XOS

User's Guide

For XOS version 3.2

August, 1999

Documentation release 3.0.2

Table of Contents

Preface

This manual is intended as a reference for all users of XOS. It provides information on using the system, getting around in the multitasking environment, and configuring XOS to obtain optimum performance from your system.

What is an Operating System?

In its basic form, an operating system is a program that lets the computer run other programs. These programs that run at the command of the user are called applications. The operating system is loaded when the computer is first turned on and stays in memory until the system is shut down. An OS, as the term is usually abbreviated, provides a few basic functions and usually comes with some utilities, or useful programs that let the user do things like copy files, delete files, and so on. These utilities are not part of the basic operating system (called the *kernel* of the OS) because they are not used frequently enough to justify taking up memory. Utilities, then, are applications that are provided with the operating system.

What, then, is an Operating Environment?

This phrase has been creeping, slowly but surely, into the computer world. An operating environment is simply an overgrown operating system; it has many utilities and functions and allows the user to do more things than a simple operating system. Unfortunately, the price for this increased functionality can usually be counted in large amounts of memory and disk space used by the operating environment.

XOS is an operating system designed to run on PCs using CPUs which conform to the Intel 386 architecture. This includes the 80386, 80486, i486, Pentium, and compatible processors manufactured by Intel and others. It fits the definition of operating environment by being greatly enhanced in functionality over the standard Disk Operating System, or DOS. However, XOS does not require excessive amounts of system resources.

Chapter 1

Introduction

XOS is a *multitasking* operating system. In other words, it can run more than one program at the same time; all programs share the system in a safe, controlled environment. XOS can also support several users sharing the system via terminals, and thus is also a *multiuser* system.

Because XOS is a true multitasking operating system, it enables any application program to act like a TSR (terminate and stay resident program). It also enables tasks including network services to run in the background simultaneously with other programs. XOS can support many users, but has no requirement for complex system management functions.

XOS is compatible with DOS. This means that most standard DOS programs will run under XOS, as an XOS *session*. XOS fully supports the DPMI version 0.9 specification, allowing protected mode DOS programs (both 16-bit and 32-bit) to run under XOS.

The DOS file and record locking extensions are implemented, allowing network aware DOS applications to run correctly. Each session can be thought of as a separate machine, with a common file system for all machines. Each task is isolated from other tasks' unruly behavior by the XOS kernel.

In addition, XOS provides a native mode API (Application Program Interface) which provides significantly increased functionality over the DOS API. The native mode API calls can be used in native XOS programs and in DOS programs, providing an easy path for enhancing existing programs.

XOS includes complete networking capabilities based on the Internet protocol suite (usually referred to as TCP/IP). It supports both Ethernet and serial port network connections.

XOS can be configured to provide a command line interface which is exactly compatible with DOS or to provide an extended interface which, while not completely DOS compatible, retains the general feel of the DOS user interface while offering significantly enhanced features.

All programs, real mode DOS, protected mode DOS (DPMI), as well as native XOS, execute transparently via the same user interface and from the same command prompt; a user need not be concerned with the operating mode of a program before running it.

All files on the system are accessible by all programs. There is no need to move files between DOS and 32-bit disk areas or to run DOS programs in special compatibility boxes.

The screen and keyboard connected to the machine that is actually running XOS are collectively called the *console terminal.* Those terminals connected to the host system via serial I/O are known as *remote terminals*. These act like terminals on a mainframe system. The console terminal is capable of interfacing with multiple programs by means of XOS sessions: Each program is assigned a screen, to which it displays output. The user switches between screens and thus sessions, with two keystrokes. Most programs continue to execute as background tasks while the user is working in another session.

In brief, XOS is designed as an enhanced DOS compatible environment allowing the continued use of existing software written for DOS while supporting a multitasking, multiuser environment and advanced 32-bit applications.

Multitasking: A Simple Introduction

XOS is a *multitasking* operating system. This simply means that XOS allows a computer to run several programs at once. Unlike some other operating systems, programs do not go to sleep if they aren't being used. They will continue to run normally whether they are in the *background* (meaning that the user is not looking at them) or in the *foreground* (the user is watching the program and typing input into it.)

A multitasking environment implies several changes from the standard DOS environment. For example, all programs are sharing the same computer, using the same *resources*. These resources are the computer's memory, disk, screen, and keyboard. Though XOS makes every attempt to keep programs from getting unruly, there are certain programs which simply must have exclusive use of one or another *resource.* An example of this is the CHKDSK program, which checks disks for errors and problems. In order to fix these problems correctly, CHKDSK needs to have the disk it is checking all to itself.

In general, resource sharing is either allowed or not allowed, depending on the operating system; it is perhaps best described by analogy.

The most basic of multitaskers is the virtual machine, which creates several rooms within the computer system and spends its time running from room to room, activating and deactivating each program in turn. The operating system carries all the system resources with it when running around and assigns them like library books, on a first-come, first-served basis. Also like library books, any other programs in other rooms must wait for the first program to finish before they can use the book.

XOS implements an "enhanced virtual machine" environment, which has its roots in the basic multitasker, but implements a more logical and complete resource sharing system. Again, the operating system creates rooms with programs in them, each isolated from the other. Also, the operating system maintains control of the system resources (which, you will remember, are such things as the display, keyboard, disk, and I/O ports). However, in this case, the books (resources) are not lent out to each program; rather each program asks the operating system to perform a task, and XOS then uses the books. Think of it as a non-circulating book collection, where the borrower asks the librarian, rather than having the books lent to an individual borrower.

In addition, each program can send and receive letters from other programs in other rooms. These letters are called Interprocess Messages and can transmit any kind of information between XOS aware (native XOS) programs on the system.

Since the XOS environment is based on the extended virtual machine architecture, hung or otherwise misbehaving software will not require a system reboot to terminate, and will not affect other programs running on the system. The operating system can simply be told to remove the offending room (session) and the program in it will also disappear.

Chapter 2

Overview of XOS

This chapter provides a general overview of the XOS user interface. This includes descriptions of various items used throughout the system, such as device names and environment strings. It also describes the use of various special keyboard keys.

This chapter is not intended to be a basic introduction to the use of a computer system. It is assumed that the reader is already familiar with the use of some computer system and will be using this material mainly as a reference describing how XOS works.

It should be noted that all of the information in this manual refers to the operation of the standard XOS command processor or the various standard XOS utility programs. When a user program is running, it takes over control of the console and keyboard and may implement a completely different user interface than what is described here. The standard interface described here will again be in effect after such a program terminates.

Using the Keyboard

The keyboard provides the primary means used to interactively enter commands and data. The standard XOS command processor and most of the standard XOS utilities use the keyboard in line mode. In line mode, information is entered as lines with each line terminated with the `<ENTER>` character. Until the terminating character is typed, characters already typed can be edited using a number of special editing keys. The command processor supports two different sets of editing keys. The first set uses most of the function keys and is compatible with the line editing features of all versions of DOS. The second set uses the editing and cursor keys and is basically compatible with the editing features provided by the DOSKEY TSR under DOS. This second set is also available with user programs which use line mode input.

The caps lock, numlock, and scroll lock keys provide their normal functions. The caps-lock key toggles the caps lock state, which is indicated by the caps lock light on the keyboard. When the caps lock state is on, all alphabetic keys generate upper case letters when neither shift key is pressed and lower case letters when either shift key is pressed.

The numlock key toggles the numlock state, which is indicated by the num-lock light on the keyboard. When the numlock state is on, the numeric keypad keys generate the characters indicated on the keys. When the numlock state is off, the numeric keypad keys act as editing keys as indicated.

The scroll lock key toggles the scrolllock state, which is indicated by the scroll lock light on the keyboard. The XOS command processor does not directly use the scroll lock key. The scroll lock state is ignored by the command processor.

Basic line editing

The basic line editing functions are available when any program is using line mode input. The standard XOS command processor uses line mode input.

Characters typed on the keyboard are inserted into the input line at the cursor position. The cursor is then positioned after the character just inserted. When in insert mode and the cursor is not at the end of the line, all characters to the right of the cursor are shifted right one position to make room to insert the character typed. When in overstrike mode and the cursor is not at the end of the line, the character typed overlays the character under the cursor.

The following table summarizes the basic line editing keys.

Key	**Function performed**
Insert	Toggle between insert and overstrike mode
Home	Move cursor to beginning of line
End	Move cursor to end of line
Backspace	Delete character before cursor
Delete	Delete character under cursor
Esc	Delete entire input line
Ä	Move cursor right
î	Move cursor left
^U (Ctl U)	Delete all characters from cursor to beginning of the line
^X (Ctl X)	Delete entire input line

Command recall

The command recall feature operates only when the standard XOS command processor is in use. This feature allows previously executed commands to be recalled, optionally edited, and then executed without having to re-enter the command. The ⌠ (cursor up) and ⊕ (cursor down) keys are used to scan through previously typed commands. Typing ⌠ recalls the previous command. The recalled command is displayed with the cursor positioned at the end of the command just as if it had just been entered from the keyboard. The editing keys described above can then be used to modify the command if desired. The command is executed by pressing <ENTER>. Typing ⌠ again recalls the next previous command, etc. Typing ⊕ recalls the next more recent command. Typing the page up key recalls the oldest command. Typing the page down key recalls the most recent command. If an exclamation point followed by a string of characters is entered, the most recent command beginning with the string entered will be recalled. If an exclamation point followed by a number is entered, the command referenced by the number will be recalled. The number of commands which are saved for recall is specified with the HISTORY command (see chapter 3, page 91). The HISTORY command can also be used to display a list of all saved commands.

DOS compatible line editing using function keys

The standard XOS command processor also supports DOS compatible line editing using the function keys. This capability is actually a combination of command editing and command recall. It allows character to be retrieved from the most recently executed command and insert them into the command currently being entered from the keyboard.

The following table summarizes the use of the function keys.

Key	Function performed
F1	Insert corresponding character from previous command
F2	Insert character from previous command up to but not including the character typed after F2
F3	Insert the entire previous command
F4	Skip to character in previous command matching character typed after F4
F5	Replace the previous command with the command being entered
F6	End-of-file character (same as control-Z)
F7	Show command history
F8	Search for command that begins with the previous entered characters
F9	Prompt for a command number to recall

Virtual Screens and the System Menu

XOS associates multiple virtual screens with the console display. Each virtual screen is completely independent of all other virtual screens and can be used to control a session, that is, a command processor and one or more user programs. The number of virtual screens is established during system initialization. Most systems will be configured for between four and eight virtual screens, although up to twenty virtual screens can be supported.

The system uses the `<Print-Screen>` key to invoke a system menu. The system menu is used to select which virtual screen will be displayed on the real console display.

The system menu displays the number of each virtual screen (1-9,A-K) and the name of the program being run in the session which is controlled by the corresponding virtual screen. A virtual screen can be selected by entering its number. The menu can be cleared without selecting a different virtual screen by entering `<ENTER>` or `<Print-Screen>`.

Additional functions will be implemented through the system menu in future versions of XOS.

Devices

A device is generally thought of as some physical device connected to the system which can be used to input or output data. Disk drives, tape drives, and terminals are some examples of such devices. XOS extends the definition of a device to also include more abstract devices. For example, the XOS IPM device provides a mechanism for transferring data between processes. Several network devices are implemented which provide access to various levels of the network protocol stacks, all of which use the same physical network interface.

XOS groups devices into device classes. A device class usually consists of all devices of a particular type. For example the DISK device class consists of all random access mass storage devices directly connected to the system. The TRM device class consists of all terminal devices, which includes serial ports and the console display and keyboard.

Device Names

XOS uses a 1 to 16 character alpha-numeric name to identify all devices, including disks. The standard convention is that a device name consists of 1 or more letters which identify the device class followed by a numeric decimal unit number which identifies the device unit within the class. This may optionally be followed by a single letter followed by another decimal value which identifies a sub-unit. The letter is chosen to indicate the kind of sub-unit. For example, TRM0S3 identifies virtual screen 3 of terminal 0 and D0P2 identifies partition 2 of hard disk 0. TRM2 identifies a serial port (which does not support virtual screens) and D0 identifies an individual hard disk, independent of its partition structure.

The initial alphabetic part of the name is usually derived from the device class name, sometimes with one or more letters removed to make the name shorter to allow for unit and sub-unit numbers, but this is not a requirement. A class driver is free to name its devices independent of its class name. For example, the XOS DISK device class uses D to name hard disks and F to name floppy disks. The TRM device class, which includes the console display/keyboard and serial ports, uses TRM to name all of its devices.

This format for device names is only a convention. It is not enforced by XOS. The only actual restriction is that device names must begin with a letter and contain only letters and digits.

Disk class devices also each support two alternate names. The first alternate name can be set using the DEVCHAR command or the device characteristics system call, and is normally used to name a disk according to the DOS disk naming conventions. That is, as A, B, C, etc. A utility program, DOSDRIVE, is provided which assigns the same DOS format name to each disk in the system as would DOS. This program is normally run as part of the XOS start up procedure. The second alternate name is set to be the volume name of the disk, as specified by the disk's file structure. This feature is not used for DOS disks since they do not have a well defined volume name. DOS disks do have a volume label, but it is often not properly formatted for use as a device name and almost always was not chosen as a unique name to use when referencing the disk.

The XOS disk names uniquely identify a disk unit and, where applicable, a partition. Unlike the DOS disk names, the XOS names do not change when the system configuration is changed by adding or removing disks.

Floppy disks are identified as Fn, with F0 being the DOS A and F1 being the DOS B. If a system contain additional floppy disks, they are normally named as F2, F3, etc., although there is no restriction that the number be contiguous.

Hard disks are named as Dn and DnPm. The Dn format name represents an entire hard disk independent of its partition structure. It is normally only used when referencing the partition table on the disk, although it can also be used to perform raw mode reads and writes to the entire disk, independent of the disk's partitions. There is no corresponding DOS disk name, although the name of the first partition can be used under DOS for some of these functions. The DnPm format name represents an individual disk partition. D0P1 is the first partition on disk unit 0 (the first hard disk). This is normally the DOS disk C, although since different versions of DOS set up the disk partition table differently, this is not always the case.

XOS attempts to assign partition numbers in the same order as DOS, but may not succeed in all cases, especially with disks which were set up using third party disk partitioning software. XOS first scans the partition table looking only for standard DOS partitions (including huge partitions) and assigns these partition numbers in the order they are found. It then scans the partition table looking for extended DOS partitions and assigns partition numbers to the logical volumes contained in the extended partitions in the order they are found. Finally, the partition table is scanned looking for any remaining non-DOS partitions and assigns partition numbers to them in the order they are found.

File Specifications

A file specification completely specifies an individual file on some disk in the system or on a remote disk. It has the following general format:

DEV:NETADDR::RDEV:PATH\NAME.EXT

where DEV is the device name (as discussed above), NETADDR is an optional network address (used only when DEV specifies a network device), RDEV is the remote device name (also used only when DEV specifies a network device), PATH is the directory path, NAME is the file name, and EXT is the file extension.

If DEV and NETADDR are not included, the device Z: is assumed. If DEV is not included but NETADDR is included, the device NET: is assumed. Z: and NET: are logical names (see below) which are defined to specify the desired default devices.

When an XOS network device is used to access files on a non-XOS remote system (also referred to as a foreign system), the format of the part of the file specification following the :: is specified by the remote system. It may have any format required by the remote system subject only to the requirement that characters which XOS does not allow in a file specification cannot be used. One such character is / (which is reserved as a switch/option prefix by XOS). When accessing files on a remote UNIX system, XOS converts the \ character to /, allowing UNIX file specifications to look much like XOS file specifications.

Note that while the XOS device drivers treat the remote part of the file specification as an arbitrary string, some XOS utilities (such as DIR) make some assumptions about the format of a file specification. This means that some advanced options (such as the ... notation) may not work with all foreign systems.

When a network device is specified, the remote device can also be a network device, allowing any level of multiple remote access. This is generally not a desirable (efficient) way to access remote files, but may be necessary in the special case where a remote system has access to a second network to which the local system is not connected and does not provide automatic routing or bridging between the networks.

Wildcard File Specifications

Most XOS system calls which take file specifications as arguments allow partially specified or wildcard file specifications. A wildcard file specification includes one or more of the following wild-card characters:

Character	Description
?	Matches any single character
*	Matches any number of characters
{	Begins wildcard lists
}	Terminates wildcard list

For example, the wild-card specification A*X.ZZZ would match the names AX.ZZZ, ABX.ZZZ, and ABCX.ZZZ. It would not match ABC.ZZZ or XA.ZZZ. A?X.ZZZ would match ABX.ZZZ but would not match AX.ZZZ or ABCX.ZZZ, since the ? matches exactly one character. The range of characters matched by a * terminates at the period which separates the name and extension. Thus *.XY would match any file name with the extension XY but would not match ABC.DXY. *.*XY would match ABC.DXY, however. It should be noted that this is an extension of the DOS handling of the * character, since DOS only allows * at the end of the name or extension parts of a file specification. All valid DOS wildcard file specifications will behave the same under XOS as under DOS.

The wildcard list feature provides an additional extension to DOS wildcard handling. There are two kinds of wildcard lists: single character lists and compound lists. A single character list has the form:

{abcd}

where 2 or more characters are delimited by braces. This will match any one of the characters specified. For example:

A{DEF}X.ZZZ

would match ADX.ZZZ, AEX.ZZZ, or AFX.ZZZ only. A compound list has the form:

{{one}{two}{three}}

where 2 or more items are delimited by braces. Each item is delimited by braces. This will match any one of the items. For example:

NAME.{{EXE}{COM}{IMG}}

would match NAME.EXE, NAME.COM, or NAME.IMG.

The use of wildcard lists is optional and must be enabled by setting the O$WILDLIST bit in the command bits argument for the XOS system call used to search for the file. Wildcard lists cannot be used with DOS system calls. This restriction is necessary to preserve full DOS compatibility since left and right braces are often used as DOS file name characters.

Destination Wildcard File Specifications

XOS uses an extension of the wildcard file specifications described above to specify destination names for copy or rename operations.

A destination wildcard file specification looks much like a wildcard file specification but actually performs quite a different function. It specifies how a destination file specification is constructed using information from the successful search for the source file. The following special characters are used in a destination wildcard file specification:

Character	Description
?	Replace with corresponding ? or single level { } match from source
~?	Same as ?
~n?	Replace with n-th ? or single level { } match from source
*	Replace with entire corresponding field (name or extension) from source
~*	Replace with corresponding * or two level { } match from source
~n*	Replace with n-th * or two level { } match from source

For example, the command:

RENAME A*X.ZZZ *.XXX

would rename all files which matched A*X.ZZZ to have the same name but the extension XXX. The command:

RENAME A*X.ZZZ Z~*Q.XXX

would also change the names of the matching files. ABCX.ZZZ would be renamed to ZBCQ.XXX.

This is a super set of the DOS handling of destination wildcard file specifications. It adds a powerful matched sub-string replacement capability to the simple name and extension replacement provided by DOS. Assuming that the ~ character is not used, all valid DOS destination wildcard file specifications will work as expected under XOS.

Logical Names

XOS also supports a system of logical names. A logical name has the same format as a device name and can be used any place that a device name can be used. It is an arbitrary name which is defined to be equivalent to another device name, optionally followed by a string of characters (which normally represents a directory path). Logical name definitions can be nested, i.e., the definition of a logical name may be a logical name. This nesting is limited to 6 levels to make detection of definition loops reasonably efficient.

A logical name can be substituted or assigned. An assigned logical name is a simple alternate name for a device. The definition must be a device name only. No directory path is allowed. When an assigned name is used, the defined name is simply substituted for the logical name. If an attempt is made to set a current directory path for an assigned logical name, the current directory path is set for the underlying physical device. A substituted name behaves like a device name in its own right. Its definition can include a directory path specification. More importantly, a current directory path can be associated with the name, just as if it were a physical disk name. When a substituted logical name is expanded, the current directory path associated with the logical name is appended to the definition of the logical name. The resulting string replaces the logical name in the file specification.

It should be noted that XOS assigned logical names are mostly equivalent to the DOS logical disk names created with the DOS ASSIGN command. XOS substituted logical names are mostly equivalent to the DOS logical disk names created with the DOS SUBST command.

A substituted logical name can be defined to represent a disk on a remote system. This name then behaves exactly like a local disk name. For example, if we have a file C:\SOME\WHERE\FOO.BAZ on the remote system LIZARD::, we can use the following definitions:

LOGICAL/SUB L:=LIZARD::C:
CD L:\SOME\WHERE
TYPE L:FOO.BAZ

Alternately, we could use:

LOGICAL/SUB L:=LIZARD::C:\SOME\
CD L:\WHERE
TYPE L:FOO.BAZ

Two special logical names are built into XOS. The name Z: is used as a default device name when no device name or network address is specified. The definition of this name thus specifies the default disk. The name NET: is used as a default device name when only a network address is specified. Since most XOS configurations will support only one network connection, this name will normally be defined to be the remote file system device using that interface (usually XFP0:). In configurations supporting more than one network connection, it is used to specify which is the default network connection.

Environment Strings

XOS provides a mechanism for associating the definition of various environment strings with a process. An environment string is a simple string of characters associated with a name. Programs executing in the process can retrieve the string by specifying its name. This is similar to the DOS environment capability, but is quite a bit more general in that environment strings can be defined or redefined at any time, without any restrictions on the space required to store the strings. This provides a powerful mechanism for specifying various defaults for programs and for transferring data between programs.

XOS environment strings are maintained by the operating system. The environment string definitions are stored outside of the user process' address space and are accessed using system calls. Each process in the system has a private set of environment definitions which are inherited from the process' parent when the process is created.

Even though each process has its own private set of environment string definitions, a process can change the definitions for other processes (privilege restrictions permitting) as well as it own definitions.

When a DOS program is run under XOS, the environment string definitions are copied into a memory block in the DOS virtual machine, completely emulating DOS's use of environment strings. When a DOS program runs an XOS program as a result of a DOS exec function, the XOS environment strings are initialized as specified in the exec function.

XOS also provides a set of system level environment string definitions, which are not directly associated with any process in the system. These definitions are used to initialize the environment string definitions for processes created by the INIT process. These are generally top level command processors or background programs (often referred to as symbionts).

Normally, the environment strings defined for a session's command processor (the session level process) are considered to be the current set of environment strings for the session. The SETENV command displays and modifies these definitions by default. The DOS compatible SET command always modifies these definitions.

Environment strings are typically used for a number of different purposes. The major one is to specify permanent defaults for various programs. Since environment string names are global to a session, there can be a problem of name conflicts if defaults are to be defined for many different programs. XOS has established a convention for naming the environment strings used to specify defaults in order to prevent such conflicts. The use of this convention is optional, but it is highly recommended. All environment string names which follow this convention begin with the ^ character and consist of three sections, separated by ^ characters as follows:

^VENDOR^PROGRAM^USAGE

where VENDOR is the name of the vendor of the program which uses the environment string. All environment strings used by XOS utilities use the vendor name XOS. PROGRAM is the name of the program and USAGE is a descriptive name for the individual environment string. For example, the XOS COPY command uses an environment string ^FTE^COPY^OPT to specify default command line options. Most XOS commands use the environment string ^FTE^GCP to specify certain global options (such as the level of DOS compatibility) which are of interest to all programs.

Programmers planning to distribute XOS programs using this convention should contact XOS Systems to register a unique vendor name. See Chapter 12 for contact information.

Note that while it does not follow this convention, the XOS command processor does use the PROMPT environment string to specify the format of the command prompt for compatibility with DOS.

XOS does not use an environment string to specify the directories to search when loading programs as does DOS (the PATH environment string). Instead, it uses the logical name CMD: (which is generally defined as a search list logical). The PATH and SET commands and the DOS environment segment created when a DOS program is loaded use the CMD: logical definition to generate a DOS PATH environment string for compatibility.

Chapter 3

Command Reference

The commands described in this chapter are available on all standard XOS systems. A command consists of a command name, optionally followed by command line arguments and/or options, and is entered at the XOS command prompt.

Options are preceded by a slash (/) or dash (-) character, and the value for the option (if any) is placed after an equal (=) sign. If the value needs to contain a space, tab, slash (/), quote (“), comma(,), dash (-), or equal (=), it must be placed in “quotes”.

Arguments are any items following the command name which do not begin with a slash or dash character. Arguments can also be assigned a value by following the argument with an equal sign followed by the value.

The valid options, arguments, and values depend on the command.

For example:

SETENV ABC = “/OPTION=4, /LINES=50"

where the environment variable ABC is set to the string specified in quotes. Here, the quotes are necessary because of the presence of the slash, comma, and space characters in the string.

LOGICAL XYZ: = “C:,D:”

where the logical name XYZ is assigned to represent the list of devices specified in quotes. Here, the quotes are needed because of the comma character.

Unless otherwise stated, options may be placed anywhere on the command line. For example, a valid command to set up the XOS system accounting directory would be:

```
LOGICAL /SYS/QUIET ACT: = _D0P1:\XOS\ACT\
```

Here, LOGICAL is the command keyword, /SYS and /QUIET are options, ACT: is an argument and _D0P1:\XOS\ACT\ is the value for the argument.

Common Options

A number of system commands use common option definitions. These options are explained below. Parts of the options are shown in {braces}; the command can be entered as the entire word (omitting the braces) or as the abbreviation appearing outside the braces. Nested braces are used to indicate multiple abbreviations.

For example, /P{AU{SE}} means that /P, /PAU, and /PAUSE are all valid ways of entering the /PAUSE option.

The abbreviation {/option} is used in the command syntax section of the XOS commands which follow. This should be taken as meaning {/option {/option {...}}}. That is, that options are optional, and more than one can be used if they are not mutually incompatible. Also, unless otherwise noted, these options can appear anywhere on the command line, even intermixed with the command arguments.

