifiers : /OFF - Remove Interrupt I/O driver

The INTIO utility installs a temporary device driver which

performs buffered interrupt driven input from the serial device. This

allows the the system to receive serial data at much higher speeds,

and supports XON/XOFF flow control for both receive and transmit data

streams.

The '/OFF' qualifier removes a previously installed interrupt I/O

driver. No other parameters are required when using this qualifier.

INTIO is a hardware dependant utility which may vary between

systems.

LDIR LDIR

Format : LDIR [<file or directory spec>]

Type : Application

Area of use : File & Directory maintenance

Related commands : FILES, DIR

Qualifiers : /TOTAL - Only display total number

of matching directories.

The LDIR command lists the names of all directories on the disk

which contain files matching the specified pattern.

If the pattern specified does not contain an explicit directory

prefix it defaults to '[\*]', which shows all directories that have a

matching file.

If the pattern specified does not contain an explicit filename and

type, it defaults to '\*.\*', which shows the directory if any file

exists with that prefix.

LOAD LOAD

Format : LOad <filename> [<addr>]

Type : Internal

Area of use : File access

Related commands : SAVE

Qualifiers : NONE

The LOAD command loads the specified file from disk to memory at

the specified address. If no address is specified, the file is loaded

at its load address, as indicated by its directory entry.

The file is BLOCK loaded, so a multiple of 512 bytes of data is

always transferred.

MAPL MAPL

Format : MAPL

Type : Application

Area of use : Application programming

Related commands : MOVEAPL

Qualifiers : NONE

MAPL (Micro APL) is an interactive computer language which is a

subset of the APL language. MAPL has it's own library system of

workspaces, and supports user defined functions and variables,

recursion, and many of the basic operators from APL. MAPL supports

integer and character types.

Unlike APL, MAPL does not require a special terminal, but uses

normal upper and lower case ASCII characters to represent the many

special operators.

For more information, consult the MAPL manual.

MOVEAPL MOVEAPL

Format : MOVEAPL <filename> <address>

Type : Application

Area of use : Porting MAPL applications

Related commands : MAPL

Qualifiers : NONE

The MOVEAPL command is used to change the RAM load address of a

MAPL workspace file. This must be done before you can use the MAPL

workspace on a system with a different memory map from the one on

which it was created.

In addition to changing the load address of the file, MOVEAPL

adjusts the symbol table pointers in the workspace to reflect the new

load address of each variable or function.

The correct load address for a particular system may be determined

by examining the load address of the MAPL.EXE image in the [SYSTEM]

directory, and adding an 8K ($2000) offset to it.

RAID RAID

Format : RAID/<dev> [<filename> [<operands>]]

Type : Special Utility

Area of use : Application programming

Related commands : ASM, ASP, BUILD

Qualifiers : /<dev> - Debug to this device.

RAID (Runtime Aid for Interactive Debugging) is a program which

aids in the debugging process, by allowing the user to monitor and

control the execution of a program under test.

RAID includes a full 6809 disassembler and contains code

breakpoint and software single step facilities.

The '/<dev>' qualifier causes RAID to communicate with the

specified serial device (0-7) for the duration of the debugging

session. If no device is specified, RAID assumes the current console

device.

If <filename> is specified, RAID loads the file at its load

address, sets the program counter to that address, points the 'Y'

register to the remainder of the command line allowing the program

under test to receive its operands, and then begins the interactive

debugging session, prompting with 'Raid>'.

If no operands are given, RAID simply enters the interactive

debugging session.

When it is first invoked, RAID attaches itself to the DOS in such

a way that RAID becomes essential for the DOS to operate. If RAID is

terminated in any abnormal way, it is advisable to reboot DOS.

For more information, consult the RAID manual.

READ READ

Format : REad <drive> <sector> <addr> [<size>]

Type : Internal

Area of use : Disk maintenance

Related commands : WRITE

Qualifiers : NONE

The READ command reads disk sectors from the specified drive, and

places them in memory at the specified address. If <size> is not

specified, one sector is transferred.

The READ command bypasses the normal DOS file system, and should

only be used by persons familiar with the disk structure during

recovery procedures. It is not intended that READ be used in the

normal every day operation of the system.

REDIRECT REDIRECT

Format : REDIRECT <filename> [<device>]

Type : Special Utility

Area of use : I/O Redirection

Related commands :

Qualifiers : /ALL - Pass ALL character codes

/CLOSE - Terminate redirect

/WRITE - Redirect writes to device

The REDIRECT utility installs a temporary device driver which

causes accesses to the specified serial device to be redirected to a

file.