When an option begins with a slash character (/), it can be placed immediately after the preceding element on the command line. When it begins with a dash character (-) it must be preceded with a space character ( ). This is necessary since a dash is a valid character in file specifications and other command line elements. It should be noted that the slash character is the preferred character for indicating an option. The dash character is accepted for compatibility with DOS.

Some of the common options are listed below.

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ONFIRM}

This option is used with commands that delete, move, or copy files or directories. It specifies that the system will stop and ask the user to confirm the action which is about to be performed for each file or directory.

/NOC{ONFIRM}

This option is used with commands that delete, move, or copy files or directories. It specifies that the system will not stop and ask the user to confirm the action which is about to be performed for each file or directory.

/Q{UI{ET}}

This option takes no arguments. It specifies that the command should not report its status, any informational messages, or successful completion. This does not affect reporting of errors. This option is often used in batch files to reduce unnecessary screen output.

/NOQ{UI{ET}}

This option takes no arguments. It specifies that the command should report its status, informational messages, and successful completion. This is the standard default for all commands but it may have to be specified to override a /QUIET option specified as an initial option for a command.

/V{ER{BOSE}}

This option takes no arguments. It specifies that the command should report full detail on successful completion. It will also cause some commands to provide additional information when errors are reported. It may also cause more detailed error messages to be generated. Note that /VERBOSE is not the opposite of /QUIET. These are independent options which generally affect the reporting of different information, or of the same information in different ways, depending on the command.

/NOV{ER{BOSE}}

This option takes no arguments. It specifies that the command should not report full detail during command execution or upon completion. This option will not suppress error reporting but will cause less detailed error messages to be generated.

DOS Compatible Commands

All commands which are DOS compatible accept arguments which start with a non-alpha character without any space between the command and the first argument. For example, CD\MISC (no space) is a valid command, as is CD \MISC (with a space). Commands which are unique to XOS always require a space between the command name and the first argument. If an option which begins with a slash immediately follows the command name, a space is never required.

Some commands are provided only for compatibility with DOS programs or batch files which may call them. These commands have only the features found in the DOS commands by the same name; they are clearly documented, along with their XOS equivalent, if any. For example, the SET command will work exactly as under DOS (quirks and all.) The XOS command SETENV replaces this command with a more comprehensive and logical command structure.

There is a program called COMMAND.COM packaged with the XOS system. It is provided solely for those DOS programs that attempt to run COMMAND (via the COMSPEC environment variable or otherwise). It is a simple DOS COM file which loads the XOS command processor, SHELL. DOS programs can directly load SHELL, but some expect the name to be COMMAND.COM, the value of the COMSPEC environment variable notwithstanding.

ADDUNIT

Syntax:

ADDUNIT {/options} class{:} unit = num {item1 = value1 {item2 = value2 {...}}}

Purpose:

Adds new device units.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Initial option values are set from the ^XOS^ADDUNIT^OPT environment string. If no options are specified, the defaults are:

/

Description:

This command is used to add physical hardware units to a device class. This command creates a device unit associated with the specified device class and associates it with the specified hardware.

THIS INFORMATION WILL BE PROVIDED LATER.

Example:

ALIAS

Syntax:

ALIAS {/options} {keyword{={value}}}

Purpose:

Creates, changes, and deletes command aliases.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Initial option values are set from the ^XOS^ALIAS^OPT environment string. If no options are specified, the defaults are:

/

Description:

The ALIAS command allows certain keywords to be used in the place of others, usually to shorten or simplify commands. A keyword can be a command, value, or argument to a command or any combination of letters and/or numbers. It is possible to alias most XOS commands (except ALIAS). This command should be used with caution.

Typing ALIAS without any arguments will list all currently active aliases.

Typing ALIAS *keyword* will list the alias (if any) for the given keyword.

Typing ALIAS *keyword*= will remove the alias for the given keyword.

Typing ALIAS *keyword=@Filename{.BAT}* will set the keyword to the command or commands in the batch file specified. Thereafter, typing the keyword will be the equivalent of running the batch file, but without having to access the disk. If the extension is omitted, .BAT is assumed.

Finally, typing ALIAS *keyword=value* will set the keyword to the value. The keyword will be displayed on the command line, but the value will be used instead when the command is interpreted by the SHELL.

Example:

Alias the keyword DL to mean DIR /LONG /PAUSE /SORT=FILE:

C:>ALIAS DL=DIR /LONG /PAUSE /SORT=FILE

Remove the alias DL:

C:>ALIAS DL=

List all active aliases:

C:>ALIAS

ALIB

Syntax:

Purpose:

Options:

Default:

Example:

ALINK

Syntax:

Purpose:

Options:

Default:

Example:

AMAC

Syntax:

Purpose:

Options:

Default:

Example:

AMAKE

Syntax:

Purpose:

Options:

Default:

Example:

ATTRIB

Syntax:

ATTRIB {/options} {+/-S} {+/- H} {+/- R} {+/- A} filename

Purpose:

Displays or changes file attributes.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for any changes to be made to a file attribute.

/NOC{ON{FIRM}}

This option causes the system not to request confirmation for each change to be made to a file attribute.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/SOR{T} = value

This option specifies how (or if) the list of files displayed is to be sorted. Valid values are ASC{ENDING}, REV{ERSE}, FIL{E}, EXT{ENSION}, DAT{E}, SIZ{E}, and NON{E}.

/NOSORT

This option specifies that the list of files should not be sorted. This is the same as /SORT=NONE.

Default Options:

Initial option values are set from the ^XOS^ATTRIB^OPT environment string. If no options are specified, the defaults are:

/NOQUIET /NOCONFIRM /SORT=FILE

Description:

ATTRIB is a DOS style command with some XOS extensions. It is used to look at or change any of the file attributes which are part of the directory entry for each file. Files may be specified using wildcard or ellipsis specifications.

The attributes determine the following parameters and can be set ON by using + or OFF by using -. Specifying none of the possible parameters will display the attribute bits for the selected filename(s).

A:

Archive bit. When set, indicates that the file has been modified since it was last backed up.

H:

Hidden file. Will not display in a directory while this bit is set.

R:

Read only file. Cannot be modified while this bit is set.

S:

System file. Used for system only files.

Example:

To display the attribute bits on all files on C: root directory:

C:>ATTRIB C:*.*

A—- TESTA.FIL
-H— TESTB.FIL
A-R- TESTC.FIL
AHRS TESTD.FIL
—— TESTE.FIL
—R- TESTF.FIL

Then, to set TESTE.FIL to Read-only:

C:>ATTRIB +R TESTE.FIL

—R- TESTE.FIL

To set the archive bit on all files on D:, sorting output by file name:

C:>ATTRIB +A D:\...*.* /SORT=FILE

<A long alphabetized list of files will be output>

BINCOM

Syntax:

Purpose:

Options:

Default:

Example:

CHDIR

Syntax:

CHDIR {options} {{device:}path} OR

CD {/options} {{device:}path}

Purpose:

Displays or sets the current directory.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

Default Options:

There are no initial or default options for CHDIR.

Description:

This command displays or sets the current directory.

If no argument is given, the current directory for the default disk is displayed.

If only the device (which must be a disk or a substituted logical name which maps to a disk) is given, the current directory for that disk is displayed.

If only a path is given, the current directory for the default disk is set to be that path.

If both a device and path are given, the current directory for the disk specified is set to the path given.

Example:

To display the current directory:

C:>CD <ENTER> OR C:>CHDIR <ENTER>

C:\DATA\MISC

To display the current directory for drive A:

C:>CD A:

A:\

To set the current directory on drive A: to \XYZ

C:>CHDIR A:\XYZ

Note:

For MS-DOS compatibility, CD does not require a space between the command and the argument when the argument begins with a backslash character. For example, CD\TEMP does not require a space, but CD C:\TEMP does. It is recommended that a space be used at all times.

CHKDSK

Syntax:

CHKDSK {/options} {device:}

Purpose:

Tests the integrity of the files and disk structure, and optionally corrects problems that are encountered.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/A{UTO}

This option activates AUTOMATIC mode which causes CHKDSK to automatically correct any problems or errors that are found. Confirmation of corrections will be displayed on the screen.

/D{ISPLAY}

The /D{ISPLAY} option will cause additional informational messages to be displayed at various points in the disk processing.

/F{IX}

This option is the same as /AUTO and is provided for DOS compatibility.

/S

The /S option will alter the way CHKDSK displays its information. By default, a box will appear on the screen containing information; using the /S option will cause all information to be displayed in a linear fashion suitable for use with non-ANSI terminals, redirection to output files, or other non-display devices such as printers.

/T{ABLE}=n

The /T{ABLE}=n option will force CHKDSK to use one specified copy of the file allocation table (FAT) for comparison with the disk data. This option is only recommended for advanced users.

/V{ERIFY}

This option causes CHKDSK to display all filenames and paths as they are processed. In this mode, CHKDSK will look like the DOS equivalent command.

/X

This option allows the user to correct problems manually as they are encountered. This option is only recommended for advanced users.

Default Options:

There are no initial or default options for CHKDSK.

Description:

CHKDSK is a disk data diagnostic and repair tool. It can be used to check files (programs and data) for a variety of technical problems, such as FAT errors, lost chains, and directory problems. CHKDSK will not check the disk for physical defects of any type, including those causing read and write errors. If it is suspected that such errors exist, they should be corrected BEFORE using CHKDSK. Otherwise, CHKDSK will abort with an error message. Also, if there are hardware errors, CHKDSK may read invalid information from the FAT and/or directories and repair errors which don't exist, destroying valid data.

CHKDSK should be used, for example, when directory entries contain obviously incorrect information (random or spurious characters in the file name, or the date or time, etc.) or when a program will not run, or a program cannot access its configuration or data files.

Note:

CHKDSK will not make any changes to the disk unless the /A, /F, or /X options are specified. This behavior is the same as the DOS equivalent command.

Example:

This command will automatically repair any errors that are found on the current drive:

C:>CHKDSK /A

This command will display the maximum amount of information for drive C:, but will not make changes to the disk:

```
C:>CHKDSK /V /D C:
```

CLS

Syntax:

CLS

Purpose:

Clears the display and returns the cursor to the upper left corner of the screen.

Options:

None.

Default Options:

None.

Description:

The CLS command clears the screen to the currently set foreground and background colors (see COLOR).

Example:

REM Programs output to the screen here...

...

REM Now, clear all that off.

CLS

CLSCHAR

Syntax:

CLSCHAR {/options} class{:} {char1=value1 {char2=value2 {...}}}

Purpose:

To display and modify characteristics of device classes.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option specifies that a short text description should be displayed for each characteristic displayed.

/NOVER{BOSE}

This option specifies that a short text description should not be displayed for each characteristic displayed.

Default Options:

Initial option values are set from the ^XOS^CLSCHAR^OPT environment string. If no options are specified, the defaults are:

/NOQUIET /NOVERBOSE

Description:

This command is used to modify and display class characteristics for any device class in the system. Refer to chapter 5 for a description of the device classes and the class characteristics.

The first argument specifies the device class. It may optionally be terminated with a colon. This argument is required and may not have a value. Each additional argument specifies a characteristic. If no characteristics are specified, the current values of all characteristics for the device class are displayed. If one or more characteristics are specified with values, the characteristics are set to the values specified. The values of all characteristics specified (with or without a new value being specified) are displayed (unless /QUIET was specified).

Example:

To add more disk cache data buffers:

```
C:>CLSCHAR DISK NUMDBUF=150

  NUMDBUF {Number of disk data buffers) = 150
```

Note that the CLSCHAR SYSTEM: and SYSCHAR commands are equivalent.

COLOR

Syntax:

COLOR {/options} {arg1=value1 {arg2=value2 {...}}}

Purpose:

Sets the default display colors

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/G{RAPHICS}

Specifies that the graphics mode colors will be changed.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/T{EXT}

Specifies that the text mode colors will be changed.

Default Options:

Initial option values are set from the ^XOS^COLOR^OPT environment string. If no options are specified, the defaults are:

/TEXT

Description:

This command sets the default console display colors for text and graphics modes. Foreground, background, fill colors, and display attributes can be set independently. The default colors are used when displaying text to the screen using normal text terminal output functions. They have no effect when writing directly to the screen buffer or using the graphics functions.

If the command is given with no arguments, the current default colors are displayed. If the /TEXT or /GRAPHICS options is specified, the indicated set of default colors are set to the values given. If the attribute value is not given, the current attribute value is not changed. If the fill colors are not given, they default to the corresponding foreground and background colors. If the background color is not specified, a value of 0, or black, is used.

Parameters:

The valid parameters are listed in the table below:

Parameter	**Meaning**
FGC	Foreground color
BGC	Background color
FGF	Foreground fill color
BGF	Background fill color
ATR	Attributes

The values for the FGC, BGC, FGF, and BGF arguments may be specified either as a keyword or a number as follows:

Keyword	**Number**	**Meaning**
BLA{CK}	0	Black
BLU{E}	1	Blue
GRE[EN]	2	Green
CYA{N}	3	Cyan
RED	4	Red
VIO{LET}	5	Violet
BRO{WN}	6	Brown
WHI{TE}	7	White
GRA{Y}	8	Gray

Keyword	Number	Meaning
BBL{UE}	9	Bright blue
BGR{EEN}	10	Bright green
BCY{AN}	11	Bright cyan
BRE{D}	12	Bright red
BVI{OLET}	13	Bright violet
YEL{LOW}	14	Yellow
BWH{ITE}	15	Bright white

These values assume that the standard palette and DAC (digital-analog converter) values are used.

The following values are valid for the ATR argument. Each value is valid only for the mode indicated.

Value	Mode	Meaning
BLI{NK}	Text	Enable blinking
NOB{LI{NK}}	Text	Disable blinking
UND{ERLINE}	Text	Enable underlining
NOU{ND{ERLINE}}	Text	Disable underlining
XOR	Graphics	Enable XOR write mode
NOX{OR}	Graphics	Disable XOR write mode

Examples:

To set the foreground to red and the background to white:

C:>COLOR FGC=RED BGC=WHITE

To display the current color values:

C:>COLOR

Graphics base color values:

FGC=White(7) BGC=Black(0) FGF=White(7) BGF=Black(0)
ATR=Noxor

Text base color values:

FGC=Green(2) BGC=Black(0) FGF=Green(2) BGF=Black(0)
ATR=Nounderline,blink

CONFIG

Syntax:

Purpose:

Options:

Default:

Example:

COPY

Syntax:

COPY {/options} source destination

Purpose:

Copies files.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/A{SCII}

This option causes copy to stop copying for the source file when a ^Z (control-Z) character is encountered. This option only needs to be specified when copying files created using the DOS version 1 FCB functions which did not keep track of the exact length of files and multiple source files are being combined into a single destination file.

/B{INARY}

This option causes COPY to copy all bytes in the file irrespective of the contents of the file. Control Z is not considered to be a special character.

/BUF{FER}=size

This option allows the specification of the maximum size of buffer that will be used for the copy, in Kbytes.

/C{ON{FIRM}}

This option causes the system to request confirmation for each file to be copied.

/NOC{ON{FIRM}}

This option causes the system not to request confirmation for each file to be copied.

/D{EL{ETE}}

This option causes incomplete output (destination) files to be deleted. This behavior is DOS compatible..

/NOD{EL{ETE}}

This option causes incomplete output (destination) files not to be deleted. This behavior is **not** DOS compatible.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/SOR{T} = value

This option specifies how (or if) the files to be copied as the result of a wildcard specification are to be sorted before the actual copy is done. Valid values are ASC{ENDING}, REV{ERSE}, FIL{E}, EXT{ENSION}, DAT{E}, SIZ{E}, and NON{E}.

/NOSORT

This option specifies that the files to be copied as the result of a wildcard specification should not be sorted before the actual copy is done. This is the same as /SORT=NONE.

/STA{RT}=filename

This option allows the COPY to begin with the specified filename and proceed normally from that point to the end of the copy operation. This allows an interrupted multiple file copy to be restarted from the point where it was interrupted. It is important that the /SORT option be set as desired, since the order of the source files will determine the list of files to be copied.

/T{OT{ALS}}=value

This option causes the display of totals for the copy run. The value can be:

BYTES

The total number of bytes copied will be displayed

NOBYTES

The total number of bytes will not be displayed..

FILES

The total number of files copied will be displayed.

NOFILES

The total number of files copied will not be displayed.

RATE

The transfer rate (bytes/second) will be displayed.

NORATE

The transfer rate will not be displayed..

If no value is specified, all of the above totals are displayed.

/NOT{OT{ALS}}

This specifies that no totals should be displayed.

/VER{BOSE}

This option specifies that the source and destination file specifications should be displayed as each file is copied.

/NOVER{BOSE}

This option specifies that the source and destination file specifications should not be displayed as each file is copied.

/V{ER{IFY}}

This option specifies that the source and target files will be compared after the copy to insure an error free copy.

/NOV{ERIFY}

This option specifies that the source and target files will not be compared after the copy.

Default Options:

Initial option values are set from the ^XOS^COPY^OPT environment string. If no options are specified, the defaults are:

/NOQUIET /NOCONFIRM /BINARY /BUFFER=16 /SORT=NAME /TOTALS=FILES /NOVERBOSE

Description:

The source file is copied to the destination file. Multiple files can be copied in a single pass. Wildcard characters are allowed in both the source and destination specifications. The ellipsis notation is allowed in the final directory of the source and target paths. It indicates that all directories below the last specified directory are to be included in the command. If the destination specification is not given, *.* is assumed. Wildcard characters in the destination specification must match those in the source specification. For example, ABC*.* and DEF*.* are compatible wildcards, while \ABC.* and \GHI*.XXX are not.

If the ellipsis notation is used in the source specification and not in the destination specification, all files are copied into the single destination directory specified. For example,

COPY A:\...*.* C:\TEMP

will copy all files in all directories of A: into \TEMP on C:.

Conversely, if an ellipsis is used in the destination and not in the source, it will have no effect since only one directory is being copied.

If the ellipsis notation is used in both the source and destination specification, the source directory structure is duplicated for the destination. Any required directories which do not exist are created.

Note:

This is the only instance in which the COPY command will create a directory. Otherwise, if the directory does not exist, the copy will fail.

For example,

COPY C:\MISC\...*.TXT D:\TEMP\...*.TXT

This command will copy all files with the extension .TXT in C:\MISC and all its subdirectories into D:\TEMP and subdirectories of the same name on D:. If these subdirectories do not exist, they will be created.

Example:

To copy all files with a .DIS extension and give the copies a .DAT extension, using a 12,000 byte copy buffer:

C:>COPY /BUFFER=12000 \XYZ*.DIS \ABC*.DAT

nnn files copied

COUNT

Syntax:

Purpose:

Options:

Default:

Example:

CRSHSAVE

Syntax:

Purpose:

Options:

Default:

Example:

DATE

Syntax:

DATE

Purpose:

Displays and changes the current system date.

Options:

None.

Default Options:

None.

Description:

DATE is a DOS compatible command which does not conform to the standard XOS command syntax. It displays the system date, and then asks for a new date.

If a new date is specified, the system date is set to the date specified. If a date is not specified (ENTER only), the system date is not changed.

This command is provided only to provide DOS compatibility. The DAYTIME command is the preferred XOS command.

Example:

To display or change the current date:

C:>DATE

Current date is Monday 01-Jan-90

Enter new date (mm-dd-yy):

DAYTIME

Syntax:

DAYTIME {/options} {DD/MMM/YY} {HH:MM{:SS}}

Purpose:

Displays or sets the current date and/or time.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/RTC

Specifies that the date and time value should be obtained from the real time (CMOS) clock.

{NORTC

Specifies that the date and time value should not be obtained from the real time (CMOS) clock.

/SYS

Specifies that the date and time value should be obtained from the system.

{NOSYS

Specifies that the date and time value should not be obtained from the system.

/SETRTC

Specifies that the real time (CMOS) clock should be set.

/SETSYS

Specifies that the system date and time should be set.

/12

Specifies that the time should be displayed in 12 hour format.

/24

Specifies that the time should be displayed in 24 hour format.

Default Options:

Initial option values are set from the ^XOS^DAYTIME^OPT environment string. If no options are specified, the defaults are:

/SYS/24

Description:

This command displays or sets the system or real time clock date and time. If the /SETSYS and/or /SETRTC options are specified, the corresponding date and/or time value is set. The source of the value can be the system date and time (/SYS specified), the real time clock date and time (/RTC specified) or a date and/or time value entered as a command argument. If only the date is entered, only the date is changed. If only the time is entered, only the time is changed.

If /SETSYS or /SETRTC is specified, one (and only one) of /SYS, /RTC, or a date/time value must be specified. The user must have the SYSADMIN privilege to change either the system or real time value. The dates value must be specified in the form *dd-mmm-yy* where *dd* is the day of the month, *mmm* is the three letter abbreviation for or name of the month, and *yy* is the year (2 or 4 digits). The time value must be specified as *hh:mm* or *hh:mm:ss*, where *hh* is hours (24 hour notation), *mm* is minutes, and *ss* is seconds.

Example:

To display the current date and time:

C:>DAYTIME

DAYTIME: System time and date is 22:35:56 on Monday, 9-Mar-92

To set the system time to 2:00pm:

C:>DAYTIME /SETSYS 14:00

Monday 14:00:00 01-Jan-90

DEFAULT

Syntax:

DEFAULT {/options} CMDNAME default_options

Purpose:

Allows the editing of environment strings.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/D{ELETE}

This option removes the default command line.

/EDIT

If this option is specified, no default_options should be specified following CMDNAME. In this case, the existing contents of the environment string will be displayed as if it had been entered from the keyboard, It can then be edited using the normal keyboard input editing keys.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/S{YSTEM}

This option specifies that system level defaults are to be used for the given command. The system level defaults are used if no private default is set.

Description:

DEFAULT provides a convenient method for setting or modifying the environment strings which specify the default options for most XOS commands.

Options specified after the CMDNAME argument are copied to the environment string ^XOS^PROGNAME^OPT which will be used whenever the command PROGNAME is issued to set its initial options. Options specified before the CMDNAME argument apply to this command itself, as described above.

If GLOBAL is specified as the CMDNAME argument, special global default options are set which apply to all commands. The valid global options are listed below.

/DOSQUIRK

Valid values are ON and OFF. A value of ON indicates that all commands should exactly emulate the corresponding DOS command behavior. A value of OFF indicates that all commands should not exactly emulate DOS commands but can implement various XOS extensions.

/DOSDRIVE

Valid values are ON and OFF. A value of ON indicates that disk drive names should be reported using the DOS drive letter (A:, C:, etc.) whenever possible. A value of OFF indicates that disk drive names should be reported using the XOS disk name (F0, D0P1, etc.).

Example:

C:>DEFAULT DIR /W

This will set the default directory display to a wide, multi-column display.

C:>DEFAULT GLOBAL /DOSQUIRK=ON

This will set all commands to default to exact DOS emulation.

DELETE

Syntax:

DELETE{ /options} file1 {file2 {...}}

DEL {/options} file1 {file2 {...}}

See also ERASE.

Purpose:

Deletes files.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for each file to be deleted.

/NOC{ON{FIRM}}

This option causes the system to not request confirmation for each file to be deleted.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/SOR{T} = value

This option specifies how (or if) the files to be copied as the result of a wildcard specification are to be sorted before the actual copy is done. Valid values are ASC{ENDING}, REV{ERSE}, FIL{E}, EXT{ENSION}, DAT{E}, SIZ{E}, and NON{E}.

/NOSORT

This option specifies that the files to be copied as the result of a wildcard specification should not be sorted before the actual copy is done. This is the same as /SORT=NONE.

Description:

The file(s) specified are deleted. Once they are removed, they can no longer be accessed, and the space they occupied on the disk will be reused by XOS. Wildcard and ellipsis characters are allowed. DEL can accept more than one filename or wildcard per command line.

Note:

The response from DEL will vary with the state of the DOSQUIRK option. When DOSQUIRK is ON, no response will be generated. When DOSQUIRK is OFF, a list of files will be generated according to the specified command line options.

Example:

To delete the file COMM.DAT in the directory \TERM:

C:>DEL \TERM\COMM.DAT

To deletes all files with the .BAK extension in the current directory and all its subdirectories:

C:>DELETE ...*.BAK

DEVCHAR

Syntax:

DEVCHAR {/options} device{:} {char1=value1 {char2=value2 {...}}}

Purpose:

To display and modify device characteristics.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option specifies that a short text description should be displayed for each characteristic displayed.

/NOVER{BOSE}

This option specifies that a short text description should not be displayed for each characteristic displayed.

Default Options:

Initial option values are set from the ^XOS^DEVCHAR^OPT environment string. If no options are specified, the defaults are:

Description:

This command is used to modify and display device characteristics for any device unit in the system. Refer to chapter 6 for a description of the devices and the device characteristics.

The first argument specifies the device. It may optionally be terminated with a colon. This argument is required and may not have a value. Each additional argument specifies a characteristic. If no characteristics are specified, the current values of all characteristics for the device unit are displayed. If one or more characteristics are specified with values, the characteristics are set to the values specified. The values of all characteristics specified (with or without a new value being specified) are displayed (unless /QUIET was specified).

Example:

To set the basic characteristics for the first four sessions on the console:

C:>DEVCHAR TRM0S1: IFLOW=ON SESSION=YES FLOW=XON

C:>DEVCHAR TRM0S2: IFLOW=ON SESSION=YES

C:>DEVCHAR TRM0S3: IFLOW=ON SESSION=YES

C:>DEVCHAR TRM0S4: IFLOW=ON SESSION=YES

DIR

Syntax:

DIR {/options} {filespec1 {filespec2 {...}}}

Purpose:

Produces a directory listing.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/ALL

This option causes all files to be listed, including hidden and system files.

/DIR

This option causes only subdirectories to be listed.

/DISPLAY = value

This option specifies the format of the display produced. Valid values are:

ACROSS

The names are ordered in rows.

DOWN

The names are ordered in columns.

{NO}PATH

The path name is (or is not) included in the directory heading.

{NO}TOTALS

The totals are (or are not) displayed after the directory listing.

{NO}VOLUME

The volume label is (or is not) displayed in the directory heading.

/FULL

This option produces a detailed directory listing. There is no corresponding directory listing format.

/LONG

This option produces a long directory listing. This is equivalent to the default DOS directory format. The exact format displayed depends on the setting of the DOSQUIRK option.

/NAMES

This option includes filenames in the directory listing.

/NONAMES

This option excludes filenames from the directory listing.

/ONE

This option produces a one column directory listing.

/P{AUSE}

This option causes a pause for each screen full of output.

/NOP{AUSE}

This option does not pause for each screen full of output.

/SHORT

This option produces a short directory listing. This is equivalent to the DOS /W directory format. The exact format displayed depends on the setting of the DOSQUIRK option.

/SORT = value

This option specifies the order for directory listing as follows:

ASC{ENDING}

Sort in increasing order.

DAT{E}

Sort by file creation date and time.

EXT{ENSION}

Sort by file extension then by filename.

FIL{E}

Sort by filename then by file extension.

NON{E}

Do not sort. Display the files in directory order.

R{EV{ERSE}}

Sort in descending order.

SIZ{E}

Sort by file size.

S{UB{DIR}}

List subdirectories then files.

/TOTALS

This option displays totals only.

/W{IDE}

This option is the same as /SHORT. It is included for DOS compatibility.

Default Options:

Description:

This command produces a directory listing of the files on a specified disk drive. The file specification given determines which files are included in the listing. Wildcard and ellipsis notation are allowed in a file specification. If no file specification is given, *.* is assumed. More than one file and/or wildcard may be specified on the command line.

Many options are possible to the directory listing. It is recommended that the user select a preferred directory format using command line options and enter it as the default in the USTARTUP.BAT file using the DEFAULT command.

Example:

To list all files in the current directory:

C:>DIR

To produce a long listing of all files in the current directory which have a .DAT extension:

C:>DIR /L *.DAT

To produce a listing of all files on the disk D:, pausing between screens, and sorting by extension:

C:>DIR D:\...\ /SOR EXTENSION /P

DISMOUNT

Syntax:

DISMOUNT {/options} disk:

Purpose:

Removes a disk from the system.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

Default Options:

Description:

DISMOUNT removes a disk from the system. All information about the disk is discarded. If there are any unwritten blocks in the disk cache for the disk, they are discarded. This command is normally not needed. It should never be needed for fixed (non-removable) disks. It can be used with removable disk to insure that data does not remain in memory after a disk has been changed. The system normally does this automatically.

Obviously, this command should be used with care. It should not be issued when any output files are open on the disk unless data loss can be tolerated.

Note:

There is no corresponding MOUNT command. Disks are mounted automatically whenever they are accessed.

Example:

To dismount the A: floppy disk:

C:>DISMOUNT A:

DISPLAY

Syntax:

DISPLAY {/options} {arg1=value1 {arg2=value2 {...}}

Purpose:

Displays or changes the console display mode.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Parameters:

The overall mode of the display may be specified with one of the following parameters:

TEXT

All text modes.

MONO

MDA graphics modes.

CGA2

2 color CGA graphics modes.

CGA4

4 color CGA graphics modes.

EGAM

EGA/VGA mono graphics modes.

EGA4

4 color EGA graphics modes.

EGA16

16 color EGA graphics modes.

VGA16

16 color VGA graphics modes.

VGA256

256 color VGA graphics modes.

The mode may be modified by specifying one or more of the following items:

ADA{PTER} = value Display adapter type.

The value may be either MON{O} or COL{OR} to specify either a monochrome or color style adapter. Note that EGA and VGA display adapters can be configured as either color or monochrome adapters. MDA display adapters can only be configured as monochrome adapters.

BLI{NK} = value Blink feature enable.

The value may be either ON or OFF to enable or disable the blink feature in text mode.

COL{UMNS} = value Number of text columns.

This specifies the desired number of text columns. The system selects the next highest value which is supported by the mode specified.

HOR{IZ} = value Number of horizontal pixels.

This specifies the desired number of horizontal pixels. In text mode this value is used in determining the underlying resolution for the display considering the COLUMNS values specified. In graphics modes it directly specifies the desired resolution.

NOCLEAR

When present, this parameter specifies that the display should not be cleared, even when switching between text and graphics modes.

NOLOAD

When present, this parameter specifies that the display adapter's character generator should not be loaded, even if the new display mode requires a different character set.

ROW{S} = value Number of text rows.

This specifies the desired number of text rows. The system selects the next higher value which is supported by the mode specified. For example, specifying 55 rows on an EGA will display 43 rows.

PAL{ETTE} = value Display adapter palette type.

The value may be either MON{O} or COL{OR} to specify either a monochrome or color palette. Note that EGA and VGA display adapters can be configured with either a color or monochrome palette. MDA display adapters can only have a monochrome palette.

UND{ERLINE} = value Underline feature enable.

The value may be either ON or OFF to enable or disable the underline feature in text mode.

VER{T} = value Number of vertical pixels.

This specifies the desired number of vertical pixels. In text mode this value is used in determining the underlying resolution for the display considering the ROWS values specified. In graphics modes it directly specifies the desired resolution.

Finally, a BIOS mode value can be specified. This will cause the display to be set up to match the BIOS display specified.

BIOS = value BIOS mode specification.

Specifying a value for the BIOS argument sets the basic mode and also sets the modifier values as shown in the table below. If a BIOS value is specified, no other argument should be specified.

BIOS value	Mode	ADAPTER/ PALETTE	COL	ROW	VERT	HORI Z
0x0, 0x1	TEXT	COLOR	40	25	Note 1	Note 2
0x2, 0x3	TEXT	COLOR	80	25	Note 3	Note 4
0x4, 0x5	CGA4	COLOR	40	25	320	200
0x6	CGA2	COLOR	80	25	640	200
0x7	TEXT	MONO	80	25	720	350
0xD	EGA16	COLOR	40	25	320	200
0xE	EGA16	COLOR	80	25	640	200
0xF	MONO	MONO	80	25	640	350
0x10	EGA16	COLOR	80	25	640	350
0x11	MONO	MONO	80	30	640	480
0x12	EGA16	COLOR	80	30	640	480
0x13	VGA256	COLOR	40	25	320	200

For all BIOS modes, the BLINK feature is ON and the UNDERLINE feature is OFF, except for mode 0x7 where the UNDERLINE feature is ON.

Description:

This command displays or changes the mode of the console display.

Examples:

To display the current display mode:

```
C:>DISPLAY
```

```
Mode = TEXT BIOS = 0x3
Adapter=Color Palette=Color Blink=On  Underline=Off
Rows =25    Columns=80    Vert=400 Horiz    =720
```

To set to 50 row text mode:

```
C:>DISPLAY ROWS=50
```

```
Mode = TEXT BIOS = 0x3
Adapter=Color Palette=Color Blink=On  Underline=Off
Rows =50    Columns=80    Vert=400 Horiz    =720
```

DOSCOM

Syntax:

DOSCOM {/options}

Purpose:

Sets up the system to use the standard serial ports as the DOS COM1 and COM2 devices.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command attempts to add terminal class units 1 and 2 (TRM1: and TRM2:) in order to use the standard hardware COM1 and COM2 serial ports. If this is successful, logical device COM1: is defined as TRM1: and COM2: as TRM2:.

Note:

DOSCOM does not set up COM3 and COM4. This cannot be done automatically since there is no standard unique interrupt assigned for either of these serial ports and XOS requires that each standard serial port in the system have a unique interrupt assignment. If COM3 and/or COM4 are to be set up, this must be done explicitly using the ADDUNIT and LOGICAL commands.

This command will usually be included in the startup command file, STARTUP.BAT. It will normally not be used after the system is running, although it can be used at any time, provided that the standard serial ports have not been set up.

Examples:

To set up the standard serial ports:

C:> DOSCOM

DOSDRIVE

Syntax:

DOSDRIVE {options}

Purpose:

Assigns the standard DOS disk drive letters to the disks in the system.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command automatically assigns DOS-style drive letters to the system standard disks. This is limited to the first two floppy disk drives and the first two hard disk drives.

Under DOS, a scheme is used for assigning letters to disks and disk partitions that is potentially somewhat confusing. Drive letters are assigned in the order that DOS is made aware of the disks, starting with C: (A: and B: are always reserved for the floppy drives, even if there are less than two floppy drives present). This can cause problems in systems with two hard disks, since changing the number of partitions on the first hard disk will shift the drive letters assigned to partitions on the second hard disk.

This command assigns DOS drive letters using the same rules as DOS. It is normally executed in the STARTUP.BAT file when the system is initialized, although it can be executed at any time.

Note:

Alternately, the DEVCHAR command can be used to assign fixed DOS drive letters to the system's disks.

Examples:

To assign standard DOS disk letters to the standard disks:

C:> DOSDRIVE

DOSLPT

Syntax:

DOSLPT {/options}

Purpose:

Sets up the standard printer (parallel ports only) devices.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command sets up the first two standard parallel printer devices. Parallel printer setup is somewhat complex since there exists some confusion about which I/O register values correspond to which printer port. This command uses the same algorithm as DOS to assign printer ports. The XOS PPR class units 1 (PPR1:) and 2 (PPR2:) are added if the corresponding physical interfaces exist in the system, and logical names LPT1 and LPT2 are defined as the corresponding XOS names.

DOSLPT is normally executed in the STARTUP.BAT file when the system is initialized, although it can be executed at any time. Note that alternately, the ADDUNIT and LOGICAL commands can be used to explicitly set up the parallel printer ports.

Examples:

To set up the standard parallel printer ports:

```
C:> DOSLPT
```

DUMP

Syntax:

Purpose:

Options:

Default:

Example:

DUMPLOG

Syntax:

Purpose:

Options:

Default:

Example:

ECHO

Syntax:

Purpose:

Options:

Default:

Example:

ERASE

Syntax:

ERASE {/options} file1 {file2 {...}}

See also DELETE and DEL

Purpose:

Deletes files.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for each file to be erased.

/NOC{ON{FIRM}}

This option causes the system to not request confirmation for each file to be erased.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

The file(s) specified are deleted. Once the files are removed, they can no longer be accessed and the space they occupied on the disk will be reused by XOS. Wildcard and ellipsis characters are allowed. ERASE can accept more than one filename or wildcard per command line.

Note:

The response from ERASE will vary with the state of the DOSQUIRK option. When DOSQUIRK is ON, no response will be generated. When DOSQUIRK is OFF, a list of files will be generated according to the specified command line options.

Example:

To delete the file COMM.DAT in the directory \TERM:

C:>ERASE \TERM\COMM.DAT

To deletes all files with the .BAK extension in the current directory and all its subdirectories:

C:>ERASE ...*.BAK

EXE2RUN

Syntax:

Purpose:

Options:

Default:

Example:

FIND

Syntax:

Purpose:

Options:

Default:

Example:

GECKO

Syntax:

Purpose:

Options:

Default:

Example:

GENSYM

Syntax:

Purpose:

Options:

Default:

Example:

GETDSPTP

Syntax:

GETDSPTP

Purpose:

Determines the console display type.

Options:

None

Default Options:

None

Description:

This command determines the console display type and returns a corresponding termination code. This code can be checked with the IF command in a batch file to perform a different action for different display types. It is used in the STARTUP.BAT batch file at system startup time to load the correct console display driver.

HISTORY

Syntax:

HISTORY {/options} {n}

Purpose:

Displays the current command history or sets the size of the command history.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

Default Options:

None.

Description:

The HISTORY command allows the display of command history and the modification of the history buffer size.

Typing HISTORY with no arguments will display the current history size, in number of commands, as well as the commands presently in the history.

Typing HISTORY n, where n is a number between 0 and 65,535, will set the command history to that number of commands. The default is 20 commands.

This is a resident command which is implemented in the command SHELL.

See the Command Recall section in Chapter 2 of this manual for information regarding keyboard commands logged by History.

Examples:

To list the command history:

C:>HISTORY

Command History (5):
5:CLS
4:DIR /P
3:CD \XOS\SYS
2:DIR /LONG
1:HISTORY

To retrieve command #4 in the history:

!4 <ENTER>

In this example, the command DIR /P will be displayed on the command line.

KILLPROC

Syntax:

KILLPROC{/options} processid

Purpose:

Removes a process from the system.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command removes a process from the system. The full process ID (PID) must be specified as SEQ.NUM where SEQ is the process sequence number and ID is the process number. Process IDs can be obtained from the SYSDIS function.

Example:

To remove process 33.2:

C:>KILLPROC 33.2

Process 33.2 killed

LABEL

Syntax:

LABEL {drive:}

Purpose:

Adds, removes, or changes the volume label of a disk.

Options:

None.

Default Options:

None.

Description:

This command is used to change the volume label of a disk. The volume label is displayed at the top of DOS compatible directory listings and is used as information for the user only.

When the command is executed, it will display the current volume label for the specified drive or the current drive if no drive is specified. It will then prompt for the volume label. You can type in a new volume label, or press ENTER to keep the current label or delete the label. The program will ask, Delete current volume label (Y/N)?. Press Y to delete the label or N to keep the same label.

Example:

To see the label for drive c:

C:>LABEL C:

Volume in drive C: is DEVELOPMENT
Enter new volume label:_

At this point, enter the new label or press ENTER to leave the current label unchanged.

LKELOAD

Syntax:

LKELOAD {/options} lkename lke_parameters

Purpose:

Load a Loadable Kernel Extension (LKE) into memory.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Parameters:

Description:

This command loads a loadable kernel extension into memory and gives it a copy of the lke_parameters. If the copy of XOS built with XDT is loaded, the LKE symbols will be added to the debug symbol table.

LOGICAL

Syntax:

LOGICAL {/options} {logicalname: {= {definition}}}

or

LOGICAL /RENAME oldname newname

Purpose:

Lists, defines, renames, or undefines logical device names.

Options:

/H{EL{P}} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/E{DI{T}}

This option retrieves the current definition for the logical name specified so that it can be edited. A single logical name must be specified without an equal sign or definition.

/NOE{DI{T}}

This option negates the /EDIT option. Since this is the normal default condition, this option only needs to be used if the /EDIT option is specified as an explicit default.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/REN{AME}

This option renames the logical name without changing the definition. Two logical names separated by an equal sign must be specified. The old name is specified first (before the equal sign).

/NOREN{AME}

This option negates the /RENAME option. Since this is the normal default condition, this option only needs to be used if the /RENAME option is specified as an explicit default.

/ROO{TED}

This option indicates that the logical name is to be a rooted logical name. See chapter 2, page 14 for a discussion of rooted logical names.

/NOROO{TED}

This option indicates that the logical name is to not be a rooted logical name. See chapter 2, page 14 for a discussion of rooted logical names.

/SUB{ST}

This option indicates that the logical name is to be substituted. See chapter 2, page 14 for a discussion of substituted logical names.

/NOSUB{ST}

This option indicates that the logical name is to not be substituted. See chapter 2, page 14 for a discussion of substituted logical names.

/SYS{TEM}

This option indicates that the logical name is being defined or changed at system level. System level logical names are used to initialize the session level names whenever a new session is created. They are not used otherwise.

/SES{SION}

This option indicates that the logical name is being defined or changed at session level.

Default Options:

Default option values are set from the ^XOS^LOGICAL^OPT environment string. If no options are specified, the defaults are:

/SESSION

If the /RENAME or /EDIT option is in effect, any /SUBST, /NOSUBST, /ROOTED, or /NOROOTED default options are ignored.

Description:

This command provides the means for controlling logical device names. XOS provides a complete system of logical names which are used for many system functions. See chapter 2, page 14 for a complete discussion of logical names.

If no name is specified, all logical names at the indicated level (SESSION or SYSTEM) are displayed. If a logical name is specified without a value (no = after the name), the current value of the logical name is displayed. If a logical name is specified with a value, the value is changed. If a null value is specified (an = after the name but no value after the =), the logical name is deleted.

If a logical name is to be defined as a search list logical, the names in the list are separated with commas. The entire list must be enclosed in double quotes.

If the /EDIT option is specified, a single logical name must be specified without an equal sign or definition. The current definition for that logical name is displayed and can be edited using the normal line editing characters.

If the /RENAME option is specified, two logical names separated by an equal sign must be specified. The first logical name (which must be defined) is changed to be the second logical name specified (which must not be defined).

If the /EDIT or /RENAME option is specified, the /SUBST, /NOSUBST, /ROOTED, or /NOROOTED options specified as defaults are ignored. Any of these options specified on the command line modify the substituted or rooted state of the logical name being edited or renamed. If none of these options are specified with the command, the substituted or rooted state of the logical name is not changed.

Example:

Deletes the logical name TEMP:, reporting results:

C:>LOGICAL /NOQ TEMP:=

Logical name “XXX:” deleted at session level

To display all logical names:

C:>LOGICAL

Matching logical names defined at session level:

CMD: = D0P1:\XOS\CMD

TEMP: = C:\TEMP

To display the definition for the CMD: logical name:

C:>LOGICAL CMD:

Matching logical names defined at session level:

SYS: = Z:,C:\XOS\CMD\,C:\DOSPRG\,D:\MISC\

To specify the definition of a “search list” logical name:

LOGICAL LIST: = “C:\DIR1\,C:\DIR2\”

Logical name “LIST:” defined as “C:\DIR1\,C:\DIR2\ at session level

LPRT

Syntax:

Purpose:

Options:

Default:

Example:

MKBOOT

Syntax:

MKBOOT {/options} device:

Purpose:

Installs the XOS bootstrap on a disk.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/AUTO

This option specifies that the default program should always be loaded. In this case there is no interaction with the user at boot time. Any values specified for /TIMEOUT or /Fn are ignored.

/NOAUTO

This option specifies the user should be asked for the program to load at boot time.

/DEFAULT = value

This option specifies the name of the default program to load. This program is loaded when ENTER is pressed, when the timeout period expires, or if the /AUTO option is specified when the bootstrap is installed.

/Fn = value

This option specifies the programs to be associated with each of the function keys. n may have a value between 1 and 12.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/TIM{EOUT} = value

This option specifies the timeout value for the bootstrap. When a timeout value (in seconds) is specified, the bootstrap will load the default program after the time period has expired with no input from the keyboard. Specifying a timeout value of 0 is equivalent to specifying the /NOTIMEOUT option.

/NOTIM{EOUT}

This option specifies that there is no timeout period when booting.

Default Options:

Initial option values are set from the ^XOS^MKBOOT^OPT environment string. If no options are specified, the defaults are:

/NOQUIET /NOTIMEOUT /NOAUTO /DEF=XOS

Description:

MKBOOT installs the XOS bootstrap on the disk, *device:*.

Each of the function keys, F1-F9, can be set to boot a specific program. Function key F10 always boots DOS. Programs (other than DOS) to be booted must be in a directory which is in the root directory. If no directory is specified, the directory XOS\SYS is used. A program can be booted from any disk on the system, independent of which disk the bootstrap was initially loaded from, except that DOS can only be booted for the disk from which the bootstrap was initially loaded.

Disk names can be specified as either DOS drive letters or XOS disk names.

Example:

To make drive C: (the first hard disk, first partition) bootable using F2 for XOS and F10 for DOS:

MKBOOT D0P1: /F1=C:XOS

MKDIR

Syntax:

MKDIR {/options} dirname   OR

MD {/option} dirname

Purpose:

Creates a directory.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command creates the specified directory. The directory to be created must not already exist but any parent directories or device names specified in the path must exist.

If the directory specified by *dirname* already exists, MKDIR will terminate with an error message stating that the directory already exists.

Example:

To create the directory XYZ in the directory \ABC

C:>MKDIR \ABC\XYZ

MODE

Syntax:

MODE parameters

Purpose:

Redirects printer ports, sets mode for serial ports, and sets display parameters.

Options:

None.

Default Options:

None.

Description:

The MODE command is provided for DOS compatibility. It does not conform to the XOS standard command syntax. It is recommended that this command be used as little as possible due to the limited nature of the options it provides.

MODE has three functional areas, each having its own parameter syntax:

Printer Parameters:

LPTx=COMy redirects LPT port X(1,2, or 3) to COM port Y(1 or 2).

LPTx (no options specified) cancels redirection.

LPTx:width,LPI,timeout flag: (This syntax will undo redirection to COMx). Any of these parameters are optional.

Width: 80 or 132 (characters)

LPI: 6 or 8 (lines per inch)

Timeout: P for infinite retry, and any other character for normal retry.

Serial Port Parameters:

COMx: Baud Rate,Parity,Word Size,Stop Bits,Timeout Flag

Parity: E , M, N, O, S (single character only, specifies Even, Mark, None, Odd, or Space parity)

Word Size: 7 or 8 bits

Stop Bits: 1 or 2

Timeout: P for infinite retry and any other character for normal retry.

Display Parameters:

The following values are valid. Certain modes will not work on certain displays, as specified below:

40: Set 40 columns in the current display type

80: Set 80 columns in the current display type

BW40: Set 40 column monochrome mode

BW80: Set 80 column monochrome mode

CO40: Set 40 column color (all but monochrome display)

CO80: Set 80 column color (all but monochrome display)

MONO: Resets the display adapter (mono display only)

COLOR: Resets the display adapter (color display only)

Example:

MORE

Syntax:

Purpose:

Options:

Default:

Example:

MOVE

Syntax:

MOVE {/options} oldfile newfile

Purpose:

Moves one or more files to a different directory.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for each file to be moved.

/NOC{ON{FIRM}}

This option causes the system not to request confirmation for each file to be moved.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option specifies that the old and new file specifications should be displayed as each file is moved.

/NOVER{BOSE}

This option specifies that the old and new file specifications should not be displayed as each file is moved.

Default Options:

Description:

This command acts like a COPY command which deletes the source file or files. Move will only work within the same physical device or partition. It will not, for example, move files from a floppy disk to a hard disk, or between partitions on a hard disk. MOVE will not accept ellipses, but will accept other wildcards.

The *oldfile* parameter specifies the file or files to move.

The *newfile* specifies the destination path, and optional filenames, if the files are to be renamed during a move. If the destination path is omitted, it is assumed to be the current directory.

Examples:

To move all files in \ABC\ and subdirectories to \DEF\ and create the same subdirectories

C:>MOVE \ABC\ \DEF\

temp.txt
misc.bat
2 files moved

To move all files on all directories on C: with a .TMP extension into D:\MISC, and rename them to a .BAK extension.

E:>MOVE

3 files moved

NETLINK

Syntax:

Purpose:

Options:

Default:

Example:

NETMODEM

Syntax:

Purpose:

Options:

Default:

Example:

NETSHOW

Syntax:

Purpose:

Options:

Default:

Example:

OBJDMP

Syntax:

Purpose:

Options:

Default:

Example:

PATH

Syntax:

PATH{=}{;}{{device:}path}{{;{device:}path}...}

Purpose:

Changes the list of paths searched when executing commands.

Options:

None.

Default Options:

None.

Description:

This function is provided for DOS compatibility. As such, it behaves exactly the same as the equivalent MS-DOS command. The XOS logical name CMD: performs the same function for the standard XOS environment, but adds certain features not found in the PATH command. It is recommended that the LOGICAL command be used to redefine the CMD: logical name instead of using the PATH command.

Typing PATH with no parameters will display the current path.

Typing PATH; will remove the existing path specification.

Devices and/or paths specified in the PATH command are separated by semicolons. Invalid devices or paths will not be detected by the PATH command: the invalid device or path specification will be ignored when searching the PATH.

Example:

To set the current path to include C:, C:DIR, and C:\DIR2\MENU:

```
C:>PATH C:\;C:\DIR;C:\DIR2\MENU
```

PING

Syntax:

PING {dev:}address{::}

Purpose:

Sends an IP ping message to a remote system and receives a response.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/T{IM{E}} = value

Specifies that repeated ping messages should be sent and sets the repeat interval. The value gives the interval in seconds.

Default Options:

Description:

The PING command provides a simple network diagnostic capability. It is normally used to verify that a remote system is available on the network or that the local network connection is functioning correctly.

The single argument must specify at least the IP address or domain name of the remote system. If a device is not specified, IPS0: is used. If a device is specified, it must be an IPS device. The double colon after the IP address or domain name is optional.

An IP ping packet is sent to the system specified. The round-trip time is displayed when the response is received. If the /TIME option is specified, this is repeated indefinitely at the specified interval until the process is terminated (usually by typing ^C).

PROMPT

Syntax:

PROMPT prompt_string

Purpose:

Sets or changes the command prompt.

Options:

None.

Default Options:

None.

Description:

This command is provided for DOS compatibility. It is recommended that the SETENV command be used for the PROMPT environment variable.

The command prompt can be completely customized using the options listed below. All options listed below must be preceded by a $ character.

Standard delimiters (comma, equal sign, semicolon, space, or tab) must be preceded by a null character. A null character is a $ followed by any character not listed below.

$

Dollar sign

_

Carriage return/Line feed (CRLF)

b

vertical bar (|)

d

Current date

e

ESC character (ASCII 27)

g

Greater than sign (>)

h

Backspace character (erases the previous character)

l

Less than sign (<)

n

Default drive

q

Equal sign (=)

t

Current time

p

Current path (drive + subdirectory)

Example:

To set the prompt to the current date and drive/path on the next line with a greater than (>) sign:

C:>PROMPT d_pg

01-Jan-90
C:\>_

To set the command prompt to DRIVE:\PATH>

C:>PROMPT PG

C:\COMM\TEMP>_

RENAME

Syntax:

RENAME {/options} oldname newname

Purpose:

Changes the name of a file.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for each file to be renamed.