The '/WRITE' qualifier causes redirect to operate on writes to the

device. Any characters sent to the device will be written to the

named file.

The '/ALL' qualifier causes redirect to write all control codes to

the file exactly as they are sent to the device. If this qualifier is

not given, lines of data are written to the file with only a carriage

return separator, thereby creating a standard text file.

If the '/WRITE' qualifier is not given, redirect operates on reads

of the device. Each request to read a character from the device will

return the next character from the file. When the end of the file is

encountered, the next read will indicate that no character is ready,

after which the original device driver will be restored.

The '/CLOSE' quailfier is used to terminate a previously invoked

REDIRECT. No other parameters are required with this qualifier.

If no <device> parameter is given, REDIRECT operates on the

current console device.

RETURN RETURN

Format : RETURN

Type : Internal

Area of use : Application programs

Related commands : NONE

Qualifiers : NONE

The RETURN command is used to return to an application that has

temporarily entered the dos via a system call.

When an application program enters the dos this way, only internal

commands and utility programs should be run. If any other

applications are executed, the original application will be

terminated.

RUN RUN

Format : RUn <pgmname>

Type : Internal

Area of use : Application programs

Related commands : NONE

Qualifiers : NONE

The RUn command loads the specified file into memory at its load

address and executes it. The file type '.EXE' is assumed.

The indicated file (with the name <pgmname>.EXE) is loaded into

memory at its Load address, and execution is transferred to offset

zero (0) within that code.

The remainder of the command line is made available to the

executing program so that parameters can be evaluated.

A file will not be executed if it does not have 'E' execute

permission set in its protection string.

SAVE SAVE

Format : SAve <filename> <addr> <size>

Type : Internal

Area of use : File access

Related commands : LOAD

Qualifiers : NONE

The SAVE command saves a memory image in the specified file.

The memory image is saved starting at the indicated address (in

hex), for the number of 512 byte disk sectors specified by <size>.

If the file did not previously exist, it is created, and its load

address is set to the value of <addr>.

SEARCH SEARCH

Format : SEARCH <filespec> <search string>

Type : Utility

Area of use : File maintenance

Related commands :

Qualifiers : /BINARY - Treat input as binary file.

/CASE - Convert lower to upper case.

/HEX - Search string is in hex.

/TOTAL - Display only total found.

The SEARCH utility scans specified files, and displays a list of

the occurrences of the given search string. Wildcard ('\*') characters

may be specified in the <filespec> operand. The search string may be

delimited by any character which is not part of the actual search

string.

If the '/BINARY' qualifier is given, SEARCH treats the input file

as a binary file. The $FF end of file character used in ASCII text

files will be treated as data, and SEARCH will scan all the way to

the end of the last sector of the file. The output display will

contain only the offset into the file when this qualifier is used.

If the '/CASE' qualifier is given, SEARCH will convert any lower

case characters from the input file into upper case before doing the

comparison. This allows case insensitive searches to be performed.

NOTE: If the SEARCH command is used from DOS interactive mode, any

lower case characters in the search string will be converted to upper

case.

The '/HEX' qualifier indicates to SEARCH that the search string is

being given as a series of hexidecimal values. No delimiters are

needed when this qualifier is used.

The '/TOTAL' qualifier causes SEARCH to only output the names of

the files being searched, and the total number of occurrences of the

search string in each file.

SET SET

Format : SEt COnsole [Input=<device>

Output=<device>]

DEBug ON|OFF

DEFault <directory>

DRive <drive> [Address=<value>

Cylinders=<value>

Heads=<value>

Sectors=<value>]

FIle <filename> [Load\_address=<addr>

Protection=<protection\_string>]

MEMory <addr> <data> [<data> ...]

MESsage ON|OFF

SYstem <directory>

TRace ON|OFF

Type : Internal

Area of use : System Maintenance

Related commands : SHOW

Qualifiers : NONE

The SET command controls many functions relating to the internal

operation of the operating system.

SET CONSOLE allows you to select the serial device to be used for

console input and/or output.

SET DEBUG enables/disables display of the operating system calls

that are being executed by application and utility programs.

SET DEFAULT changes the default directory prefix that is to apply

to all file accesses not explicitly indicating the directory.

SET DRIVE changes the physical characteristics associated with a

particular disk drive. (See FORMAT)

SET FILE changes attributes of the specified file. Load\_address

specifies the new load address to be assigned to the file. Protection

specifies a new protection string to be assigned to the file:

R - Read permission.

W - Write permission

E - Execute permission.

D - Delete permission.

SET MEMORY alters the contents of system memory. Beginning with

the specified address, the given data values are written sequentially

to memory.

SET MESSAGE enables/disables the display of DOS error messages.