/NOC{ON{FIRM}}

This option causes the system not to request confirmation for each file to be renamed.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option specifies that the old and new file specifications should be displayed as each file is renamed.

/NOVER{BOSE}

This option specifies that the old and new file specifications should not be displayed as each file is renamed.

Default Options:

Description:

This function is used to change the name of a file without copying or moving it. Only the file's name and extension can be changed; all other information about the file (date and time stamp, attribute bytes, etc.) remain the same. Note that hidden, read-only, and system files can be renamed. RENAME does not support ellipses, but does support other wildcards.

Example:

To rename all .BAK files to .OLD, override the /QUIET option, and confirm each rename:

C:>RENAME /NOQUIET /CONFIRM *.BAK *.OLD

To rename all files with no extension on D: to files with the extension .XXX:

C:>RENAME D:\...*. D:\...*.XXX

To rename VP.BAK to VP.BKU in the current directory:

C:>REN VP.BAK VP.BKU

RMBOOT

Syntax:

Purpose:

Options:

Default:

Example:

RMDIR

Syntax:

RMDIR {/option} dirname1 {dirname2 {...}}

RD {/option} dirname1 {dirname2{...}}

Purpose:

Removes (deletes) an empty directory.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/C{ON{FIRM}}

This option causes the system to request confirmation for each directory to be removed.

/NOC{ON{FIRM}}

This option causes the system not to request confirmation for each directory to be removed.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option causes the name of each directory to be displayed as the directory is removed.

/NOVER{BOSE}

This option causes the name of each directory to not be displayed as the directory is removed.

Default:

Description:

This command deletes the directories named. Each directory must be empty, except for the . and .. entries. RMDIR supports wildcards and ellipsis directory name specifications.

Example:

To remove the directory D:\TEMP:

C:>RMDIR D:\TEMP

RUN2EXE

Syntax:

Purpose:

Options:

Default:

Example:

RUNDMP

Syntax:

Purpose:

Options:

Default:

Example:

SET

Syntax:

SET { parameter={ value}}}

Purpose:

Lists, defines, or undefines a DOS environment variable.

Options:

None.

Default Options:

None.

Description:

This command is provided for compatibility with MS-DOS. It does not conform to the standard XOS command syntax. It is fully compatible with the DOS SET command. SETENV is the equivalent preferred XOS command.

If no arguments are specified, all environment variables are listed.

If an argument is specified, it must be an environment variable name and must be followed by an equal sign. There must be no space between the environment variable name and the equal sign. If there is, the trailing space(s) becomes part of the name.

Note:

If the equal sign is the last character on the line, the environment variable specified is deleted. Otherwise, the environment is defined as the string which follows the equal sign, including all spaces. If the string after the equal sign is placed in quotes, the quotes will be included in the definition of the environment variable.

This command does not allow the listing of individual environment variables.

Examples:

To set the TEMP environment variable:

C:>SET TEMP=C:\UTILITY\TEMP

To remove the DD4 variable:

C:>SET DD4=

SETENV

Syntax:

SETENV {/options} { parameter {= {value}}}

Purpose:

Lists, defines, or undefines XOS environment variables.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/E{DIT}

This option retrieves the current definition for the logical name specified so that it can be edited.

/PID = seq.num

This option sets environment variable for the specified process

/PL = num

This option sets the process level for the environment variable relative to current process.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/R{EN{AME}}

This option renames the logical name without changing the definition

/SL = num

This option sets the process level for the environment variable relative to session.

/SYS

This option sets system level environment variable.

Default Options:

Description:

This function manipulates the XOS environment variables. Functionally, this command is equivalent to the DOS compatible SET command but it conforms to the standard XOS command syntax and provides additional features.

Example:

Sets the XOS prompt to DRIVE:\PATH> (this is the equivalent of the PROMPT PG command)

C:>SETENV PROMPT=PG

Renames the variable TEMP to PDRIVE1:

C:>SETENV /RENAME TEMP=PDRIVE1

Sets the search path for XOS programs:

C:>SETENV CMD:="SYS:, C:\, D:\MISC"

SHELL

Syntax:

SHELL progname

Purpose:

Runs the SHELL from a child process.

Options:

None.

Default Options:

None.

Description:

This command executes a user shell, from which the user can enter XOS system level commands, or execute other programs. This command can also run *progname* as a shell, allowing custom user interfaces to be designed. Any command line parameters following *progname* will be passed to the shell program as its command line.

The standard XOS shell is designed to resemble the DOS COMMAND shell, and will act like the COMMAND shell in many ways. As noted in this manual, there are additional commands and options to commands supported by XOS. Even with these modifications, the details of using the shell and the overall appearance and actions will be familiar to DOS users.

Example:

To run the standard XOS command shell:

C:>SHELL

SHOW

Syntax:

SHOW {/options} parameters

Purpose:

Provides a display of system level information.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

Default Options:

Description:

This function displays technical information about various system parameters. Such information is usually not required for correct system operation.

Parameters:

The parameter can be one of the following:

DCS

Displays information about usage of the system's disk cache buffers and some additional information about other system data blocks (SDBs).

DEVICE

Displays information about devices connected to the system.

DISK

Lists all disks available on the system and displays information about each disk.

LKE

Displays information about all LKEs currently loaded. The information includes the name, version, compatibility level, and type of the LKE. Also displayed are the memory offsets and sizes of the code, data, and, if present, the symbol table parts of the LKE.

Example:

To show the disk cache status:

C:>SHOW DCS

```
                Total Avail Free 0  1 2 3 4 >4
System buffers: 203   124   14 110 77 1 1 1 0
Data buffers:   301   301    0 301  0 0 0 0 0
```

SYMBIONT

Syntax:

SYMBIONT {/options} programname parameters

Purpose:

Loads a program as a symbiont.

Options:

The options must be specified before the programname argument.

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Default Options:

Description:

This command loads a program as a symbiont, which is a program which runs independently in the background. The program which is loaded is not associated with any user session. The program to be loaded must be in the directory specified by the XOSSYS: logical name. A path cannot be specified. This is a security feature, since most symbionts are privileged programs.

All options and arguments specified after the programname argument are passed to the program as its arguments and options.

Note:

In order to execute as a symbiont, a program must be written to execute the necessary initial handshake with the system and must not attempt to access its controlling terminal since symbionts run without a controlling terminal.

SYSCHAR

Syntax:

SYSCHAR {/options} {char1=value1 {char2=value2 {...}}}

Purpose:

To display and modify characteristics of the SYSTEM: device class.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

/VER{BOSE}

This option specifies that a short text description should be displayed for each characteristic displayed.

/NOVER{BOSE}

This option specifies that a short text description should not be displayed for each characteristic displayed.

Default Options:

Initial option values are set from the ^XOS^SYSCHAR^OPT environment string. If no options are specified, the defaults are:

/NOQUIET /NOVERBOSE

Description:

This command is used to modify and display class characteristics for any device class in the system. Refer to chapter 5 for a description of the device classes and the class characteristics.

Each argument specifies a SYSTEM: class characteristic. If no characteristics are specified, the current values of all characteristics for the device class are displayed. If one or more characteristics are specified with values, the characteristics are set to the values specified. The values of all characteristics specified (with or without a new value being specified) are displayed (unless /QUIET was specified).

Note:

The SYSCHAR command is equivalent to CLSCHAR SYSTEM command.

Example:

SYSDIS

Syntax:

SYSDIS {interval}

Purpose:

Provides a real-time display of system operation.

Options:

None.

Default Options:

None.

Description:

The SYSDIS command provides a real time display of the operation of the system including the active sessions and processes, as well as memory and CPU usage.

The single argument specifies the update interval for the display in seconds. If it is not specified, a value of 1 second is used.

While the display is active, two commands are allowed:

I*n*

Where *n* is the interval in seconds, changes the display update interval.

Q

Terminates the display and returns to the command prompt.

Example:

TELNET

Syntax:

TELNET {dev:}address{::}

Purpose:

Implements an extended Telnet client.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/ANS{I}

This option prohibits the Telnet connection from using the extended XOS Telnet features. Only ANSI control characters are used for cursor positioning. This mode is automatically selected when connecting to a Telnet host which does not support the extended XOS features. If this mode is selected when connecting to an extended XOS telnet host, the pseudo-console used by the host to run programs emulates a serial port. It does not support most of the keyboard or display BIOS calls or direct screen memory reads or writes.

/XOS

This option causes an attempt to use the extended XOS Telnet features. This option has no effect if the Telnet host to which a connection is being made does not support the extended features. It also has no effect if the user's terminal is not a console display. The extended features provide support for all of the keyboard and display BIOS calls and for direct reads or writes to screen memory. The terminal emulated by the host on the remote system appears to be a console display.

/POR{T} = value

This option specifies the publicly known port on the remote system to which a TCP connection is established. If this option is not specified, port 23 (which is the standard Telnet public port) is used.

Default Options:

Description:

The TELNET command invokes the Telnet client to establish a terminal-like connection to a remote system.

The single argument must specify at least the IP address or domain name of the system to connect to. If a device is not specified, TCP0: is used. If a device is specified, it must be a TCP device. The double colon after the IP address or domain name is optional.

Example:

TIME

Syntax:

TIME

Purpose:

Displays and changes the current system time.

Options:

None.

Default Options:

None.

Description:

TIME is a DOS compatible command which does not conform to the standard XOS command syntax. It displays the system time, and then asks for a new time.

If a new time is specified, the system time is set to that specified. If a time is not specified (ENTER only), the system time is not changed.

This command is provided only to provide DOS compatibility. The DAYTIME command is the preferred XOS command.

Example:

C:>TIME

Current time is 15:34:28.40

Enter new time:_

TOUCH

Syntax:

Purpose:

Options:

Default:

Example:

TYPE

Syntax:

TYPE {/options} file1 {file2 {...}}

Purpose:

Lists files to the screen.

Options:

/H{ELP} or /?

This option takes no arguments. It causes a description of the command to be displayed. This description includes a listing of all valid options and arguments. If this option is specified, the display of this information is the only action taken. Any other options or arguments specified are ignored. All options displayed with an * are the defaults for this command.

/P{AU{SE}}

This option specifies that output should be paused at the end of each screen of output.

/NOP{AUSE}

This option specifies that output should not be paused at the end of each screen of output.

/NAM{ES}

This option specifies that the name of each file should be displayed (in inverse video) before the file is displayed

/NONAM{ES}

This option specifies that the name of each file should not be displayed before the file is displayed

/WRA{P}

This option specifies that lines longer than the line with of the output device should be broken into multiple lines by inserting carriage return/line-feed characters as necessary.

/NOWRA{P}

This option specifies that lines longer than the line width of the output device should not be broken into multiple lines.

Default Options:

Initial option values are set from the ^XOS^TYPE^OPT environment string. If no options are specified, the defaults are:

/NOPAUSE /NONAMES /NOWRAP

Description:

This command copies the file(s) specified to the session's controlling terminal. Wildcard and ellipsis filenames are allowed. Note that the only difference between this command and the MORE command is the initial option defaults.

Examples:

Displays all .DOC files in the current directory, pausing between screens and displaying the filename of each file:

```
C:>TYPE /PAUSE /NAMES *.DOC

[INFO.DOC]
This is a test.  This file is designed as a test of the
TYPE utility.  It is only three lines long.
This is the last line of the text file.
```

VER

Syntax:

VER

Purpose:

Displays the XOS version number.

Options:

None.

Default Options:

None.

Description:

This command is the equivalent of the DOS VER command, except that the information displayed is specific to XOS. The same information can be obtained from the SYSTEM device class using the SHOW SYSTEM command.

Example:

C:>VER

XOS Version 1.10

DOS Emulator version 3.3

VOL

Syntax:

VOL {drive:}

Purpose:

Displays a disk's volume label.

Options:

None.

Default Options:

None.

Description:

This command displays the volume label of the disk specified. It is the equivalent of the LABEL command, except that the volume label of the disk cannot be changed. If no drive is specified, the label of the current disk is displayed.

Example:

C:>VOL

Volume in drive C: is DRIVE-C

Chapter 4

Batch File Commands

This section describes the batch file commands which are available under the XOS environment. As in MS-DOS, most of these commands can be used at any time, even on the command line (the @ and : cannot be used on the command line), but are most useful in .BAT files.

A batch file is a text file with the extension .BAT which contains batch commands (the commands defined in this section of the manual). It may be executed by entering its name at an XOS prompt at any time. Any command arguments or options specified are passed to the batch file as parameters. For example, when running a batch file which requires a source and a target file directories: CPYFIL A: C:, the A: and C: are called parameters. Execution of batch files may be nested to any level, subject to availability of memory and other system resources. In other words, a batch file may run another batch file, which runs another, and so forth... without limits on the number of times this can happen. To maintain MS-DOS compatibility, a batch file that is run without using the CALL command will cause control to return to the command prompt upon completion.

Batch files can use any environment variables usable by the DOS shell, and can execute any command which can be executed from the command prompt.

:

Syntax:

:label

Purpose:

To provide a label for a GOTO branch.

Options:

None.

Description:

The : batch command is used to specify a batch label used by the GOTO command. Only the first 8 characters in the label are used by GOTO; the rest are ignored. Upper and lower case characters are treated as equivalent in labels and by the GOTO command.

Example:

:TOP

ECHO This will print until you type ^C

GOTO TOP

@

Syntax:

@batch_command

Purpose:

To suppress the echo of a single batch command.

Options:

None.

Description:

The @ batch command is used to suppress the echo of a single batch command without having to use ECHO OFF followed by ECHO ON. This also lets a batch file turn off echo using @ECHO OFF so that the ECHO OFF command will not display at the beginning of the batch file.

Example:

@ECHO OFF

BATOPT

Syntax:

BATOPT {/{NO}QUIET} {{CCP|ECHO|EXTEND}{={ON|OFF}}

Purpose:

To change how batch handles some of its internal operations.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

/Q{UI{ET}}

This option specifies that no output should be generated unless an error occurs.

/NOQ{UI{ET}}

This option specifies that informational output should be generated.

Description:

The BATOPT batch command permits a batch file to change the handling of various batch operations. The valid commands are as follows:

CCP - Control C prompting

ECHO - batch command echoing

EXTEND - extended batch command operation.

If CCP=ON is set then typing ^C to a running batch file will cause the following message to be printed:

Terminate batch job (Y/N)?

If CCP=OFF is set then typing ^C to a running batch file will cause the file to stop with no message.

The ECHO parameter works like the batch ECHO command and is included here both for completeness and to allow a batch file to test for the state of ECHO.

The EXTEND command enables the additional batch command RETURN and removes the restriction of directly accessing all batch parameters.

If a parameter is specified with no value, its value is displayed (unless /QUIET has been specified) and the errorlevel is set to 0 if it is OFF and 1 if it is ON. This permits you to test in a batch file if a specified parameter is set or not.

Example:

REM Turn off Control C prompting

BATOPT CCP=OFF

REM Show how to test for value of EXTEND

BATOPT/QUIET EXTEND

IF ERRORLEVEL 1 ECHO ERRORLEVEL SHOWS EXTEND IS ON

IF NOT ERRORLEVEL 1 ECHO ERRORLEVEL SHOWS EXTEND IS OFF

CALL

Syntax:

CALL filename {arguments}

Purpose:

To invoke another batch file and resume the current file after its completion.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The CALL batch command permits one batch file to call another, and to be able to return and execute the next batch command when the CALL is done.

The arguments are passed to the called batch file as command line arguments.

Example:

REM Call clear batch file to do cleanup at
REM the end of a run

CALL CLEAR

REM Return here when CLEAR is done.

ECHO

Syntax:

ECHO {ON|OFF|displayed text|:}

Purpose:

To control the display of the executing batch file on the user console

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The primary form of ECHO is with a string of text following. This will let a batch file display text on the user console to indicate batch file status or possible error conditions. If echo is on when this form is used, the text will appear twice, once with ECHO in front of the text and the second time with the text only.

The secondary form of ECHO is to control the display of the batch file that is executing. After the point at which ECHO OFF is encountered in the batch file, no other commands will display on the user console.

When a batch file is run, both the commands and any output from those commands will print on the user console. This is the default echo state, and corresponds to issuing the ECHO ON batch command. Using the ECHO: command (note that there is no space between the ECHO command and the colon(:)) will display a blank line.

Normally, ECHO OFF is displayed before it takes effect. If this is not desirable, @ECHO OFF will suppress display of the ECHO OFF command.

After the point at which ECHO ON is encountered in the batch file, all commands will again be displayed on the user console.

The final form of ECHO with no arguments is used to display the current state of echoing. If the echo state is on, the message ECHO is ON will be displayed.

Example:

ECHO OFF

ECHO This is a test of echo printing a message

ECHO:

ECHO The previous line is blank.

ECHO ON

FOR

Syntax:

FOR %%variable IN(list) DO command

Purpose:

To provide the ability to execute the same command over multiple items.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The FOR batch command is used to conditionally execute a command multiple times with different parameters.

The *%%variable* specifies which batch file variable is to be used in the FOR command.

The *(list)* is a list of items to compare against the variable.

The *command* is the command to execute, and can include batch file parameters.

Example:

REM This FOR will compile all C programs in

REM the current directory.

FOR %%I IN (*.C) DO CC.EXE %%I

GOTO

Syntax:

GOTO label

Purpose:

To control execution flow in a batch file.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The GOTO batch command changes the position in the batch file where execution will occur.

Execution will continue at the **LABEL** specified. If the label does not exist, the batch file will terminate.

Example:

REM do commands here...

IF ERRORLEVEL=20 GOTO FAILED

REM misc. other commands here...

GOTO END

:FAILED

ECHO Error encountered in run

RETURN

:END

REM Fall off the end of the batch file here.

IF

Syntax:

IF {NOT} condition command

Purpose:

The IF command is used for conditional control in a batch file.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The IF batch command is used to test for various conditions and execute a command if they are true. The optional modifier NOT may be used to reverse the sense of the condition. The conditions are as follows:

ERRORLEVEL n: Checks if the errorlevel from the last program or batch file was equal to or greater than the value n.

var1==var2: Checks if var1 is equal to var2

EXIST file: Checks if the specified file exists on the disk using the specified path.

Example:

IF %1==BAK ECHO BAK is for backup files

IF EXIST %1.EXE COPY %1.C %1.OLD

PAUSE

Syntax:

PAUSE {text}

Purpose:

To temporarily suspend the execution of a batch file.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

The PAUSE batch command is used to temporarily suspend the execution of a batch file and display the message “Strike a key when ready...”. Execution of the batch file will resume with any keypress except for CTRL-C or CTRL-BREAK.

Any *text* which is specified will replace the standard “Press any key to continue...” message displayed by the PAUSE command.

Example:

ECHO Please remove floppy #1 and replace it

ECHO with floppy #2.

PAUSE

REM

Syntax:

REM text string

Purpose:

To include comments in a batch file.

Options:

None.

Description:

The REM batch command is used to include comments in a batch file. Comments are not executed, and are provided by the writer of the batch file as reference or other information. Comments are only displayed if ECHO is turned on.

Example:

REM ********************************

REM   System startup batch file

REM ********************************

RETURN

Syntax:

RETURN {number}

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Purpose:

To exit from the current batch file and return to the calling batch file if there is one.

Description:

The RETURN batch command will return control to the calling batch file with a specific return value *number*, if specified, or with the return value from the last program run if *number* is not specified. This is equivalent to the SET ERRORLEVEL command, which is used for conditional branching in batch files.

Note:

This command is specific to XOS and will only work when the XOS batch extensions are enabled.

Example:

REM Batch file looks for a .C file, and

REM returns 200 if it does not exist.

REM Otherwise it returns the program

REM (CC.EXE) exit status

IF NOT EXIST %1.C RETURN 200

CC.EXE %1.C

RETURN

SHIFT

Syntax:

SHIFT

Purpose:

To shift the current list of batch arguments left by one argument.

Options:

/H{ELP} or /?

Displays a help screen. All other options or arguments are ignored.

Description:

To provide compatibility with DOS batch files that only access 10 batch file arguments at a time, the SHIFT command moves all arguments left one position, so that %1 is discarded, %2 becomes %1, and so forth until the tenth argument on the line becomes %9. Once the arguments have been shifted, there is no way to shift them back.

Example:

```
REM Type all specified names to the console

:TOP

IF %1.==. RETURN

ECHO “”""""""""""""""""""""""""

TYPE %1

ECHO “”""""""""""""""""""""""""

SHIFT

GOTO TOP
```

Chapter 5

Class Characteristics

This chapter describes the class characteristics for all standard device classes in the system. A class characteristic is a named item which specifies a value associated with a device class. Class characteristics are the primary mechanism by which the user obtains and sets the values of the various items which specify the state and operation of the XOS system. They provide a standardized interface which apply to all device classes.

Class characteristics are displayed or modified using the CLSCHAR command (see chapter 3, page 41).

A device class is either the collection of all devices of a specific type or is an abstract class which is not associated with any actual devices. For example, the TRM device class includes all terminal like devices. The DISK device class includes all disk devices, that is, all devices which support local file systems.

Abstract classes provide a mechanism which is used to access various values associated with the system as a whole e.g., the amount of memory available or the response of the system to CTL-ALT-DEL or with individual processes.

Table 5.1 gives a summary of the standard device classes.

Table 5.1 - Standard Device Classes	
Class name	**Description**
SYSTEM	Abstract class for system wide values
PROCESS	Abstract class for process values
SESSION	Abstract class for session values
DISK	Local mass storage devices
SPL	Spooled devices
TRM	Terminal devices
PCN	Pseudo-console devices
IPM	Interprocess message device
NULL	Null device
PPR	Parallel printer devices
NET	Network hardware level devices
SNAP	Network link level devices
ARP	Network Internet ARP level devices
IPS	Network Internet IP level devices
UDP	Network Internet UDP level devices
TCP	Network Internet TCP level devices
TLN	Network Internet TLN level devices
XFP	Network Internet XFP level devices

The remainder of this chapter describes each of these device classes and the associated class characteristics in detail.

Table 5.2 lists the names that are used to describe the format of the characteristic values.

Table 5.2 - Characteristics value formats		
Name	**Type**	**Description**
DECV	Numeric	Decimal value
HEXV	Numeric	Hex value
OCTV	Numeric	Octal value
BINV	Numeric	Binary value
DECB	Numeric	Multiple item decimal value (a.b.c.d)
HEXB	Numeric	Multiple item hex value (a-b-c-d)
OCTB	Numeric	Multiple item octal value (a-b-c-d)
VERN		Version number value (a.b.c)
TIME	Text	Time value (hh:mm:ss.fff)
DATE	Text	Date value (dd-mmm-yy)
DT	Text	Date and time value
TEXT	Text	Text value (case is ignored)
STR	Text	String value (case is significant)

Values are displayed according to the format indicated. A new value specified must match the basic type of the value (numeric or text). All numeric input values default to decimal radix, even if a different radix is indicated by the format. A number which begins with a leading 0 is taken as octal and a number which begins with a leading 0x is taken as hex. Most characteristics place limits on the range of acceptable numeric values on on the acceptable syntax or length of text values. Characteristics with formats for which no type is indicated can be displayed only. Their values can not be changed.

In the tables which follow, the Set column indicates if the value of the characteristic can be changed. An S indicates the value can be changed, a blank indicates that it cannot be changed. A V indicates that the value can be verified. If a value is specified, it is compared to the valid value for the characteristic. If it is not the same, an error is indicated.

SYSTEM Class Characteristics

The SYSTEM device class is an abstract device class which provides a mechanism for accessing and changing various values associated with the XOS system as a whole. The SYSTEM class characteristics are summarized in Table 5.3.

Table 5.3 - SYSTEM class characteristics

Name	Format	Set	Description
ALMLIMIT	DECV	S	Maximum number of alarms per process
AVAILMEM	DECV		Available memory (in KB)
COUNTRY	DECV	S	Country code for system
DEBUG	TEXT		Exec debugger present
DOSVER	VERN	S	Default DOS emulator version number
FPUENB	TEXT	S	Default floating point processor enable state
FPUTYPE	TEXT		Floating point processor type
HIGHDMA	HEXV	S	Highest physical address for DMA
INITIAL	TEXT	S	Run initial command SHELL on startup
KBRESET	TEXT	S	Can reset system from console keyboard
LOADDATE	DATE		Date system was loaded
LOADTIME	TIME		Time system was loaded
LOGIN	TEXT	S	User login required
NUMFLPY	DECV		Number of floppy disk units in system
NUMHARD	DECV		Number of hard disk units in system
NUMPAR	DECV		Number of parallel ports in system
NUMSER	DECV		Number of serial ports in system
OMLIMIT	DECV	S	Overhead memory limit (in KB)
PMLIMIT	DECV	S	Protected mode memory limit (in KB)
PROCTYPE	TEXT		Processor type
PROINUSE	DECV		Number of processes/shared sections in use
PROLIMIT	DECV	S	Limit of number of processes/shared sections
REALBASE	DECV	S	Base address for real mode image (in KB)
REALSIZE	DECV	S	Default size of real mode DOS image (in KB)
RMLIMIT	DECV	S	Real mode memory limit (in KB)
SELINUSE	DECV		Number of global selectors in use
SELNUM	DECV		Number of global selectors created
SERNUM	DECV		Kernel serial number
SPEED	DECV		Processor speed factor
STATE	HEXV	S	System state
SYSNAME	STR		Name of the system
TMLIMIT	DECV	S	Total memory limit (in KB)

Table 5.3 - SYSTEM class characteristics			
Name	**Format**	**Set**	**Description**
TOTALMEM	DECV		Total memory in system (in KB)
USERMEM	DECV		User memory in system (in KB)
WSLIMIT	DECV	S	Working set size limit (in KB)
XFFINUSE	DECV		Number of extended fork frames in use
XFFLIMIT	DECV	S	Limit of number of extended fork frames
XFFMAX	DECV	S	Maximum extended fork frames in use
XFFNUM	DECV		Current number of extended fork frames
XMBAMAX	DECV		Maximum number of available exec buffers
XMBAVAIL	DECV		Number of available exec buffers
XMBINUSE	DECV		Number of exec buffers in use
XMBMAX	DECV		Maximum number of exec buffers in use
XMBRESRV	DECV	S	Number of reserve exec buffer pages
XOSVER	VERN		XOS version number

The following sections describe each of the SYSTEM class characteristics in detail.

ALMLIMIT - Maximum number of alarms per process

This class characteristic sets or returns the maximum number of alarms that can be active for a process. Its purpose is to prevent a program from using an excessive number of system resources. The initial value is 16, which should be adequate for most systems. It should be increased if programs indicate an ER_TMALM error.

AVAILMEM - Available memory

This read only class characteristic returns the amount of memory (in KB) currently available for allocation.

COUNTRY - Country code for system

This class characteristic sets or returns the system's current default country code. This specifies the country for the XOS national language features. The country code is the same as the internal telephone prefix for a given country. The initial value is 1 (USA). This value is used to initialize the country code value for a process when it is created.

DEBUG - Exec debugger present

This read only class characteristic returns a value of YES if the exec debugger (XDT) is loaded with the kernel or a value of NO if it is not loaded.

DOSVER - DOS emulator version number

This class characteristic sets or returns the default DOS emulator version number. This is the number returned to a DOS program which issues the get version number DOS system call. Currently, it has no other effect. Future versions of XOS may use this value to customize the behavior of the DOS emulator to match specific versions of DOS.

FPUENB - Floating point processor enabled

This class characteristic set or returns the current floating point processor enable state. Valid values are YES indicating that the FPU is enabled and NO indicating that the FPU is not enabled. This value is used a process' initial FPU state when the process is created. This value is initialized to YES if a floating point processor is present and to NO if not. The value cannot be changed unless a floating point processor is present.

FPUTYPE - Floating point processor type

This class characteristic returns the type of floating point processor available on the system. Possible value are NONE, 80387, and 80487.

XOSVER - XOS version number

This class characteristic returns the version number of the system.

HIGHDMA - Highest physical address for DMA

This class characteristic set or returns the highest physical address for DMA transfers. The initial value of this class characteristic is 0xFFFFFFFF, which is the highest possible physical address. This value allows DMA transfers to any memory on the system. This is correct for many 386/486 machines, but a few have hardware limitations such that DMA transfers to memory above a certain (machine dependent) address will not work correctly. Some machines will not support DMA transfers to what is referred to as reserved or shadow memory. This is a small amount of memory (between 128KB and 386KB) which is usually physically mapped just below 16MB. If there is a problem with DMA transfers on a given machine, setting this value to 0xF00000 (15MB) will usually correct the problem. However, some machines may require a different value.

INITIAL - Run initial command shell on startup

This class characteristic specifies if the command shell should be run on session 1 on the system's console at startup. This value is only meaningful at startup time. It can be set using a CLSCHAR or SYSCHAR command in the STARTUP.BAT file. The value of this characteristic immediately after STARTUP.BAT is executed is used to determine if a command shell is started. Valid values are YES and NO. The initial value is NO.

KBRESET - Can reset system from console keyboard

This class characteristic specifies if the CTL-ALT-DEL key sequence from the console keyboard causes a system re-boot. Valid values are YES and NO. The initial value is YES, which enables the CTL-ALT-DEL key combination to re-boot the system. Changing this value requires the ADMIN privilege.

LOADDATE - Date system was loaded

This class characteristic returns the date the system was loaded in the format dd-mmm-yy, where *dd* is a one or two digit day of the month, *mmm* is the three letter name for the month, and *yy* is the last two digits of the year.

LOADTIME - Time system was loaded

This class characteristic returns the time the system was loaded in the format hh:mm:ss, where *hh* is hours (24 hour format), *mm* is minutes, and *ss* is seconds.

LOGIN - User login required

This class characteristic specifies if the LOGIN feature is used. A value of YES indicates that users must specify a user name and password before using the system. A value of NO indicates DOS style operation with no formal login required. The initial value is NO. Setting this value to YES with a command in the STARTUP.BAT file guarantees that no one will be able to access the system without logging in.

NUMFLPY - Number of floppy disk units in system

This read only class characteristic returns the number of floppy disk in the system.

NUMHARD - Number of hard disk units in system

This read only class characteristic returns the number of hard disks in the system.

NUMPAR - Number of parallel ports in system

This read only class characteristic returns the number of parallel ports in the system.

NUMSER - Number of serial ports in system

This class characteristic returns the number of serial ports in the system.

OMLIMIT - Overhead memory limit (in KB)

This class characteristic sets or returns the maximum amount of system overhead memory allowed per process. A value of 0 indicates that there is no limit. There is also a PROCESS class OMLIMIT characteristic for each process. The actual value used is the smaller of these two values. The initial value is 0. The minimum valid value is 30. Values less than 100 may prevent some programs (especially DPMI programs) from running. The value should be left at 0 unless it is suspected that a specific program is causing a problem by using too much system overhead memory.

PMLIMIT - Protected mode memory limit (in KB)

This class characteristic sets or returns the maximum amount of protected mode memory allowed per process. A value of 0 indicates that there is no limit. There is also a PROCESS class PMLIMIT characteristic for each process. The actual value used is the smaller of these two values. The initial value is 0.

Note: Many DPMI programs allocate all available memory in the system, regardless of how much memory is available. Setting this value to a reasonable level will prevent such programs from using all of the system's memory, while still allowing them to run. A reasonable value is between 1000 and 4000, depending on the individual programs being run and the total amount of memory available. If a mixture of programs are being run, it may be better to set the limit at the process level instead, using the PMLIMIT class characteristic for the PROCESS class.

PROINUSE - Number of processes/shared sections

This read only class characteristic returns the number of processes and shared memory sections currently active in the system.

PROLIMIT - Limit of number of processes/shared sections

This class characteristic specifies the maximum number of processes and shared memory sections which can be active in the system. Making this value very large does not consume any system resources. It simply serves as a limit to prevent overloading the system with too many processes or shared memory sections. The initial value is 10,000, which is effectively no limit.

REALBASE - Default base address for real mode image

This class characteristic specifies the default base address for real mode images (in KB). This is the lowest real mode address at which memory is allocated (excluding memory between 0 and 0xFFF, which is always available). This value is used to initialize the PROCESS class characteristic REALBASE when a level 1 process is created, that is, when a new command shell is created. The initial value of this characteristic is 0.

This value serves two purposes. First it allows a lower limit to be placed on the memory used by DOS programs. A few DOS programs do not work correctly if loaded too low in memory. Second, it can be used to prevent the use of real mode memory between 0x1000 and 0xFFFF. Use of memory in this area can cause conflicts with I/O devices which use I/O registers between 0x1000 and 0xFFFF on some early steppings of the 80386 due to a bug in the processor.

REALSIZE - Default maximum size of real mode DOS image

This class characteristic specifies the default maximum size of a real mode DOS image (in KB). This effectively specifies the amount of base memory available to a DOS program running under XOS. This value is used to initialize the PROCESS class characteristic REALBASE when a level 1 process is created, that is, when a new command shell is created. The initial value of this characteristic is 640.

RMLIMIT - Real mode memory limit (in KB)

This class characteristic sets or returns the maximum amount of real mode memory allowed per process. A value of 0 means no limit. There is also a PROCESS class RMLIMIT characteristic for each process. The actual value used is the smaller of these two values. The initial value is 0.

It is generally not necessary to limit real mode memory usage using this value since there is an inherent limit of 640KB imposed by the architecture. A lower limit is not usually necessary or desirable.

SELINUSE - Number of global selectors in use

This read only class characteristic returns the number of global selectors currently in use by the system. Global selectors are used by the system when addressing memory and represent a finite, but large, resource.

SELNUM - Number of global selectors created

This read only class characteristic returns the number of global selectors which have been created by the system. This value will generally be slightly larger than the SELINUSE value.

SERNUM - Kernel serial number

This class characteristic returns the kernel serial number for the copy of XOS being used. Each copy of XOS is serialized with a unique serial number. This number is used to verify that multiple systems on a network are not running from the same copy of XOS. Any attempt to modify this serial number will result in a fatal system error.

SPEED - Processor speed factor

This read only class characteristic returns a number which is roughly proportional to processor speed. This value is not intended to be a comprehensive measure of processor speed, but is just a simple measure of the speed of a simple timing loop. It is used internally by the system to calibrate very short time delays. A 16MHz 386DX machine should give a value of about 10. Other speed machines should be roughly proportional, with machines with a cache giving somewhat higher than expected values.

STATE - System state

This class characteristic sets or returns a value which represents the current state of the system. The value is bit encoded as described in Table 5.4.

Table 5.4 System state values

Bit	Name	Meaning
0	SS$STRTCOMP	System startup complete

Setting or getting this value requires the ADMIN privilege. This value is used internally by the system and should not be changed by the user.

SYSNAME - System name

This class characteristic returns the name of the system. This is a text string which specifies the name and version of the XOS operating system.

TMLIMIT - Total memory limit (in KB)

This class characteristic sets or returns the maximum total amount memory allowed per process. This is the sum of real mode, protected mode, and system overhead memory. A value of 0 means no limit. There is also a PROCESS class TMLIMIT characteristic for each process. The actual value used is the smaller of these two values. The initial value is 0.

TOTALMEM - Total memory in system

This read only class characteristic returns the total amount of memory (in KB) in the system.

USERMEM - User memory in system

This read only class characteristic returns the total amount of user memory (in KB) in the system. User memory is memory which is not allocated to the kernel.

WSLIMIT - Work set size limit (in KB)

This class characteristic sets or returns the maximum working set size allowed per process. This is the maximum amount of physical memory that will be allocated to a process. A value of 0 means no limit. There is also a PROCESS class WSLIMIT characteristic for each process. The actual value used is the smaller of these two values. The initial value is 0.

XFFINUSE - Number of extended fork frames in use

This read only class characteristic returns the number of extended fork frames currently in use by the system.

XFFLIMIT - Limit of number of extended fork frames

This class characteristic specifies a maximum limit for the number of extended fork frames used by the system. Setting this limit to a very large value does not consume any system resources. This value provides a way to limit the maximum amount of I/O activity in the system. Normally there is no need to use this limit. It is initially set to 10,000, which is effectively no limit.

XFFMAX - Maximum number of extended fork frames in use

This class characteristic records the maximum number of extended fork frames that have been in use at any one time.

XFFNUM - Current number of extended fork frames

This read only class characteristic returns the number of extended fork frames currently in use by the system.

XMBAMAX - Maximum number of available exec buffers

This read only class characteristic returns the maximum number of exec buffers which have been available on each of the exec buffer free lists. There are seven exec buffer free lists, corresponding to buffer sizes of 64, 128, 256, 512, 1024, 2048, and 4096 bytes. Note that each of the maximum values is maintained independently.

XMBAVAIL - Number of available exec buffers

This read only class characteristic returns the number of exec buffers currently on each of the exec buffer free lists. There are seven exec buffer free lists, corresponding to buffer sizes of 64, 128, 256, 512, 1024, 2048, and 4096 bytes.

XMBINUSE - Number of exec buffers in use

This read only class characteristic returns the number of exec buffers currently in use. Seven numbers are returned, corresponding to exec buffer sizes of 64, 128, 256, 512, 1024, 2048, and 4096 bytes.

XMBMAX - Maximum number of exec buffers in use

This read only class characteristic returns the maximum number of exec buffers which have been in use. Seven numbers are returned, corresponding to buffer sizes of 64, 128, 256, 512, 1024, 2048, and 4096 bytes. Note that each of the maximum values is maintained independently.

XMBRESRV - Number of reserve exec buffer pages.

This class characteristic sets or returns the number of reserve buffer pages (each page is 4096 bytes) to be maintained by the system. Reserve pages are allocated from the general memory pool and held in reserve for use when exec buffers must be allocated at fork level. The initial value is 2. This may have to be increased to 4 or more if there is heavy network activity.

PROCESS and SESSION Class Characteristics

The PROCESS and SESSION device classes are abstract device classes which provide a mechanism for accessing and changing various values associated with individual processes in the system. The two classes are identical except for the default process referenced. The PROCESS class initially references the process which issues the system call to set or return characteristics values. The SESSION class initially references the session level process for the process issuing the system call. This is usually the command processor.

It should be noted that the process referenced for both classes can be changed by setting the values of certain characteristics. These characteristics do not actually change any values associated with the process but simply select which process is referenced by any characteristics which follow. They can also be used to return the indicated values for the process currently being referenced.

The PROCESS and SESSION class characteristics are summarized in table 5.5

Table 5.5- PROCESS and SESSION class characteristics

Name	Format	Set	Description
CONTRM	TEXT	S	Select process by name of controlling terminal
COUNTRY	TEXT	S	Country code
FPUENB	TEXT	S	Floating point processor enabled
LABLKS	DECV		Number of linear address blocks in use
LAINUSE	DECV		Linear address space in use (in KB)
LALARGE	DECV		Largest available linear address block (in KB)
NAME	STR	S	Process name
NUM	DECV	S	Select process by number
OMALLOW	DECV	S	System overhead memory allowed (in KB)
OMINUSE	DECV		System overhead memory in use (in KB)
OMLIMIT	DECV	S	System overhead memory limit (in KB)
PMALLOW	DECV	S	Protected mode memory allowed (in KB)
PMINUSE	DECV		Protected mode memory in use (in KB)
PMLIMIT	DECV	S	Protected mode memory limit (in KB)
PRIV	STR	S	Current process privileges
PRIVAVL	STR	S	Available process privileges
REALBASE	DECV	S	Base offset for real mode image (in KB)
REALSIZE	DECV	S	Default size of real mode DOS image (in KB)
RMALLOW	DECV	S	Real mode memory allowed (in KB)
RMINUSE	DECV		Real mode memory in use (in KB)

Table 5.5- PROCESS and SESSION class characteristics

Name	Format	Set	Description
RMLIMIT	DECV	S	Real mode memory limit (in KB)
SEQ	DECV	V	Process sequence number
SHRDELAY	DECV	S	File sharing delay value
SHRRETRY	DECV	S	File sharing retry value
TMALLOW	DECV	S	Total memory allowed (in KB)
TMINUSE	DECV		Total memory in use (in KB)
TMLIMIT	DECV	S	Total memory limit (in KB)
WSALLOW	DECV	S	Working set size allowed (in KB)
WSINUSE	DECV		Working set size (in KB)
WSLIMIT	DECV	S	Working set size limit (in KB)

The following section describes each of the PROCESS and SESSION class characteristics in detail.

CONTRM - Controlling terminal

This class characteristic returns the name of the currently selected process's controlling terminal or selects a process based on the terminal name specified. When the value of this characteristic is set, the process which is lowest in the process tree which has the specified terminal as its controlling terminal is selected for further class characteristic processing. If there is more than one process at this level, which one is selected is not defined. The terminal name must be a physical device name specified without a trailing colon.

COUNTRY - Country code

This class characteristic sets or returns the current country code for the currently selected process. This specifies the country for the XOS national language features. The country code is the same as the internal telephone prefix for a given country. The initial value is set from the value of the SYSTEM class characteristic COUNTRY when the process is created.

FPUENB - Floating point processor enabled

This class characteristic set or returns the current floating point processor enable state for the selected process. Valid values are YES (indicating that the FPU is enabled) and NO (indicating that the FPU is not enabled). This value is initialized from the SYSTEM class characteristic FPUENB when the process is created. The value cannot be changed unless a floating point processor is present.

LABLKS - Number of linear address blocks in use

This class characteristic returns the number of DPMI memory blocks that have been allocated.

LAINUSE - Linear address space in use (in KB)

This class characteristic returns the amount of linear address space which is in use, in KB.

LALARGE - Largest available linear address block (in KB)

This class characteristic returns the size, in KB, of the largest available contiguous linear address space.

NAME - Process name

This class characteristic returns or sets the name of the currently selected process.

NUM - Process number

This class characteristic returns the number of the currently selected process or selects a process based on process number. When the value of this characteristic is set, the process in the system with a matching process number is selected for further class characteristic processing. The process number is a 32-bit value. The high order 16-bits of the value contain the process sequence number and the low order 16 bits contain the process index. If the process sequence number is specified as 0, any process with the specified index is selected. If the process sequence number is specified as a non-0 value, an ER_CHARV error is returned if it does not match the current sequence number of the process whose index was specified.

OMALLOW - System overhead memory allowed (in KB)

This class characteristic returns or sets the maximum amount of system overhead memory, in KB, allowed for the selected process. A value of 0 indicates that there is no limit. The value can be decreased but cannot be increased. The value is initialized to the value specified by the parent process (up to the parent's own OMALLOW value) when the process is created. This value is not used directly when allocating memory but serves as an upper limit when changing the value of the OMLIMIT class characteristic.

OMINUSE - System overhead memory in use (in KB)

This class characteristic returns the amount of system overhead memory, in KB, in use by the selected process.

OMLIMIT - System overhead memory limit (in KB)

This class characteristic sets or returns the maximum amount of system overhead memory allowed for the selected process . A value of 0 means no limit. There is also a SYSTEM class OMLIMIT characteristic. The actual value used is the smaller of these two values. The initial value is the value specified by the parent process when the selected process is created, but it cannot be more than the parent's OMALLOW value. The minimum valid value is 30. The maximum valid value is the value of the process' OMALLOW characteristic. Values less than 100 may prevent some programs (especially DPMI programs) from running. The value should be left at 0 unless it is suspected that a specific program is causing a problem by using too much system overhead memory.

PMALLOW - Protected mode memory allowed (in KB)

This class characteristic returns or sets the maximum amount of protected mode memory, in KB, allowed for the selected process. A value of 0 indicates that there is no limit. The value can be decreased but cannot be increased. The value is initialized to the value specified by the parent process (up to the parent's own PMALLOW value) when the process is created. This value is not used directly when allocating memory but serves as an upper limit when changing the value of the PMLIMIT class characteristic.

PMINUSE - Protected mode memory in use (in KB)

This class characteristic returns the amount of protected mode memory, in KB, in use by the selected process.

PMLIMIT - Protected mode memory limit (in KB)

This class characteristic sets or returns the maximum amount of protected memory allowed for the selected process. A value of 0 means no limit. There is also a SYSTEM class PMLIMIT characteristic. The actual value used is the smaller of these two values. The initial value is the value specified by the parent process when the selected process is created, but it cannot be more than the parent's PMALLOW value. The maximum valid value is the value of the process' PMALLOW characteristic.

Many DPMI programs allocate all available memory in the system, regardless of how much memory is available. Setting this value to a reasonable level will prevent such programs from using all of the system's memory, while still allowing them to run reasonably. A reasonable value is between 1000 and 4000, depending on the individual programs being run and the total amount of memory available.

PRIV - Current process privileges

This class characteristic returns or sets the current process privileges for the currently selected process. Note that this characteristic does not change process selection. The value must be a string which consists of a list of privilege items separated by + or -. If the string begins with + or -, the privileges specified are added (+) or removed (-) for the selected process. If the first character is not + or -, the process's current privileges are cleared and then set to all privileges preceded with +. A privilege cannot be added unless it is an available process privilege as reported by the PRIVAVL characteristic.

PRIVAVL - Available process privileges

This class characteristic returns or sets the current available process privileges for the currently selected process. Note that this characteristic does not change process selection. Privileges cannot be added to the current value. The value must be a string with the same format as specified for the PRIV characteristic.

REALBASE - Base offset for real mode image (in KB)

This class characteristic specifies the default base address for real mode images (in KB). This is the lowest real mode address at which memory is allocated (excluding memory between 0 and 0xFFF, which is always available). The initial value is taken from the parent process' REALBASE value when the process is created.

This value serves two purposes. First it allows a lower limit to be placed on the memory used by DOS programs. A few DOS programs do not work correctly if loaded too low in memory. Second, it can be used to prevent the use of real mode memory between 0x1000 and 0xFFFF. Use of memory in this area can cause conflicts with I/O devices which use I/O registers between 0x1000 and 0xFFFF on some early steppings of the 80386 due to a bug in the processor. The initial value of this characteristic is 0.

REALSIZE - Default size of real mode DOS image (in KB)

This class characteristic specifies the default maximum size of a real mode DOS image. This effectively specifies the amount of base memory available to a DOS program running under XOS. The initial value of this characteristic is set from the parent process' value when the process is created.

RMALLOW - Real mode memory allowed (in KB)

This class characteristic returns or sets the maximum amount of real mode memory allowed for the selected process. A value of 0 indicates that there is no limit. The value can be decreased but cannot be increased. The value is initialized to the value specified by the parent process (up to the parent's own RMALLOW value) when the process is created. This value is not used directly when allocating memory but serves as an upper limit when changing the value of the RMLIMIT class characteristic.

RMINUSE - Real mode memory in use (in KB)

This class characteristic returns the amount of real mode memory in use by the selected process.

RMLIMIT - Real mode memory limit (in KB)

This class characteristic sets or returns the maximum amount of real mode memory allowed for the selected process. A value of 0 indicates that there is no limit. There is also a SYSTEM class RMLIMIT characteristic. The actual value used is the smaller of these two values. The initial value is the value specified by the parent process when the selected process is created, but it cannot be more than the parent's RMALLOW value. The maximum valid value is the value of the process' RMALLOW characteristic.

It is generally not necessary to limit real mode memory usage using this value since there is an inherent limit of 640 KB imposed by the architecture. A lower limit is not usually necessary or desirable.

SEQ - Sequence number

This class characteristic gets the sequence number of the currently selected process or verifies that the sequence number specified matches that of the currently selected process. If it does not match, an ER_CHARV error is returned.

SHRDELAY - File sharing delay value

This class characteristic returns or sets the delay time, in milliseconds, used when a file sharing violation occurs. The initial value is 50 milliseconds.

SHRRETRY - File sharing retry value

This class characteristic returns or sets the number of times an operation will be retried after a file sharing violation. The initial value is 0 (no retry) in XOS environments. The value is set to 3 when a DOS environment is created.

TMALLOW - Total memory allowed (in KB)

This class characteristic returns or sets the maximum total amount of memory allowed for the selected process. A value of 0 indicates that there is no limit. The value can be decreased but cannot be increased. The value is initialized to the value specified by the parent process up to the parent's own TMALLOW value when the process is created. This value is not used directly when allocating memory but serves as an upper limit when changing the value of the TMLIMIT class characteristic.

TMINUSE - Total memory in use (in KB)

This class characteristic returns the total amount of memory by the selected process.

TMLIMIT - Total memory limit (in KB)

This class characteristic sets or returns the total maximum amount of memory allowed for the selected process. A value of 0 means no limit. There is also a SYSTEM class TMLIMIT characteristic. The actual value used is the smaller of these two values. The initial value is the value specified by the parent process when the selected process is created, but it cannot be more than the parent's TMALLOW value. The maximum valid value is the value of the process' TMALLOW characteristic.

WSALLOW - Working set size allowed (in KB)

This class characteristic returns or sets the maximum working set size, in KB, allowed for the selected process. A value of 0 means no limit. The value can be decreased but cannot be increased. The value is initialized to the value specified by the parent process (up to the parent's own

WSALLOW value) when the process is created. This value is not used directly when allocating memory but serves as an upper limit when changing the value of the WSLIMIT class characteristic.

WSINUSE - Working set size (in KB)

This class characteristic returns the current size, in KB, of the working set of the selected process.

WSLIMIT - Working set size limit (in KB)

This class characteristic sets or returns the maximum working set size allowed for the selected process. A value of 0 means no limit. There is also a SYSTEM class WSLIMIT characteristic. The actual value used is the smaller of these two values. The initial value is the value specified by the parent process when the selected process is created, but it cannot be more than the parent's WSALLOW value. The maximum valid value is the value of the process' WSALLOW characteristic.

DISK Class Characteristics

The DISK device class includes all disk-like devices. There are many devices which support a local file system. Some systems (such as DOS) refer to these as block devices. The DISK class characteristics are summarized in Table 5.6.

Table 5.6 - DISK class characteristics

Name	Format	Set	Description
AHEAD	DECV	S	Maximum read-ahead blocks
LIMIT	DEVC	S	Maximum number of disk devices allowed
MAXIMUM	DECV	S	Maximum number of in use disk devices
NUMBER	DECV		Number of in use disk devices
NUMDBUF	DECV	S	Number of disk data buffers
NUMSBUF	DECV	S	Number of disk system buffers

The following sections describe each of the DISK class characteristics in detail.

AHEAD - Maximum read-ahead blocks

This class characteristic returns or sets the maximum number of blocks which will be read ahead when accessing a disk. This value is a maximum for the system. Individual disks may set a lower read ahead limit.

LIMIT - Maximum number of disk devices allowed

This class characteristic specifies the maximum number of disk devices which can be in use at any one time. This is effectively the maximum number of files which can be simultaneously open. Setting this characteristic to a very large value does not consume any system resources. It is initially set to 10,000, which is effectively no limit.

MAXIMUM - Maximum number of in use disk devices

This class characteristic records the maximum number of disk devices in use at any one time. This is effectively the maximum number of simultaneously open files.

NUMBER - Number of in use disk devices

This read only class characteristic reports the number of disk devices currently in use. This is effectively the number of currently open files.