SET SYSTEM changes the system directory prefix that is to apply to

all command searches.

SET TRACE enables/disables the display of commands contained

within a command file as they are executed.

SHOW SHOW

Format : SHOW COnsole

DEBug

DEFault

DRive <drive>

FIle <filename>

FRee <drive>

MEMory <addr> [<addr>]

MESsage

SYstem

TRace

Type : Internal

Area of use : System Maintenance

Related commands : SET

Qualifiers : NONE

The SHOW command displays information about various parts of the

system.

SHOW CONSOLE displays the current assignments of the console

device.

SHOW DEBUG displays the current setting (ON or OFF) of the debug

flag.

SHOW DEFAULT displays the current default directory prefix.

SHOW DRIVE displays the physical characteristics of the specified

drive.

SHOW FILE displays the full name, the load address, the disk

address, and the protection string of the specified file.

SHOW FREE displays the number of free (unused) disk blocks on the

specified drive.

SHOW MEMORY displays the contents of memory between the specified

addresses, in standard hex/ASCII dump format. If only one address is

given, 16 bytes are shown.

SHOW MESSAGE displays the current setting (ON or OFF) of the error

message output flag.

SHOW SYSTEM displays the current system directory prefix.

SHOW TRACE displays the current setting (ON or OFF) of the trace

flag.

SIM80 SIM80

Format : SIM80 <8080 code file name>

Type : Application

Area of use : Program development, compatibility.

Related commands :

Qualifiers : /DEBUG - Invoke the 8080 debugger.

SIM80 is an application program which simulates the execution of

an Intel 8080 microprocessor, by interpreting the opcodes it reads

from memory.

SIM80 loads the 8080 code file at its load address, and begins

interpreting at offset zero (0) within that image. The filetype '.80'

is assumed.

If the '/DEBUG' qualifier is given, SIM80 enters its resident 8080

debugger, after the image is loaded.

For more information, consult the SIM80 manual.

STTY STTY

Format : STTY <device> [SPEED=<baud rate>]

[PARITY=<parity type>]

[BITS=<bits per char>]

[STOP=<# stop bits>]

Type : Utility

Area of use : Serial Communications

Related commands : CONNECT, DOWNLOAD, INTIO

Qualifiers : None

STTY is a utility which sets the options for the serial

communications devices. This is a system specific utility, and the

implementation of parameters to the right of the device number may

vary, depending on the hardware in use.

The standard STTY which functions with 6551 uart devices has the

following options available:

SPEED= is used to set the baudrate of the serial device. The value

of <baud rate> must be one of: 50, 75, 110, 134.5, 150, 300, 600,

1200, 1800, 2400, 3600, 4800, 7200, 9600 or 19200.

PARITY= specifies the parity type, and must be one of: ODD, EVEN,

MARK, SPACE or NONE.

BITS= specifies the number of bits per character, and must be one

of: 5, 6, 7 or 8.

STOP= specifies the number of stop bits transmitted at the end of

each character, and must be one of: 1 or 2.

If only the device number is given, STTY will display the current

settings for that device.

TTYPATCH TTYPATCH

Format : TTYPATCH

Type : Application

Area of use : Terminal customization

Related commands : EDT, HEXED

Qualifiers : NONE

The TTYPATCH utliity provides the capability of modifying the

terminal control codes and function keys used by certain screen

oriented application programs. Any application which uses the "TTY"

interface library to control the terminal may be modified using

"TTYPATCH".

TTYPATCH operates using a line by line menu interface, allowing it

to be used regardless of the terminal type.

Any terminal may be used for screen applications, provided that it

has the following features:

1) Absolute cursor positioning

2) Clear entire screen

3) Clear to end of line

4) Clear to end of display

5) One special video mode (reverse etc)

6) Forward scrolling

For more information, consult the "TTY" manual.

TYPE TYPE

Format : TYPE <filename>|<address> [<device>]

Type : Utility

Area of use : File maintenance.

Related commands :

Qualifiers : /DISK - Display a disk sector chain.

/DUMP - Output in hex/ASCII dump format.

/MEMORY - Display absolute memory contents.

/PAGE - Wait for key every 22 lines.

The TYPE utility displays the contents of a file, a disk sector,

or of main memory. The data is displayed as ASCII text, unless the

'/DUMP' qualiifer is given, in which case it is output in

Hexidecimal/ASCII dump format.

If the '/DISK' qualifier is given, the operand is a disk sector

number, and TYPE will display the contents of that sector as well as

any sectors linked to it in the disk link table.

If the '/MEMORY' qualifier is given, the operand is a memory

address, and TYPE will display the contents of memory, starting at

that address.

If neither the '/DISK' or '/MEMORY' qualifiers are found, the

operand is a file name, and TYPE displays the contents of that file.

The '/PAGE' qualifier causes TYPE to halt the display after every

twenty two lines of output, and wait for a keypress from the console

before proceding.

If the <device> number is specified, TYPE outputs to that serial

device. Otherwise, output is sent to the console device.

When displaying text files, TYPE expands tab characters to an

appriopriate number of spaces assuming eight character tab stops.

WRITE WRITE

Format : WRite <drive> <sector> <addr> [<size>]

Type : Internal

Area of use : Disk maintenance

Related commands : READ

Qualifiers : NONE

The WRITE command WRITES disk sectors to the specified drive, from

memory at the specified address. If <size> is not specified, one

sector is transferred.

WARNING: The WRITE command bypasses the DOS filesystem, and if

improperly used, could DESTROY the contents of a disk. It should be

used only by persons familiar with disk structures during recovery

procedures. Even so, it is recomended that the recovery be performed

on a copy of the original disk.

CUBIX User Guide Page: 9

8. APPENDICIES

8.1 APPENDIX 1 - Commonly used file types

The following are some commonly used file types:

.APL - Saved MAPL workspace.

.ASM - Assembly source file

.ASP - Assembly pre-processor source file

.BAD - Used to indicate file of allocated bad sectors

.DAT - General data file

.DOC - Document processor input file

.EXE - Executable image file.

.HLP - Help libraries and files used by some applications.

.LST - Listing file.

.OBJ - Intermediate object code file.

.OUT - Document processor output file.

.TMP - Temporary files used by some utilities.

.@ - DOS command file.

.80 - SIM80 file of 8080 executable code.

CUBIX User Guide Page: 10

8.2 APPENDIX 2 - Return codes

The following return codes are generated by CUBIX:

0 - Success, command, program, or system request worked.

1 - Operand to command or SSR was invalid or not supplied.

2 - Requested file does not exist on the disk.

3 - Operation failed because file is protected.

4 - Attempt to read file not open for read.

5 - Attempt to write file not open for write.

6 - End of file encountered.

7 - Attempt to create a file failed because the file

already exists on the disk.

8 - Attempt to allocate a disk sector failed because there

is no available free space on the disk.

9 - Physical disk error occured while reading or writing

to the disk.

10 - Attempt to access a serial device for which no device

driver is installed.

11 - Attempt to download a file from a device failed because

of a format error in the file.

255 - Command was not recognized as a valid DOS command.

CUBIX User Guide Page: 11

8.3 APPENDIX 3 - Error messages

1) Operand missing or invalid

Operand to DOS command, utility, or application program was not

supplied, or was of an invalid format.

2) File not found

Requested file does not exist on the disk.

3) File protection violation

Attempt to READ, WRITE, EXECUTE or DELETE the requested file

failed because the file does not have permission for that

operation enabled.

4) File not open for read

Results from an attempt to read a file using System Service

Requests (SSR's) when the FCB specified did not indicate a file

which had been opened for read.

5) File not open for write

Results from an attempt to write a file using System Service

Requests (SSR's) when the FCB specified did not indicate a file

which had been opened for write

6) File already exists

Requested file already exists, and therefore cannot be

CREATEed.

7) Insufficent disk space

Results from any operation which attempts to allocate a free

sector on a disk when no such free sector is available.

8) Invalid device

Results from any operation to a serial device when no device

drivers have been installed for that device.

9) Download format error

Download from a serial device has been aborted due to an error

in the format of the download file.

10) Error processing file: <file specification>

An error has occured during processing for the named file.

Another error message follows, indicating the type of error.

CUBIX User Guide Page: 12

11) Error accessing block <block> on drive: <drive>

A physical disk error has occured on the named drive. Another

message follows indicating the type of error that occured.

8.4 APPENDIX 4 - Sample command files

FILE #1 'CD.@'

?\* This command file allows the user to change his default directory

?\* prefix without having to always specify the square braces around

?\* the new one, although that will still work. The new default is

?\* displayed.

SET MESSAGE OFF

SET DEFAULT ?1

?F SET DEFAULT [?1]

SHOW DEFAULT

SET MESSAGE ON

FILE #2 'ASSEMBLE.@'

?\* This command file runs the ASP assembly source file pre-peocessor,

?\* Then the ASSEMBLER, then the BUILD utility, and deletes the

?\* Intemediate files.

?O ASP...

ASP/Q ?1

?S ASM/T/E ?1

?S DELETE ?1.ASM

?S BUILD ?1

?S DELETE ?1.OBJ