NUMDBUF - Number of disk data buffers

This class characteristic returns the number of disk data buffers in the system or allocates additional disk data buffers. Disk data buffers are used to cache user data read from the disk and for read ahead of user data. If the value is specified, it is first rounded up to a multiple of 7. If this value is larger than the current number of disk data buffers in the system, additional disk data buffers are allocated to increase the total number to the value specified. If this value is equal to or smaller than the current number of disk data buffers in the system, nothing is done. Allocating additional buffers requires the ADMIN privilege.

NUMSBUF - Number of disk system buffers

This class characteristic returns the number of disk system buffers in the system or allocates additional disk data buffers. Disk system buffers are used to cache system accesses to the disk (such as directory and FAT reads) and to store information about open files. If the value is specified, it is first rounded up to a multiple of 7. If this value is larger than the current number of disk data buffers in the system, additional disk system buffers are allocated to increase the total number to the value specified. If this value is equal to or smaller than the current number of disk system buffers in the system, nothing is done. Allocating additional buffers requires the ADMIN privilege.

SPL Class Characteristics

The SPL device class includes the SPLn devices. This device is basically a filter in front of the DISK device class which implements transparent spooling to disk for data intended for printers or other non-sharable devices. There are no class characteristics associated with the SPL device class.

TRM Class Characteristics

The TRM device class includes all terminal like devices. These include the console/keyboard device and all serial port devices (except for the serial network driver). The TRM class characteristics are summarized in Table 5.7.

Table 5.7 - TRM Class Characteristics

Name	Format	Set	Description
LIMIT	DECV		Maximum number of terminal devices allowed
MAXIMUM	DECV	S	Maximum number of in use terminal devices
NUMBER	DECV		Number of in use terminal devices

The following sections describe each of the TRM class characteristics in detail.

LIMIT - Maximum number of terminal devices allowed

This class characteristic specifies the maximum number of terminal devices which can be in use at any one time.

MAXIMUM - Maximum number of in use terminal devices

This class characteristic records the maximum number of terminal devices in use at any one time. This is the maximum number of simultaneously open terminal devices.

NUMBER - Number of in use terminal devices

This read only class characteristic returns the number of terminal devices currently in use in the system. This is the number of open terminal devices.

PCN Class Characteristics

The PCN device class includes only the PCNn devices. These devices implement a device which can be used to emulate a console terminal. Each unit has two sides. One side connects to the TRM class devices as a terminal interface driver. The other side (the actual PCNn device) is accessed by a server program which reads the data output to the console screen and outputs data which appears to have been input from the console keyboard.

The PCN class characteristics are summarized in Table 5.8.

Table 5.8 - PCN Class Characteristics

Name	Format	Set	Description
LIMIT	DECV		Maximum number of PCN devices allowed
MAXIMUM	DECV	S	Maximum number of in use PCN devices
NUMBER	DECV		Number of in use PCN devices

The following sections describe each of the PCN class characteristics in detail.

LIMIT - Maximum number of PCN devices allowed

This class characteristic specifies the maximum number of PCN devices which can be in use at any one time.

MAXIMUM - Maximum number of in use PCN devices

This class characteristic records the maximum number of PCN devices in use at any one time. This is the maximum number of simultaneously open PCN devices.

NUMBER - Number of in use PCN devices

This read only class characteristic returns the number of PCN devices currently in use in the system. This is the number of open PCN devices.

IPM Class Characteristics

The IPM device class includes only the IPM: (InterProcess Message) device. This device is used for sending messages (datagrams) between processes on the same system. Since this function is implemented as a device, it allows the use of the standard device oriented system calls for interprocess communication, eliminating the need for a special set of system calls for this purpose. The IPM class characteristics are summarized in Table 5.9.

Table 5.9 - IPM class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of IPM devices allowed
MAXIMUM	DECV		Maximum number of in use IPM devices
NUMBER	DECV		Number of in use IPM devices

The following sections describe each of the IPM class characteristics in detail.

LIMIT - Maximum number of interprocess message devices allowed

This class characteristic specifies the maximum number of interprocess message devices which can be in use at any one time.

NUMBER - Number of in use interprocess message devices

This read only class characteristic returns the number of interprocess message devices currently in use in the system. This is the number of open interprocess message devices.

MAXIMUM - Maximum number of in use interprocess message devices

This class characteristic records the maximum number of interprocess message devices in use at any one time. This is the maximum number of simultaneously open interprocess message devices.

NULL Class Characteristics

The NULL device class includes only the NULL device. This is a dummy device which discards all output written to it and always indicates end of file on input. There are no class characteristics associated with the NULL device class.

PPR Class Characteristics

The PPR device class includes only the PPRn device. This is the generic parallel printer device which supports printers connected to a parallel port. The PPR class characteristics are summarized in Table 5.10

Table 5.10 - PPR class characteristics			
Name	**Format**	**Set**	**Description**
NUMBER	DECV		Number of parallel printer devices

The following sections describe each of the PPR class characteristics in detail.

NUMBER - Number of in use interprocess message devices

This read only class characteristic returns the number of parallel printer devices currently in use in the system.

NET Class Characteristics

The NET device class is the generic network device class. The NETn devices provide hardware level access to network interface. This device class also provides the foundation for the implementation of various network protocol stacks. Figure 5.1 show the general structure of the standard protocol stack provided with XOS. Additional protocols can be added at any level.

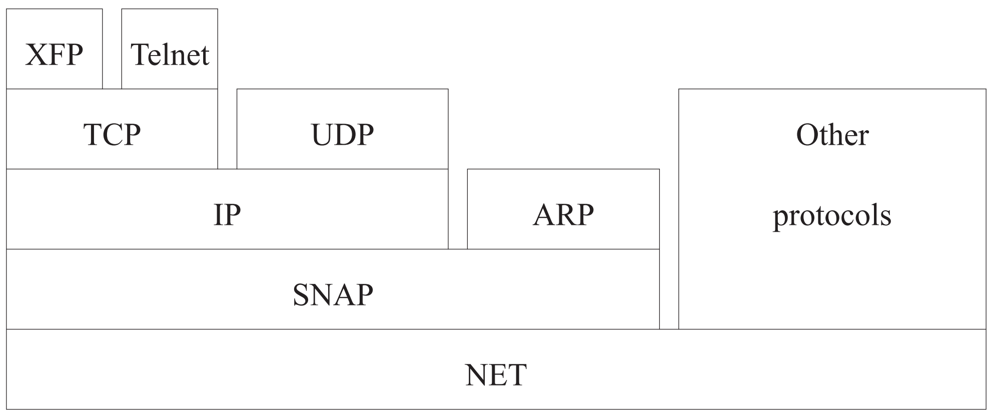

Figure 5.1 - Network protocol stacks

The class characteristics for this class specify values which affect the operation of the network interface as a whole. The NET class characteristics are summarized in Table 5.11.

Table 5.11 - NET class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of network devices allowed
MAXIMUM	DECV	s	Maximum number of in use network devices
NUMBER	DECV		Number of in use network devices

The following sections describe each of the NET class characteristics in detail.

LIMIT - Maximum number of network devices allowed

This class characteristic returns or sets the maximum number of network devices which can be in use at any one time.

NUMBER - Number of in use network devices

This class characteristic returns the number of network devices currently in use in the system. This is the number of open network devices.

MAXIMUM - Maximum number of in use network devices

This class characteristic returns the maximum number of network devices that have been in use at any one time. This is the maximum number of simultaneously open network devices.

SNAP Class Characteristics

The SNAP device class includes only the SNAPn devices. These devices provide network access at the link protocol level. This device supports the 802.2 unacknowledged connectionless service and the Bluebook or Ethernet II link level protocols. The SNAP class characteristics are summarized in Table 5.12.

Table 5.12 - SNAP class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of SNAP devices allowed
MAXIMUM	DECV	s	Maximum number of in use SNAP devices
NUMBER	DECV		Number of in use SNAP devices

The following sections describe each of the SNAP class characteristics in detail.

LIMIT - Maximum number of SNAP devices allowed

This class characteristic returns or sets the maximum number of SNAP devices which can be in use at any one time.

NUMBER - Number of in use SNAP devices

This class characteristic returns the number of SNAP devices currently in use in the system. This is the number of open SNAP devices.

MAXIMUM - Maximum number of in use SNAP devices

This class characteristic returns the maximum number of SNAP devices that have been in use at any one time. This is the maximum number of simultaneously open SNAP devices.

ARP Class Characteristics

The ARP device class includes only the ARPn devices. These devices provide network access at the ARP protocol level. ARP is a protocol which is used by other protocols to resolve physical LAN addresses. It cannot be used directly by a user program. ARPn devices cannot do I/O. They can be used only to return or set class or device characteristics values to monitor or control how the ARP protocol is being used. The ARP class characteristics are summarized in Table 5.13.

Table 5.13 - ARP class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of ARP devices allowed
MAXIMUM	DECV	s	Maximum number of in use ARP devices
NUMBER	DECV		Number of in use ARP devices

The following sections describe each of the ARP class characteristics in detail.

LIMIT - Maximum number of ARP devices allowed

This class characteristic returns or sets the maximum number of ARP devices which can be in use at any one time.

NUMBER - Number of in use ARP devices

This class characteristic returns the number of ARP devices currently in use in the system. This is the number of open ARP devices.

MAXIMUM - Maximum number of in use ARP devices

This class characteristic returns the maximum number of ARP devices that have been in use at any one time. This is the maximum number of simultaneously open ARP devices.

IPS Class Characteristics

The IPS device class includes only the IPSn devices. This is a network device which provides access to the IP protocol level. The IPS class characteristics are summarized in Table 5.14.

Table 5.14 - IPS class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of IPS devices allowed
MAXIMUM	DECV		Maximum number of in use IPS devices
NUMBER	DECV		Number of in use IPS devices

The following sections describe each of the IPS class characteristics in detail.

LIMIT - Maximum number of IPS devices allowed

This class characteristic returns or sets the maximum number of IPS devices which can be in use at any one time. The initial value is 10,000.

NUMBER - Number of in use IPS devices

This class characteristic returns the number of IPS devices currently in use in the system. This is the number of open IPS devices.

MAXIMUM - Maximum number of in use IPS devices

This class characteristic returns the maximum number of IPS devices that have been in use at any one time. This is the maximum number of simultaneously open IPS devices.

UDP Class Characteristics

The UDP device class includes only the UDPn devices. This is a network device which provides access to the UDP protocol level. The UDP class characteristics are summarized in Table 5.15.

Table 5.15 - IPM class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of UDP devices allowed
MAXIMUM	DECV		Maximum number of in use iUDP devices
NUMBER	DECV		Number of in use UDP devices

The following sections describe each of the UDP class characteristics in detail.

LIMIT - Maximum number of UDP devices allowed

This class characteristic returns or sets the maximum number of UDP devices which can be in use at any one time. The initial value is 10,000.

NUMBER - Number of in use UDP devices

This class characteristic returns the number of UDP devices currently in use in the system. This is the number of open UDP devices.

MAXIMUM - Maximum number of in use UDP devices

This class characteristic returns the maximum number of UDP devices that have been in use at any one time. This is the maximum number of simultaneously open UDP devices.

TCP Class Characteristics

The TCP device class includes only the TCPn devices. This is a network device which provides access to the TCP protocol level. The TCP class characteristics are summarized in Table 5.16.

Table 5.16 - TCP class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of TCP devices allowed
MAXIMUM	DECV		Maximum number of in use TCP devices
NUMBER	DECV		Number of in use TCP devices

The following sections describe each of the TCP class characteristics in detail.

LIMIT - Maximum number of TCP devices allowed

This class characteristic returns or sets the maximum number of TCP devices which can be in use at any one time. The initial value is 10,000.

NUMBER - Number of in use TCP devices

This class characteristic returns the number of TCP devices currently in use in the system. This is the number of open TCP devices.

MAXIMUM - Maximum number of in use interprocess message devices

This class characteristic returns the maximum number of TCP devices that have been in use at any one time. This is the maximum number of simultaneously open TCP devices.

TLN Class Characteristics

The TLN device class includes only the TLNn devices. This is a network device which implements a server for the Telnet protocol. These devices are not directly available to user programs to perform input or output. The TLNn device is useful only for accessing class and device characteristics to monitor or control the operation of the Telnet server. The actual I/O interface for the Telnet server is via a connection to the TRM class devices, where the Telnet server appears as a terminal interface driver.

The TLN class characteristics are summarized in Table 5.17.

Table 5.17 - TLN class characteristics

Name	Format	Set	Description
LIMIT	DECV	S	Maximum number of TLN devices allowed
MAXIMUM	DECV		Maximum number of in use TLN devices
NUMBER	DECV		Number of in use TLN devices

The following sections describe each of the TLN class characteristics in detail.

LIMIT - Maximum number of XFP devices allowed

This class characteristic returns or sets the maximum number of TLN devices which can be in use at any one time. The initial value is 10,000.

NUMBER - Number of in use TLN devices

This class characteristic returns the number of TLN devices currently in use in the system. This is the number of open TLN devices.

MAXIMUM - Maximum number of in use TLN devices

This class characteristic returns the maximum number of TLN devices which have been in use at any one time. This is the maximum number of simultaneously open TLN devices.

XFP Class Characteristics

The XFP device class includes only the XFPn devices. This is a network device which provides access to the XFP protocol level for remote file access. The XFP class characteristics are summarized in Table 5.18.

Table 5.18 - XFP class characteristics			
Name	**Format**	**Set**	**Description**
LIMIT	DECV	S	Maximum number of XFP devices allowed
MAXIMUM	DECV		Maximum number of in use XFP devices
NUMBER	DECV		Number of in use XFP devices

The following sections describe each of the XFP class characteristics in detail.

LIMIT - Maximum number of XFP devices allowed

This class characteristic returns or sets the maximum number of XFP devices which can be in use at any one time. The initial value is 10,000.

NUMBER - Number of in use XFP devices

This class characteristic returns the number of XFP devices currently in use in the system. This is the number of open XFP devices.

MAXIMUM - Maximum number of in use XFP devices

This class characteristic returns the maximum number of XFP devices which have been in use at any one time. This is the maximum number of simultaneously open XPF devices.

Chapter 6

Device Characteristics

This chapter describes the device characteristics for all standard devices in the system, including those implemented with loadable drivers. A device characteristic is a named item which specifies a value associated with a specific device. It is similar to a class characteristic except that a class characteristic represents a value associated with a device class as a whole, rather than with an individual device.

Device characteristics are mainly used to access permanent values, that is, items which retain their values when devices are opened and closed by programs. There are a few exceptions to this, where values are only effective until the next time the device is idle, and these are noted in the descriptions which follow.

Device characteristics provide a standardized interface which applies to all devices. It is a very flexible interface which is easily adaptable to virtually any new device type. Indeed, this degree of flexibility is such that it is difficult to summarize how device characteristics are used. This is best understood by reading the following descriptions of how individual XOS devices use device characteristics.

Device characteristic values are displayed or modified from the command level using the DEVCHAR command (see chapter 3, page 69).

This chapter lists the device characteristics by device class, and where necessary, by device type within a class. Technically, there is no relationship between characteristics for different devices which have the same name, but in practice all characteristics with the same name represent roughly equivalent values for their respective devices. There is one device characteristic that is defined for all devices (CLASS) and a small number that are defined for nearly all devices.

DISK Device Characteristics

Device characteristics for DISK class devices generally refer to the underlying physical disk or disk partition, even if the device is really a file on a file structured disk. There are a number of device characteristics which are used by all DISK class devices and there are a smaller number which are specific to individual types of disk (hard disks or floppy disks). The common characteristics for disks are summarized in Table 6.1.

Table 6.1 - Device characteristics for DISK class devices

Name	Format	Set	Description
AVAIL	DECV		Number of clusters available
CBLKSZ	DECV		Current block size
CCYLNS	DECV	S	Current number of cylinders
CHEADS	DECV	S	Current number of heads
CLASS	TEXT	V	Device class
CLSSZ	DECV		Cluster size
CLUSTERS	DECV		Total number of clusters
CSECTS	DECV	S	Current number of sectors
DBLKSZ	DECV	S	Default block size
DCYLNS	DECV	S	Default number of cylinders
DHEADS	DECV	S	Default number of heads
DOSNAME	TEXT	S	DOS name for disk
DSECTS	DECV	S	Default number of sectors
FATMODE	HEXV		FAT mode byte (DOS file system)
FSTYPE	TEXT		File structure type
INDEX	DECV		Index of unit on controller
IOREG	HEXV		Base I/O register number
MSENSOR	TEXT		Disk has media sensor
PARTN	HEXV		Partition number
PARTOFF	DECV		Partition offset
REMOVE	TEXT		Disk is removable
TYPE	TEXT		Device type
UNITTYPE	TEXT		Unit type
VOLNAME	TEXT	S	Volume name for disk

The common disk device characteristics are described in detail below.

AVAIL - Number of clusters available

This device characteristic returns the number of available clusters for a file structured disk. If the disk is not file structured, the value returned is always 0.

CBLKSZ - Current block size

This device characteristic returns or sets the current block or sector size, in bytes. File structured disks require a block size of 512 bytes, so this value cannot be changed for file structured disks or for disks accessed using the XOS disk cache. It can be changed, however, for non-file structured disks accessed in raw mode. This allows for the access of non-standard physical disk formats. This is particularly useful when reading certain non-DOS floppy disks. There are some limitations imposed by the disk controllers on block size. Controllers for hard disks usually do not allow the block size to be changed at all. The standard floppy controller requires a power of 2 between 128 and 4096. A value specified for this characteristic is in effect until a mount operation is done, when the value reverts back to the value of the DBLKSZ characteristic.

CCYLNS - Current number of cylinders

This device characteristic sets or returns the current number of cylinders on the disk. This value is set from the parameters of the file structure for file structured disks and cannot be changed for such disks. It can be changed for non-file structured disks. *This should be done with care since some disks can be physically damaged if repeatedly positioned beyond the last cylinder on the disk*. A value specified for this characteristic is in effect until a mount operation is done, when the value reverts back to the value of the DCYLNS characteristic.

CHEADS - Current number of heads

This device characteristic sets or returns the current number of heads on the disk. This value is set from the parameters of the file structure for file structured disks and cannot be changed for such disks. It can be changed for non-file structured disks. This is particularly useful when attempting to read a single sided floppy disk in a normal double sided floppy drive. A value specified for this characteristic is in effect until a mount operation is done, when the value reverts back to the value of the DHEADS characteristic.

CLASS - Device class

This device characteristic returns the name of the device class to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an error is indicated.

CLSSZ - Cluster size

This device characteristic returns the cluster size (in blocks) for a file structured disk. A cluster is the smallest amount of space allocated in the file system. For DOS and XOS file systems, it is always a power of 2, usually 1 to 8 blocks. A value of 0 is reported for non-file structured disks.

CLUSTERS - Total number of clusters

This device characteristic returns the total number of clusters on a file structured disk. A value of 0 is reported for non-file structured disks.

CSECTS - Current number of sectors

This device characteristic sets or returns the current number of sectors per track on the disk. This value is set from the parameters of the file structure for file structured disks and cannot be changed for such disks. It can be changed for non-file structured disks. A value specified for this characteristic is in effect until a mount operation is done, when the value reverts back to the value of the DSECTS characteristic.

DBLKSZ - Default block size

This device characteristic sets or returns the default block or sector size, in bytes. This value is only used when a disk is mounted. At this time it is copied to the value of the CBLKSZ characteristic to provide a default initial value. Note that for file structured disks, this value is immediately replaced by the value supplied by the parameters of the file system.

DCYLNS - Default number of cylinders

This device characteristic sets or returns the default number of cylinders. This value is only used when a disk is mounted. At this time it is copied to the value of the CCYLNS characteristic to provide a default initial value. Note that for file structured disks, this value is immediately replaced by the value supplied by the parameters of the file system.

DHEADS - Default number of heads

This device characteristic sets or returns the default number of heads. This value is only used when a disk is mounted. At this time it is copied to the value of the CHEADS characteristic to provide a default initial value. Note that for file structured disks, this value is immediately replaced by the value supplied by the parameters of the file system.

DOSNAME - DOS name for disk

This device characteristic sets or returns the DOS name for the disk. This name provides an alternate name for the disk. Normally this is used to specify the equivalent DOS disk name (a single letter) for DOS compatibility. When a program requests that the DOS name of a device be returned, this name is returned. *Setting this characteristic to a value which is other than a single letter will cause incorrect operation when running DOS programs*.

DSECTS - Default number of sectors

This device characteristic sets or returns the default number of sectors. This value is only used when a disk is mounted. At this time it is copied to the value of the CSECTS characteristic to provide a default initial value. Note that for file structured disks, this value is immediately replaced by the value supplied by the parameters of the file system.

FATMODE - FAT mode byte (DOS file system)

This device characteristic returns the value of the FAT mode byte for DOS file systems. A value of 0 is returned for disks which do not contain a DOS file system.

FSTYPE - File structure type

This device characteristic returns the type of file system mounted on the device. Possible values are:

Value	Meaning
DOS12	DOS file system with 12 bit FATs
DOS16	DOS file system with 16 bit FATs
DOS16L	Huge DOS file system (DOS version 4 and above)
XOS	XOS native file system

INDEX - Index of unit on controller

This device characteristic returns the index of a device on its controller. This value cannot be changed after the device is created.

IOREG - Base I/O register number

This device characteristic returns the base I/O register for the device. This value cannot be changed after the device is created.

MSENSOR - Disk has media sensor

This device characteristic sets or returns a value which indicates if the disk has a media sensor. This is only meaningful for removable disks. Possible values are YES and NO. Fixed disks always return NO.

PARTN - Partition number

This device characteristic returns the partition number for a hard disk partition. This is the index (1 based) for the partition table entry for the partition. If the partition is in an extended partition, the index for the entry of the extended partition ORED with 0x80 is returned. If the device is not a partition, a value of 0 is returned.

PARTOFF - Partition offset

This device characteristic returns the block offset for a disk partition. This is the block number on the underlying disk for the first block of the partition.

REMOVE - Disk is removable

This device characteristic sets or returns a value that specifies if a disk is removable or fixed. Possible values are YES (removable) and NO (fixed). The value of this characteristic cannot be changed for hard disks and is always NO.

TYPE - Device type

This read-only characteristic returns the type of the device. The value returned will be HDKA, HDKB, or HDKC for hard disk or FDKA for floppy disks.

UNITTYPE - Unit type

This device characteristic specifies the type of a disk unit. For hard disks, the value of this characteristic is always HARD and cannot be changed. Valid values for floppy disks are:

Value	Meaning
HD3	3.5" high density
DD3	3.5" double density
HD5	5.25" high density
DD5	5.25" double density
DD8	8" double density

VOLNAME - Volume name for disk

This device characteristic specifies an alternate name for a disk. The current version of XOS does not set this name. A future version will probably set this name to be the volume name specified for a disk, especially for XOS file system disks.

HDKA (PC-AT hard disk) Disk Device Characteristics

This section describes the device characteristics which are specific to the HDKA type disk device. This device supports the standard PC-AT hard disk controller. This includes the original ST-501 controller, ESDI controllers, and IDE controllers. It does not include the PS-2 disk controllers or most SCSI controllers. SCSI controllers which fully emulate the PC-AT hard disk controller registers in hardware can be used, but none of the extra features of these controllers are supported by this device.

Hard disks use all of the common DISK class device characteristics listed previously plus the characteristics described here. Table 6.2 summarizes the hard disk specific device characteristics.

Table 6.2 - Device characteristics for HDKA type disks

Name	Format	Set	Description
BUFSIZE	DECV		Size of internal disk buffer
FWVER	STR		Disk firmware version
MODELNO	STR		Disk model number
SECPINT	DECV		Maximum sectors per interrupt
SERIALNO	STR		Disk serial number

The hard disk specific device characteristics are described in detail below.

BUFSIZE - Size of internal disk buffer

This device characteristic returns the length of the hard disk controller's internal data buffer in bytes. If the controller is not buffered or does not report the length of its buffer, a value of 0 is returned.

FWVER - Disk firmware version

This device characteristic returns the hard disk controller's firmware version number. If the controller does not report this information, a null string is returned.

MODELNO - Disk model number

This device characteristic returns the hard disk controller's model number. If the controller does not report this information, a null string is returned.

SECPINT - Maximum sectors per interrupt

This device characteristic returns the maximum number of sectors which the hard disk controller can transfer with a single interrupt. If the controller is not capable of multi-sector per interrupt operation or it does not report this information, a value of 0 is returned.

SERIALNO - Disk serial number

This device characteristic returns the hard disk controller's serial number. If the controller does not report this information, a null string is returned.

pecific device characteristics are described in detail below.

FDKA (floppy disk) Disk Device Characteristics

This section describes the device characteristics which are specific to the FDKA type disk device. This device supports the standard PC-AT floppy disk controller (NEC 765/Intel 8272) and the CompatiCard-I add-in floppy disk controller made by Micro Solutions, Inc. This controller uses the same chip as the standard PC-AT controller but provides slightly more flexibility in supporting non-standard floppy types, especially 8" floppies.

Floppy disks use all of the common DISK class device characteristics listed previously plus the characteristics described here. Table 6.5 summarizes the floppy disk specific device characteristics.

Table 6.5 - Device characteristics for floppy disks

Name	Format	Set	Description
CONDESP	TEXT		Controller description
DATADEN	TEXT	S	Data density
HLTIME	DECV	S	Head load time in milliseconds
HUTIME	DECV	S	Head unload time in milliseconds
MOTIME	DECV	S	Motor off time in seconds
MSTIME	DECV	S	Motor start time in milliseconds
SRTIME	DECV	S	Step rate timing in milliseconds
TRKDEN	DECV	S	Track density
XGAPLEN	DECV	S	Gap length for transfers

The floppy disk specific device characteristics are described in detail below.

CONDESP - Controller description

This device characteristic returns the description of the floppy disk controller. Possible valid values are:

Value	Meaning
PCAT	Standard AT floppy controller
CMPT	CompatiCard floppy controller

DATADEN - Data density

This device characteristic set or returns the data density for the floppy disk drive. Valid values are:

Value	Meaning
SINGLE	Single density floppy disk
DOUBLE	Double density floppy disk
HIGH	High density floppy disk

Either the entire value string or just the first letter can be specified.

HLTIME - Head load time

This device characteristic sets or returns the head load time (in milliseconds) for the floppy disk.

HUTIME - Head unload time

This device characteristic sets or returns the head unload time (in milliseconds) for the floppy disk.

MOTIME - Motor off time

This device characteristic sets or returns the motor off time (in seconds) for the floppy disk.

MSTIME - Motor start time

This device characteristic sets or returns the motor start time (in milliseconds) for the floppy disk.

SRTIME - Step rate timing

This device characteristic sets or returns the step rate (in milliseconds) for the floppy disk.

TRKDEN - Track density

This device characteristic sets or returns the track density for the floppy disk drive. For 3.5" disks this value is always 135 and cannot be changed. For 5.25" disks this value may be 48 or 96.

XGAPLEN - Gap length for transfers

This device characteristic sets or returns the interrecord gap value used with data transfers for the floppy disk.

SPL Device Characteristics

Table 6.6 - Device characteristics for SPL class serial ports			
Name	**Format**	**Set**	**Description**
CLASS	TEXT	V	Device class
CLSNAME	STR	S	File name for closed file
CLSMSG	TEXT	S	Destination for close message
CLSTIME	DECV	S	Inactive close time-out value (seconds)
SEQNUM	DECV	S	File name sequence number
tSPLSPEC	STR	S	File specification for spooled file

The following section describes the device characteristics for SPL class devices in detail.

CLASS - Device class

This device characteristic returns the name of the device class (always SPL here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an error is indicated.

CLSNAME - File name for closed file

When a spooled file is closed, it is automatically renamed to the name specified by this device characteristic. The name or extension must contain the # character, which indicates that the current value of the spooled name sequence number is to be inserted. The # must be followed by a single digit which specifies the number of digits to insert. For example, if the CLSNAME value is S01_#4.SPL and the SEQNUM value is 52, the final name for the closed file will be S01_0052.SPL.

CLSMSG - Destination for close message

When a spooled file is closed, an IPM (Interprocess Message Device) message is sent to the destination specified by the value of this device characteristic. This will normally be the IPM name used by the UNSPOOL symbiont which is unspooling the spooled device.

CLSTIME - Inactive close time-out value (seconds)

This device characteristic specifies the maximum idle time (in seconds) for a spooled file. A spooled file is closed automatically if no output is done to the file for this interval. A value of 0 disables the automatic close feature.

SEQNUM - File name sequence number

This device characteristic specifies the current spool name sequence number. This number is used when generating both the initial spooled file name and final closed spooled name. It is incremented by 1 whenever it is used.

SPLSPEC - File specification for spooled file

This device characteristic specifies the name which is used when opening a spooled file. Its value must be a complete file specification including a device name, directory path, and name and extension. The device specified must be a mass storage device. It can be either local or remote. The name or extension must contain the # character, which indicates that the current value of the spooled name sequence number is to be inserted. The # must be followed by a single digit which specifies the number of digits to insert. For example, if the SPLSPEC value is C:\SPOOL\S01_#4.TMP and the SEQNUM value is 193, the initial spooled file will be C:\SPOOL\S01_0193.TMP.

TRM (Serial Port) Device Characteristics

This section describes the device characteristics which are specific to TRM class serial port devices. This includes serial ports using the SERA driver (standard AT serial ports) and ports using the SERB driver (DigiBoard multi-port serial interface). Table 6.7 summarizes the TRM class serial port device characteristics.

Table 6.7 - Device characteristics for TRM class serial ports

Name	Format	Set	Description
CLASS	TEXT	V	Device class
DBITS	DECV	S	Current number of data bits
IDBITS	DECV	S	Initial number of data bits
IINFLOW	TEXT	S	Initial input flow control
IINRATE	DECV	S	Initial input baud rate
IMODEM	TEXT	S	Initial modem control
INDEX	DECV		Port number on board (SERB only)
INFLOW	TEXT	S	Current input flow control
INLBS	DECV		Output ring buffer size
INRATE	DECV	S	Current input baud rate (not used)
INRBS	DECV		Input ring buffer size
INT	DECV		Interrupt level
INTRBS	DECV		Interrupt ring buffer size
IOREG	HEXV		Base I/O register number
IOUTFLOW	TEXT	S	Initial output flow control
IOUTRATE	DECV	S	Initial output baud rate
IPARITY	TEXT	S	Initial parity handling
IRATE	DECV	S	Initial baud rate
ISBITS	DECV	S	Initial number of stop bits
MODEM	TEXT	S	Current modem control
OUTFLOW	TEXT	S	Current output flow control
OUTRATE	DECV	S	Current output baud rate
OUTRBS	DECV		Input line buffer size
PARITY	TEXT	S	Current parity handling
RATE	DECV	S	Current baud rate
RATEDET	DECV	S	Baud rate detect type
SBITS	DECV	S	Current number of stop bits
SESSION	TEXT	S	Allow login on port
STSREG	HEXV		Status I/O register number (SERB only)
TYPE	TEXT		Device type

The following section describes the device characteristics for TRM class serial port devices in detail.

CLASS - Device class

This device characteristic returns the name of the device class (always TRM here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an error is indicated.

DBITS - Current number of data bits

This device characteristic sets or returns the current number of data bits in each character sent or received on the serial port. A value specified for this characteristic takes effect immediately and is in effect until the device is opened after being idle. At this time the value of this characteristic is initialized from the value of the IDBITS characteristic. The value must be between 5 and 8. Values less than 5 are taken as 5 and those greater than 8 are taken as 8.

IDBITS - Initial number of data bits

This device characteristic sets or returns the initial number of data bits in each character sent or received on the serial port. A value specified for this characteristic is not used directly but is used to initialize the value of the DBITS characteristic when the device is opened after being idle. The value must be between 5 and 8. Values less than 5 are taken as 5 and those greater than 8 are taken as 8.

IINFLOW - Initial input flow control

This device characteristic sets or returns the initial input flow control handling. A value specified for this characteristic is not used directly but is used to initialize the value of the INFLOW characteristic when the device is opened after being idle.

Value	Meaning
DSRDTR	Hardware flow control using DSR and DTR
DSR	Same as DSRDTR
CTSRTS	Hardware flow control using CTS and RTS
CTS	Same as CTSRTS
REVCTS	Same as CTSRTS but with reversed sense
REV	Same REVCTS
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

IINRATE - Initial input baud rate

This device characteristic sets or returns the initial input baud rate for the serial port. Since SERA and SERB devices do not support separate input and output baud rate specifications, setting the value of this characteristic has no effect.

IMODEM - Initial modem control

This device characteristic sets or returns the initial modem control handling state for the serial port. A value specified for this characteristic is not used directly but is used to initialize the value of the MODEM characteristic when the device is opened after being idle. Valid values for this characteristic are YES (port uses modem control features) and NO (port does not use modem control features).

INDEX - Port number on board (SERB only)

This device characteristic returns the port number on the interface board for SERB type serial devices.

INFLOW - Current input flow control

This device characteristic sets or returns the current input flow control handling state. The value specified for this parameter takes effect immediately and is in effect until the next time the device is opened while idle. At this time, the value of this characteristic is initialized to be the same as the value of the IINFLOW characteristic. The valid values for this characteristic are:

Value	Meaning
DSRDTR	Hardware flow control using DSR and DTR
DSR	Same as DSRDTR
CTSRTS	Hardware flow control using CTS and RTS
CTS	Same as CTSRTS
REVCTS	Same as CTSRTS but with reversed sense
REV	Same REVCTS
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

INLBS - Output ring buffer size

This device characteristic returns the size of the output ring buffer for the serial port. This buffer is allocated when the device is created and is used to buffer all data output to the port.

INRATE - Current input baud rate

This device characteristic specifies the current input baud rate for the serial port. Since SERA and SERB devices do not support separate input and output baud rate specifications, setting the value of this characteristic has no effect.

INRBS - Input ring buffer size

This device characteristic returns the size of the input ring buffer for the serial port. This buffer is allocated when the device is created and is used to type ahead data.

INT - Interrupt level

This device characteristic returns the number of the hardware interrupt (IRQ) being used by the serial port.

INTRBS - Interrupt ring buffer size

This device characteristic returns the size of the interrupt ring buffer for the serial port. This buffer is allocated when the device is created and is used to buffer input data at interrupt level.

IOREG - Base I/O register number

This device characteristic returns the base I/O register number for the serial interface.

IOUTFLOW - Initial output flow control

This device characteristic sets or returns the initial output flow control handling state. A value specified for this characteristic is not used directly but is used to initialize the value of the OUTFLOW characteristic when the device is opened after being idle. The valid values for this characteristic are:

Value	Meaning
DSRDTR	Hardware flow control using DSR and DTR
DSR	Same as DSRDTR
CTSRTS	Hardware flow control using CTS and RTS
CTS	Same as CTSRTS
REVCTS	Same as CTSRTS but with reversed sense
REV	Same REVCTS
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

IOUTRATE - Initial output baud rate

This device characteristic sets or returns the initial output baud rate for the serial port. Since SERA and SERB devices do not support different input and output baud rates, this characteristic is equivalent to the IRATE characteristic.

IPARITY - Initial parity handling

This device characteristic sets or returns the initial output parity handling state for the serial port. A value specified for this characteristic is not used directly but is used to initialize the value of the PARITY characteristic when the device is opened after being idle. Valid values for this characteristic are:

Value	Meaning
NONE	No parity bit is added
MARK	A marking bit is always added
SPACE	A spacing bit is always added
ODD	A bit is added to generate odd parity
EVEN	A bit is added to generate even parity

IRATE - Initial baud rate

This device characteristic sets or returns the initial baud rate for the serial port. A value specified for this characteristic is not used directly but is used to initialize the value of the RATE characteristic when the device is opened after being idle. The baud rate value may be any number. If the value is less than the lowest baud rate supported by the interface, that rate is used; otherwise the system selects the highest available baud rate which is not greater than the value specified.

ISBITS - Initial number of stop bits

This device characteristic sets or returns the initial number of stop bits for the serial port. A value specified for this characteristic is not used directly but is used to initialize the value of the SBITS characteristic when the device is opened after being idle. The value must be 1 or 2. Values less than 1 are taken as 1 and those greater than 2 are taken as 2.

MODEM - Current modem control

This device characteristic sets or returns the current modem control handling state for the serial port. A value specified for this characteristic takes effect immediately and stays in effect until the next time the device is opened while idle. At this time the value of this characteristic is initialized from the value of the IMODEM characteristic. Valid values for this characteristic are YES (port uses modem control features) and NO (port does not use modem control features).

OUTFLOW - Current output flow control

This device characteristic sets or returns the current output flow control handling state. The value specified for this parameter takes effect immediately and is in effect until the next time the device is opened while idle. At this time, the value of this characteristic is initialized to be the same as the value of the IOUTFLOW characteristic. The valid values for this characteristic are:

Value	Meaning
DSRDTR	Hardware flow control using DSR and DTR
DSR	Same as DSRDTR
CTSRTS	Hardware flow control using CTS and RTS
CTS	Same as CTSRTS
REVCTS	Same as CTSRTS but with reversed sense
REV	Same REVCTS
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

OUTRATE - Current output baud rate

This device characteristic sets or returns the current output baud rate for the serial port. Since SERA and SERB devices do not support different input and output baud rates, this characteristic is equivalent to the RATE characteristic.

OUTRBS - Input line buffer size

This device characteristic returns the size of the port's input line buffer. This buffer is allocated when the device is created and is used to buffer line mode input.

PARITY - Current parity handling

This device characteristic sets or returns the current output parity handling state for the serial port. A value specified for this characteristic takes effect immediately and stays in effect until the next time the device is opened while idle. At this time the value of this characteristic is initialized from the value of the IPARITY characteristic.

Value	Meaning
NONE	No parity bit is added
MARK	A marking bit is always added
SPACE	A spacing bit is always added
ODD	A bit is added to generate odd parity
EVEN	A bit is added to generate even parity

RATE - Current baud rate

This device characteristic sets or returns the current baud rate for the serial port. A value specified for this characteristic takes effect immediately and stays in effect until the next time the device is opened while idle. At this time the value of this characteristic is initialized from the value of the IRATE characteristic. The baud rate value may be any number. If the value is less than the lowest baud rate supported by the interface, that rate is used; otherwise the system selects the highest available baud rate which is not greater than the value specified.

RATEDET - Baud rate detect type

This device characteristic sets or returns the baud-rate detect method to be used for automatic baud-rate determination. This feature is not fully implemented in the current version of XOS.

SBITS - Current number of stop bits

This device characteristic sets or returns the current number of stop bits for the serial port. A value specified for this characteristic takes effect immediately and stays in effect until the next time the device is opened while idle. At this time the value of this characteristic is initialized from the value of the ISBITS characteristic. The value must be 1 or 2. Values less than 1 are taken as 1 and those greater than 2 are taken as 2.

SESSION - Allow port to control session

This device characteristic sets or returns a value which specifies if the port can be used to log-in a user session. Normally, this characteristic will have the value YES or NO. If the value is YES, any input from the serial port while the port is idle causes the system to create a new process and run a command shell with the port as the controlling terminal. If the value is NO, input from the serial port while it is idle is ignored. The value may also be set to any sequence of characters beginning with an underscore character. In this case, the system will send an IPM message to the IPM name formed by removing the underscore from the characteristic value whenever there is input from the serial port and it is idle. The user must have a program running which has opened an IPM device with this name and is prepared to receive the messages. This is intended to provide a method of implementing non-standard terminal based systems.

STSREG - Status I/O register number (SERB only)

This device characteristic returns the I/O register number of the DigiBoard status register. It is only valid for SERB type devices. Each DigiBoard has a single status register, regardless of the number of ports which it supports. The number of this status register is used by the system to identify the board (as opposed to the individual ports).

TYPE - Device type

This device characteristic returns the type of the device. The value returned will be SERA (standard serial port) or SERB (DigiBoard) for serial ports.

TRM (Console) Device Characteristics

This section describes the device characteristics which are specific to TRM class console devices. This includes consoles using the MGAA (Hercules compatible monochrome graphics adapter), EGAA (Enhanced Graphics Adapter and compatibles), and VGAA (Virtual Graphics Array and compatibles) type devices. Table 6.8 summarizes the TRM class console device characteristics.

Table 6.8 - Device characteristics for TRM class consoles

Name	Format	Set	Description
BELLFREQ	DECV	S	Bell tone frequency
BELLLEN	DECV	S	Bell tone length
CHARIN	DECV	S	Number of characters input
TCHAROUT	DECV	S	Number of characters output
CLASS	TEXT	V	Device class
CURFIX	TEXT	S	Special cursor fix-up enable
INLBS	DECV		Input line buffer size
INRBHELD	DECV	S	Number of times input held for flow control
INRBLOST	DECV	S	Number of input character discarded
INRBPL	DECV	S	Input ring buffer panic level
INRBS	DECV		Input ring buffer size
INRBSL	DECV	S	Input ring buffer stop level
IOUTFLOW	TEXT	S	Initial output flow control
KBCHAR	DECV	S	Number of keyboard scan codes input
KBTCHAR	DECV	S	Total number of keyboard scan codes input
OUTFLOW	TEXT	S	Current output flow control
PASSWORD	STR	S	System level password for console
PROGRAM	STR	S	Initial program to run
SESSION	TEXT	S	Allow session on console
SCSVTIME	DECV	S	Screen saver time (seconds)
XSCSVTYPE	TEXT	S	Screen saver type
TYPE	TEXT		Device type

The following section describes the device characteristics for TRM class console devices in detail.

BELLFREQ - Bell tone frequency

This device characteristic sets or returns the frequency (in Hertz) for the bell tone associated with the console.

BELLLEN - Bell tone length

This device characteristic sets or returns the length (in milliseconds) of the bell tone generated when a BELL (Cntl-G) character is output.

CHARIN - Number of characters input

This device characteristic records the number of characters input by the device.

CHAROUT - Number of characters output

This device characteristic records the number of characters output by the device.

CLASS - Device class

This device characteristic returns the name of the device class (always TRM here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device-dependent functions which only behave as expected for a specific class device.

CURFIX - Special cursor fix-up enable

This device characteristic specifies if the special cursor fix-up mode is enabled. Some display adapters do not correctly display a full block cursor. When the special fix-up mode is enabled, full block cursors are reduced by one scan line. Valid values are YES (indicating that the special fix-up mode is enabled) and NO (indicating that the special fix-up mode is disabled).

INLBS

This device characteristic returns the size of the input line buffer for the device. This value is specified when the console device is created and cannot be changed after that.

INRBHELD

This device characteristic records the number of times that input flow control has been held of input because the input ring buffer was nearly full (see description of the INRBSL device characteristic).

INRBLOST

This device characteristic records the number of input characters which have been discarded because the input ring buffer was full.

INRBPL

This device characteristic specifies the level at which panic mode is used to hold off input. This is only relevant when using XON/XOFF flow control. When the input buffer fulls up to the level specified by the INRBSL characteristic a single XOFF character is output to attempt to stop input. No additional response is made until the number of character positions available is less than the level specified by the value of this parameter. At this point, an XOFF character is output for every input character received. A value of 0 for this characteristic disables the panic mode response.

INRBS

This device characteristic returns the size of the input ring buffer for the device. This value is specified when the console device is created and cannot be changed after that. The input ring buffer is used to buffer type-ahead.

INRBSL - Input ring buffer stop level

This device characteristic specifies the level at which input is held off. The level is specified as the number of character positions available at which point input is to be held off.

IOUTFLOW - Initial output flow control

This device characteristic specifies the initial output flow control handling. A value specified for this characteristic is not used directly but is used to initialize the value of the OUTFLOW characteristic when the device is opened after being idle. The valid values for this characteristic are:

Value	Meaning
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

KBCHAR - Number of keyboard scan codes input

This device characteristic records the number of scan codes generated by the console keyboard for this device. This includes both make and break codes.

KBTCHAR - Total number of keyboard scan codes input

This device characteristic records the total number of scan codes generated by the console keyboard for all devices which can be associated with it. This includes both make and break codes.

OUTFLOW - Current output flow control

This device characteristic specifies the current output flow control handling state. The value specified for this parameter takes effect immediately and is in effect until the next time the device is opened while idle. At this time, the value of this characteristic is initialized to be the same as the value of the IOUTFLOW characteristic. The valid values for this characteristic are:

Value	**Meaning**
XONXOFF	Flow control using XON and XOFF characters
XON	Same as XONXOFF
NONE	No flow control

PASSWORD - System level password for console

This device characteristic specifies the system level password associated with the terminal device. If a password is specified, it must be entered whenever a session is started on the terminal.

PROGRAM - Initial program to run

This device characteristic specifies the name of the program to run initially when a session is started on the terminal. If no name is specified, either SHELL.IMG or LOGIN.IMG is run, depending on whether user logins are enabled for the system. Only a name can be specified. The program must be in the directory specified by the XOSSYS: logical name.

SESSION - Allow session on console

This device characteristic specifies if the port can be used to control a user session. Normally, this characteristic will have the value YES or NO. If the value is YES, any input from the serial port while the port is idle causes the system to create a new process and run a command shell (or the login program if user login is enabled for the system) with the port as the controlling terminal. If the value is NO, input from the serial port while it is idle is ignored. The value may also be set to any sequence of characters beginning with an underscore character. In this case, the system will send an IPM message to the IPM name formed by removing the underscore from the characteristic value whenever there is input from the serial port and it is idle. The user must have a program running which has opened an IPM device with this name and is prepared to receive the messages. This is intended to provide a method of implementing non-standard terminal based systems.

SCSVTIME - Screen saver time

This device characteristic specifies the time (in seconds) for the screen saver function. The screen is blanked if there is no keyboard activity for this time interval. It is unblanked as soon as any keyboard activity occurs. A value of 0 disables the screen saver function.

SCSVTYPE - Screen saver type

This device characteristic specifies the behavior of the screen saver function. A value of K or KEY specifies that only keyboard activity will be considered when blanking or unblanking the screen. A value of F or FULL specifies that program output to the screen will also be considered.

TYPE - Console type

This read-only device characteristic returns the console driver type. It will be one of MGAA, VGAA, or EGAA.

TRM (Pseudo-Console) Device Characteristics

This section describes the device characteristics which are specific to TRM class devices which are the client side of a PCN (pseudo-console) device. These devices are true TRM class devices although they do not correspond to a physical terminal device. They are used by various servers (specifically by the Telnet server) to emulate a real terminal. Table 6.9 summarizes the TRM class console device characteristics.

Table 6.9 - Device characteristics for TRM class consoles

Name	Format	Set	Description
CLASS	TEXT	V	Device class
INLBS	DECV		Input line buffer size
INRBS	DECV		Input ring buffer size
PASSWORD	STR	S	System level password for terminal
PROGRAM	STR	S	Initial program to run
SESSION	TEXT	S	Allow session on console
TYPE	TEXT		Device type

The following section describes the device characteristics for TRM class pseudo-console devices in detail.

CLASS

This device characteristic returns the name of the device class (always TRM here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

INLBS - Input line buffer size

This device characteristic returns the size of the input line buffer for the device. This value is specified when the console device is created and cannot be changed after that.

INRBS - Input ring buffer size

This device characteristic returns the size of the input ring buffer for the device. This value is specified when the console device is created and cannot be changed after that. The input ring buffer is used to buffer type-ahead.

PASSWORD - System level password for terminal

This device characteristic specifies the system level password associated with the terminal device. If a password is specified, it must be entered whenever a session is started on the terminal.

PROGRAM - Initial program to run

This device characteristic specifies the name of the program to run initially when a session is started on the terminal. If no name is specified, either SHELL.IMG or LOGIN.IMG is run, depending on whether user logins are enabled for the system. Only a name can be specified. The program must be in the directory specified by the XOSSYS: logical name.

SESSION - Allow session on console

This device characteristic specifies if the port can be used to control a user session. Normally, this characteristic will have the value YES or NO. If the value is YES, any input from the serial port while the port is idle causes the system to create a new process and run a command shell (or the login program if user login is enabled for the system) with the port as the controlling terminal. If the value is NO, input from the serial port while it is idle is ignored. The value may also be set to any sequence of characters beginning with an underscore character. In this case, the system will send an IPM message to the IPM name formed by removing the underscore from the characteristic value whenever there is input from the serial port and it is idle. The user must have a program running which has opened an IPM device with this name and is prepared to receive the messages. This is intended to provide a method of implementing non-standard terminal based systems.

TYPE - Device type

This read only device characteristic returns the console driver type. It will always be PCN.

TRM (Telnet) Device Characteristics

This section describes the device characteristics which are specific to TRM class devices which are the client side of a PCN (pseudo-console) device. These tln devices are true TRM class devices although they do not correspond to a physical terminal device. They are used by various servers (specifically by the Telnet server) to emulate a real terminal. Table 6.10 summarizes the TRM class console device characteristics.

Table 6.10 - Device characteristics for TRM class consoles

Name	Format	Set	Description
CLASS	TEXT	V	Device class
INLBS	DECV		Input line buffer size
INRBS	DECV		Input ring buffer size
PASSWORD	STR	S	System level password for terminal
PROGRAM	STR	S	Initial program to run
SESSION	TEXT	S	Allow session on console
TYPE	TEXT		Device type

The following section describes the device characteristics for TRM class pseudo-console devices in detail.

CLASS

This device characteristic returns the name of the device class (always TRM here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

INLBS - Input line buffer size

This device characteristic returns the size of the input line buffer for the device. This value is specified when the console device is created and cannot be changed after that.

INRBS - Input ring buffer size

This device characteristic returns the size of the input ring buffer for the device. This value is specified when the console device is created and cannot be changed after that. The input ring buffer is used to buffer type-ahead.

PASSWORD - System level password for terminal

This device characteristic specifies the system level password associated with the terminal device. If a password is specified, it must be entered whenever a session is started on the terminal.

PROGRAM - Initial program to run

This device characteristic specifies the name of the program to run initially when a session is started on the terminal. If no name is specified, either SHELL.IMG or LOGIN.IMG is run, depending on whether user logins are enabled for the system. Only a name can be specified. The program must be in the directory specified by the XOSSYS: logical name.

SESSION - Allow session on terminal

This device characteristic specifies if the port can be used to control a user session. Normally, this characteristic will have the value YES or NO. If the value is YES, any input from the serial port while the port is idle causes the system to create a new process and run a command shell (or the login program if user login is enabled for the system) with the port as the controlling terminal. If the value is NO, input from the serial port while it is idle is ignored. The value may also be set to any sequence of characters beginning with an underscore character. In this case, the system will send an IPM message to the IPM name formed by removing the underscore from the characteristic value whenever there is input from the serial port and it is idle. The user must have a program running which has opened an IPM device with this name and is prepared to receive the messages. This is intended to provide a method of implementing non-standard terminal based systems.

TYPE - Device type

This read only device characteristic returns the console driver type. It will always be PCN.

PCN Device Characteristics

This section describes the device characteristics for PCN class devices. These are summarized in Table 6.11.

Table 6.11 - Device Characteristics for PCN Class Devices

Name	Format	Set	Description
CLASS	TEXT	V	Device class
INLBS	DECV	S	Input line buffer size
INRBS	DECV	S	Input ring buffer size
PASSWORD	STR	S	System level password
PROGRAM	STR	S	Initial program to run
SESSION	TEXT	S	Allow session on terminal

Following is a detailed description of the device characteristic defined for the PCN device class.

CLASS - Device class

This device characteristic returns the name of the device class (always PCN here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

INLBS - Input line buffer size

This device characteristic specifies the size of the line buffer which is allocated when a PCN device is opened. This will be the value returned by the INLBS device characteristic for the terminal class device associated with the PCN device.

INRBS - Input ring buffer size

This device characteristic specifies the size of the input ring buffer which is allocated when a PCN device is opened. This buffer is used to hold type-ahead data. This will be the value returned by the INRBS device characteristic for the terminal class device associated with the PCN device.

PASSWORD - System level password

This device characteristic specifies the initial value for the PASSWORD device characteristic for TRM class devices associated with the PCN device.

PROGRAM - Initial program to run

This device characteristic specifies the initial value for the PROGRAM device characteristic for TRM class devices associated with the PCN device.

SESSION - Allow session on terminal

This device characteristic specifies the initial value for the SESSION device characteristic for TRM class devices associated with the PCN device.

IPM Device Characteristics

This section describes the device characteristics for IPM class devices. These are summarized in Table 6.12.

Table 6.12 - Device Characteristics for IPM Class Devices

Name	Format	Set	Description
CLASS	TEXT	V	Device class

Following is a detailed description of the device characteristic defined for the IPM device class.

CLASS - Device class

This device characteristic returns the name of the device class (always IPM here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

NULL Device Characteristics

This section describes the device characteristics for NULL class devices. These are summarized in Table 6.13.

Table 6.13 - Device Characteristics for NULL Class Devices

Name	Format	Set	Description
CLASS	TEXT	V	Device class

Following is a detailed description of the device characteristic defined for the NULL device class.

CLASS - Device class

This device characteristic returns the name of the device class (always NULL here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

PPR Device Characteristics

This section describes the device characteristics for PPR class devices. These are summarized in Table 6.14.

Table 6.14 - Device Characteristics for PPR Class Devices

Name	Format	Set	Description
CLASS	TEXT	V	Device class
IOREG	HEXV		Base I/O register number
INT	DECV		Interrupt number
TIMEOUT	HEXV		Default time-out value

Following is a detailed description of the device characteristic defined for the PPR device class.

CLASS - Device class

This device characteristic returns the name of the device class (always PPR here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

IOREG - Base I/O register number

This device characteristic returns the base I/O register for the device. This value cannot be changed after the device is created.

INT - Interrupt number

This device characteristic returns the interrupt number for the device. This value cannot be changed after the device is created.

TIMEOUT - Default time-out value

This device characteristic specifies the default time-out value in seconds. If no time out value is specified when doing output to the PPR device, this value is used to determine when to return a “ no response” error. A value of 0 indicates no time out.

NET Device Characteristics

This section describes the device characteristics which are specific to NET class devices. The NET devices are the low level network interfaces. These devices can be used to send and receive raw network data, but are most useful for obtaining statistics about the operation of the network using the device characteristics described here. Even though multiple NET DCBs are created as needed to allow multiple processes to access the network, the various error and usage counters accessed with these device characteristics reflect the total usage of the network interface.

This section first describes the device characteristics which are used by all NET class devices. This is followed by descriptions of the device characteristics which are specific to each type of NET device.

Table 6.15 summarizes the common device characteristics for the NET class devices.

Table 6.15 - Device characteristics for NET class devices

Name	Format	Set	Description
BADPNT	DECV	S	Number of packets discarded because of bad ring pointer
BCPKTIN	DECV	S	Number of broadcast packets input
BYTEIN	DECV	S	Number of bytes input
BYTEOUT	DECV	S	Number of bytes output
CLASS	TEXT	V	Device class
ICRC	DECV	S	Number of input CRC errors
IFRAME	DECV	S	Number of input framing errors
ILOST	DECV	S	Number of lost input packets
INT	DECV		Interrupt level
IOREG	HEXV		Base I/O register number
IOVRRN	DECV	S	Number of input overrun errors
NETADDR	HEXB		Physical network address
NOBFR	DECV	S	Number of packets discarded because no buffer available
NODST	DECV	S	Number of packets discarded because no destination for E-N protocol
OCOL	DECV	S	Number of output collisions
OCSEN	DECV	S	Number of output carrier lost errors
OHTBT	DECV	S	Number of output heartbeat errors
OHUNG	DECV	S	Number of hung output errors
OOWC	DECV	S	Number of output out of window collisions
OUNDRN	DECV	S	Number of output underrun errors
OXCOL	DECV	S	Number of excessive output collisions

Table 6.15 - Device characteristics for NET class devices			
Name	**Format**	**Set**	**Description**
PKTIN	DECV	S	Number of packets input
PKTOUT	DECV	S	Number of packets output
TYPE	TEXT		Device type

Following is a detailed description of each common device characteristic for the NET class devices.

BADPNT - Number of packets discarded because of bad ring pointer

This device characteristic records the number of input packets discarded because of an invalid network interface buffer ring pointer. This is an internal network interface error which should not occur if the hardware and driver software are working correctly.

BCPKTIN - Number of broadcast packets input

This device characteristic records the total number of input broadcast packets received.

BYTEIN - Number of bytes input

This device characteristic records the total number of bytes input.

BYTEOUT - Number of bytes output

This device characteristic records the total number of bytes output.

CLASS - Device class

This device characteristic returns the name of the device class (always NET here) to which the device belongs. When a value is specified for this characteristic, that value is compared to the name of the device class for the device. If they are the same, no further action is taken. If they are different, an ER_CHARV error is returned. This allows a program to verify that a given device is really a member of the class expected. This is useful since many devices implement device dependent functions which only behave as expected for a specific class device.

ICRC - Number of input CRC errors

This device characteristic records the total number of input CRC errors reported by the network interface. Packets with bad CRCs are discarded.

IFRAME - Number of input framing errors

This device characteristic records the total number of input framing errors reported by the network interface. Packets with framing errors are discarded.

ILOST - Number of lost input packets

This device characteristic records the total number of lost input packets reported by the network interface. A lost input packet is one that was discarded because there was no space for it in the network interface's internal buff

Chapter 7

Symbionts

This chapter describes the symbionts which provide various services on an XOS system. A symbiont is a program which executes in the background and provides services to users or to other programs.

Symbionts are started with the SYMBIONT command (see chapter 3, page 131). Symbionts generally remain active as long as the system is up after they are started, but can be terminated with the KILLPROC command (see chapter 3, page 93) if desired. Symbionts are usually started as part of the system startup procedure, but they can be started at any time by issuing the SYMBIONT command, either from a batch file or directly from the command line.

The remainder of this chapter describes the individual symbionts.

BOOTSRV Symbiont

Syntax:

SYMBIONT BOOTSRV keyword1=value1 keyword2=value2 ...

Purpose:

Implements the Internet RARP, BOOTP, and TFTP servers.

Keywords:

BOOTPDATA = filespec

This keyword specifies the name of the BOOTP data file. This file contains the database used to map between physical network addresses and IP addresses and provides other information needed when generating BOOTP responses. Only the name can be specified. The extension is always .DAT and it is always in the XOSSYS: directory. If this keyword is not specified, the file XOSSYS:BOOTP.DAT is used.

BOOTPDEV = device

This keyword specifies a device for the BOOTP server. The device must be a UDP class device. If this keyword is not specified, a BOOTP server stream is not started. It may be specified more than once to start multiple BOOTP server streams. Each server stream uses the data file specified by the most recently occurring BOOTPDATA keyword.

LOGFILE = filespec

This keyword specifies the output file for debug logging level. This keyword is ignored unless a non-zero LOGLEVEL value is also specified. If it is not specified and a non-zero LOGLEVEL value is specified, debug output is to the standard error device.

LOGLEVEL = number

This keyword specifies the local debug logging level. Debug logging provides a way to monitor the operation of the server. The value has the following meanings:

0 = No local logging (default)

1 = Log major events

2 = Log all network messages

RARPDATA = filename

This keyword specifies the name of the RARP data file. This file contains the database used to map between physical network addresses and IP addresses when generating RARP responses. Only the name can be specified. The extension is always .DAT and it is always in the XOSSYS: directory. If this keyword is not specified, the file XOSSYS:RARP.DAT is used.

RARPDEV = device

This keyword specifies a device for the RARP server. The device must be a NET class device. If this keyword is not specified, an RARP server stream is not started. It may be specified more than once to start multiple RARP server streams. Each server stream uses the data file specified by the most recently specified RARPDATA keyword.

TFTPDEV = device

This keyword specifies a device for the TFTP server. The device must be a UDP class device. If this keyword is not specified, a TFTP server stream is not started. It may be specified more than once to start multiple TFTP server streams. Each server stream has the open file limit specified by the most recently occurring TFTPNUM keyword.

TFTPNUM = number

This keyword specifies the maximum number of simultaneously open TFTP files. If this keyword is not specified, a value of 4 is used.

Options:

None.

Default:

Loglevel=0, TFTPNUM=4.

Description:

The BOOTSRV symbiont implements servers for the RARP, BOOTP, and TFTP protocols. These are all protocols which are used when booting remote systems. RARP and BOOTP allow a diskless system to obtain information about its configuration from a central database. TFTP is a simple file transfer protocol which is usually used to the initial executable image to a diskless system.

The BOOTSRV symbiont can be configured to support any one, two, or all three of these protocols. It can also support any number of server streams for each protocol, which each server stream serving a different network. Since most configurations will only support a single network, this feature will not normally be used.

FTPSRV Symbiont

Syntax:

SYMBIONT FTPSRV keyword1=value1 keyword2=value2 ...

Purpose:

Implements the Internet FTP server.

Keywords:

FTPCON = number

This keyword specifies the maximum number of simultaneously open FTP data/control connection pairs. If this keyword is not specified, a value of 4 is used. There are only 4 bytes of overhead associated with an unused but available open connection pair, so there is little harm in making this value reasonably large. Its main purpose is to provide a way to control overall system loading to prevent remote FTP access from swamping a small system.

FTPCPORT = number

This keyword specifies the TCP public port number that is used when listening for control connections. If this keyword is not specified, a value of 21, which is the standard Telnet public port, is used. This keyword does not start a server stream. It just stores the value given which is used when the next FTPDEV keyword is specified.

FTPDEV = device

This keyword specifies a device for the FTP server. The device must be a TCP class device. If this keyword is not specified, an FTP server stream is not started and the symbiont will terminate immediately. It may be specified more than once to start multiple FTP server streams. Each server stream has the queued open request, open connection limits, and port numbers specified by the most recently specified FTPOPEN, FTPCON, FTPCPORT, and FTPDPORT keywords.

FTPOPEN = number

This keyword specifies the number of queued TCP open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected.

FTPDPORT = number

This keyword specifies the TCP public port number that is used for data connections. If this keyword is not specified, a value of 20, which is the standard Telnet public port, is used. This keyword does not start a server stream. It just stores the value given which is used when the next FTPDEV keyword is specified.

LOGFILE = filespec

This keyword specifies the output file for debug logging level. This keyword is ignored unless a non-zero LOGLEVEL value is also specified. If it is not specified and a non-zero LOGLEVEL value is specified, debug output is to the standard error device.

LOGLEVEL = number

This keyword specifies the local debug logging level. Debug logging provides a way to monitor the operation of the server. The value has the following meanings:

0 = No local logging (default)

1 = Log major events

2 = Log all network messages

Options:

None.

Default:

FTPCON=4, FTPCPORT=21, FTPOPEN=2, FTPDPORT=20, LOGLEVEL=0.

Description:

The FTPSRV symbiont implements a server for the FTP protocol. This protocol provides a general file transfer capability.

IPSSRV Symbiont

Syntax:

SYMBIONT IPSSRV keyword1=value1 keyword2=value2 ...

Purpose:

Implements Internet TCP echo, TCP data sink, TCP data generator, UDP echo, UDP data sink, UDP date and time, and Domain Name System servers.

Keywords:

LOGLEVEL = number

This keyword specifies the local debug logging level. Debug logging provides a way to monitor the operation of the server. The value has the following meanings:

0 = No local logging (default)

1 = Log major events

2 = Log all network messages

LOGFILE = filespec

This keyword specifies the output file for debug logging level. This keyword is ignored unless a non-zero LOGLEVEL value is also specified. If it is not specified and a non-zero LOGLEVEL value is specified, debug output is to the standard error device.

TCPCON = number

This keyword specifies the maximum number of simultaneously open TCP connections for all servers for the TCP device. This global limit will have no effect unless it is less than the sum of the individual connection limits for the individual servers. A value of 0, which is the initial default, indicates no global limit.

TCPDEV = device

This keyword specifies theTCP device to use. This keyword does not actually start any server streams. It just specifies the device used for server streams started when a TCPxxxxC keyword is specified.

The device must be a TCP class device. This keyword must be specified before any TCPxxxxC keywords are specified. It is not needed if no TCPxxxxC keywords are given.

TCPDNSC = number

This keyword specifies the maximum number of simultaneous TCP Domain Name System connections and starts a TCP Domain Name System server stream. The TCP device, number of opens to queue, and the public port number values are taken from the values of the most recently specified TCPDEV, TCPDNSO, and TCPDNSP keywords, respectively.

TCPDNSO = number

This keyword specifies thenumber of queued TCP Domain Name System open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TCPDNSC keyword is specified.

TCPDNSP = number

This keyword specifies the TCP public port number that is used when listening for connections by the TCP Domain Name System server. This keyword does not start a server stream. It just stores the value given which is used when the next TCPDNSC keyword is specified.

TCPECHOC = number

This keyword specifies the maximum number of simultaneous TCP echo connections and starts a TCP echo server stream. The TCP device, number of opens to queue and the public port number values are taken from the values of the most recently specified TCPDEV, TCPECHOO, and TCPECHOP keywords, respectively.

TCPECHOO = number

This keyword specifies the number of queued TCP echo open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TCPECHOC keyword is specified.

TCPECHOP = number

This keyword specifies the TCP public port number that is used when listening for connections by the TCP echo server. This keyword does not start a server stream. It just stores the value given which is used when the next TCPECHOC keyword is specified.

TCPGENC = number

This keyword specifies the maximum number of simultaneous TCP data generator connections and starts a TCP data generator server stream. The TCP device, number of opens to queue, and the public port number values are taken from the values of the most recently specified TCPDEV, TCPGENO, and TCPGENP keywords, respectively.

TCPGENO = number

This keyword specifies the number of queued TCP data generator open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TCPGENC keyword is specified.

TCPGENP = number

This keyword specifies the TCP public port number that is used when listening for connections by the TCP data generator server. This keyword does not start a server stream. It just stores the value given which is used when the next TCPGENC keyword is specified.

TCPSINKC = number

This keyword specifies the maximum number of simultaneous TCP data sink connections and starts a TCP data sink server stream. The TCP device, number of opens to queue and the public port number values are taken from the values of the most recently specified TCPDEV, TCPSINKO, and TCPSINKP keywords, respectively.

TCPSINKO = number

This keyword specifies the number of queued TCP data sink open requests. Each queued open request uses slightly more than 4 KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TCPSINKC keyword is specified.

TCPSINKP = number

This keyword specifies the TCP public port number that is used when listening for connections by the TCP data sink server. This keyword does not start a server stream. It just stores the value given which is used when the next TCPSINKC keyword is specified.

TCPTIMEC = number

This keyword specifies the maximum number of simultaneous TCP time connections and starts a TCP time server stream. The TCP device, number of opens to queue and the public port number values are taken from the values of the most recently specified TCPDEV, TCPTIMEO, and TCPTIMEP keywords, respectively.

TCPTIMEO = number

This keyword specifies the number of queued TCP time open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TCPTIMEC keyword is specified.

TCPTIMEP = number

This keyword specifies the TCP public port number that is used when listening for connections by the TCP time server. This keyword does not start a server stream. It just stores the value given which is used when the next TCPTIMEC keyword is specified.

UDPDEV = device

This keyword specifies the UDP device to use. This keyword does not actually start any server streams. It just specifies the device used for server streams started when a UDPxxxxN keyword is specified.

The device must be a UDP class device. This keyword must be specified before any UDPxxxxN keywords are specified. It is not needed if no UDPxxxxN keywords are given.

UDPDNSN = number

This keyword specifies the maximum number of simultaneous UDP input requests to queue and starts a UDP Domain Name System server stream. The UDP device and the public port number values are taken from the values of the most recently specified UDPDEV and UDPDNSP keywords, respectively.

UDPDNSP = number

This keyword specifies the UDP public port number that is used when listening for input by the UDP Domain Name System server. This keyword does not start a server stream. It just stores the value given which is used when the next UDPDNSN keyword is specified.

UDPECHON = number

This keyword specifies the maximum number of simultaneous UDP inputs requests to queue and starts a UDP echo server stream. The UDP device and the public port number values are taken from the values of the most recently specified UDPDEV and UDPECHOP keywords, respectively.

UDPECHOP = number

This keyword specifies the UDP public port number that is used when listening for input by the UDP echo server. This keyword does not start a server stream. It just stores the value given which is used when the next UDPECHON keyword is specified.

UDPSINKP = number

This keyword specifies the UDP public port number that is used when listening for input by the UDP data sink server. This keyword does not start a server stream. It just stores the value given which is used when the next UDPSINKN keyword is specified.

UDPSINKN = number

This keyword specifies the maximum number of simultaneous UDP input requests to queue and starts a UDP data sink server stream. The UDP device and the public port number values are taken from the values of the most recently specified UDPDEV and UDPSINKP keywords, respectively.

UDPTIMEN = number

This keyword specifies the maximum number of simultaneous UDP input requests to queue and starts a UDP time server stream. The UDP device and the public port number values are taken from the values of the most recently specified UDPDEV and UDPTIMEP keywords, respectively.

UDPTIMEP = number

This keyword specifies the UDP public port number that is used when listening for input by the UDP time server. This keyword does not start a server stream. It just stores the value given which is used when the next UDPTIMEN keyword is specified.

UDPXECHON = number

This keyword specifies the maximum number of simultaneous UDP input requests to queue and starts a UDP extended echo server stream. The UDP device and the public port number values are taken from the values of the most recently specified UDPDEV and UDPXECHOP keywords, respectively.

UDPXECHOP = number

This keyword specifies the UDP public port number that is used when listening for input by the UDP extended echo server. This keyword does not start a server stream. It just stores the value given which is used when the next UDPXECHON keyword is specified.

Options:

None.

Default:

LOGLEVEL=0, TCPCON=0, TCPDNSO=2, TCPECHOO=2, TCPGENO=2, TCPSINKO=2, TCPTIMEO=2.

Description:

The IPSSRV symbiont implements a number of commonly used Internet servers. These are described below.

TCP echo server

This server implements the TCP echo server. It accepts TCP connections and simply echos back all data received on the connection. The connection must be terminated by the client; it is never terminated by the server.

TCP data sink server

This server implements the TCP data sink server. It accepts TCP connections and inputs and discards all input data. It never outputs any data. The connection must be terminated by the client, it is never terminated by the server.

TCP data generator server

This server implements the TCP data generator protocol. It accepts TCP connections and immediately generates continuous output consisting of an ASCII barber pole pattern. Any input data received is discarded. The connection must be terminated by the client; it is never terminated by the server.

TCP time server

This server accepts TCP connections and immediately outputs a text string which gives the current date and time. It then terminates the TCP connection.

TCP Domain Name System server

This server implements the TCP Domain Name System protocol. This is a complex protocol which is used to map between domain names and IP and mail addresses.

UDP echo server

This server implements the UDP echo server. It accepts UDP datagrams and simply echoes back each datagram received.

UDP extended echo server

This server implements the UDP extended echo server. It accepts UDP datagrams and echoes back each datagram received with some additional information inserted.

UDP data sink server

This server implements the UDP data sink server. It accepts and discards UDP datagrams. No output is ever generated.

UDP time server

This server implements the UDP time of day protocol. It accepts UDP datagrams (the contents of which are ignored) and sends back a reply containing an ASCII string which specifies the current date and time.

UDP Domain Name System server

This server implements the UDP Domain Name System protocol. This is a complex protocol which is used to map between domain names and IP and mail addresses.

SCREEN Symbiont

Syntax:

SYMBIONT SCREEN unit:number

Purpose:

Implements the XOS screen server.

Keywords:

None.

Options:

None.

Default:

None.

Description:

The XOS screen server provides functions used when switching between console virtual screens. It also displays and handles the system menu.

The single argument used when starting the screen server consists of two numeric values separated with a colon. The first number specifies the unit number for the console display the server will handle. For example, a value of 0 means that the server will handle screen functions for TRM0:. The second number specifies the number of virtual screens to create for the console. This value can be between 2 and 20. The screen server can only handle one console display. If the system has multiple console displays and a virtual screen capability is desired on more than one, a separate copy of the screen server can be run for each console.

The screen server cannot be restarted on a console. If it is terminated, it cannot be restarted until the system is re-booted. If its is terminated, the virtual screen which was displayed will continue to be displayed until the system is rebooted.

TLNSRV Symbiont

Syntax:

SYMBIONT TLNSRV keyword1=value1 keyword2=value2 ...

Purpose:

Implements the extended Internet Telnet server.

Keywords:

LOGFILE = filespec

This keyword specifies the output file for debug logging level. This keyword is ignored unless a non-zero LOGLEVEL value is also specified. If it is not specified and a non-zero LOGLEVEL value is specified, debug output is to the standard error device.

LOGLEVEL = number

This keyword specifies the local debug logging level. Debug logging provides a way to monitor the operation of the server. The value has the following meanings:

0 = No local logging (default)

1 = Log major events

2 = Log all network messages

NUMCON = number

This keyword specifies the maximum number of simultaneously open TCP connections for the server stream. If this keyword is not specified, a value of 4 is used. There are only 4 bytes of overhead associated with an unused but available open connection pair, so there is little harm in making this value reasonably large. Its main purpose is to provide a way to control overall system loading to prevent remote FTP access from swamping a small system.

NUMOPEN = number

This keyword specifies the number of queued TCP open requests. Each queued open request uses slightly more than 4KB of system memory. If this keyword is not specified, a value of 2 is used. A value of 4 or 6 is recommended if reasonably heavy usage is expected. A larger value is normally not needed unless very heavy usage is expected. This keyword does not start a server stream. It just stores the value given which is used when the next TLNDEV keyword is specified.

TLNDEV = device

This keyword specifies theTCP device to use and starts a Telnet server stream. The device must be a TCP class device. The number of TCP opens queued, the maximum number of simultaneous TCP connections, the TCP port number used, and the PCN device used are taken from the values specified by the most recent occurrences of the NUMOPEN, NUMCON, TLNPORT, and PCNDEV keywords, respectively.

TLNPORT = number

This keyword specifies the TCP public port number that is used when listening for connections. If this keyword is not specified, a value of 23, which is the standard Telnet public port, is used. This keyword does not start a server stream. It just stores the value given which is used when the next TLNDEV keyword is specified.

PCNDEV = device

This option specifies the name of the PCN (pseudo-console) class device to use when creating sessions. If this keyword is not specified, PCN0: is used. It should be noted that multiple server streams can use the same PCN device. The only reason for using different PCN devices is to make sessions created by different servers easily identifiable. This keyword does not start a server stream. It just stores the value given which is used when the next TLNDEV keyword is specified.

Options:

None.

Default:

NUMCON=4, LOGLEVEL=0, NUMOPEN=2, TLNPORT=23.

Description:

The TLNSRV symbiont implements an extended Telnet server. It supports the XOS protocol extensions which support all text mode display and keyboard BIOS functions as well as direct reads and writes to the screen buffer. This allows any text mode DOS program to be run remotely using

Telnet, even if it directly accesses the display hardware. The server also supports connections from standard Telnet clients which do not support the extended features. In this case, programs appear to be controlled from a serial terminal.

The server supports multiple server streams allowing a single server process to support multiple network connections. Each server stream can support any desired number of simultaneous connections (limited by other system resources, of course).

UNSPOOL Symbiont

Syntax:

SYMBIONT UNSPOOL keyword1=value1 keyword2=value2 ...

Purpose:

Implements the XOS unspool server.

Keywords:

OUTPUT = device

This option specifies the name of the output device.

UNIT = number

This option specifies the unit number for the unspool device to create.

DIRECT = directory

This option specifies the spooling directory. This must be a complete specification, beginning with a file structured, local non-removable device.

CLOSE = number

This option specifies the close time out value (in seconds) for automatic close. The default value is 0, which indicates no automatic close is to be done.

Options:

None.

Default:

CLOSE=0.

Description:

The unspool server copies data which has been spooled by output to a SPL class device to a physical output device. During initialization, the unspool server creates the SPL device which it will service. Each unspool server can service only a single SPL class device. Multiple unspool servers can be run to allow for multiple spooled devices.

If the unspool server is terminated, it cannot be restarted until the system is re-booted.

Chapter 8

System Error Messages

XOS error messages usually specify a five character or less mnemonic which uniquely identifies the error and a text string which describes the error. Some programs may output only the mnemonic or only the text string, but most will output both in the following format:

{ERCOD} Error message text

where ERCOD is the unique mnemonic for the error condition. This message may be preceded or followed by additional text which describes the conditions under which the error occurred or specify which system resources (such as files) were being accessed.

The following listing of error codes is in alphabetical order by the error code mnemonic. A numerically ordered list follows. The numeric listing includes just the value, mnemonic, and message text in tabular format for quick reference given the numeric value of the error code.

Alphabetical List of System Error Codes

ABORT = -150 - I/O operation aborted

An I/O operation was canceled after it was started. At least the amount of data indicated by the qab_amount value was transferred, but more may have been transferred.

ACT = -28 - Device is active

A function was specified for an active device which required that the device be inactive.

ADRER = -60 - Address out of bounds

An invalid address was specified as a parameter for a system service call.

ALDEF = -141 - Already defined

An attempt was made to define an entity (such as a shared memory segment or interprocess message name) with a name which was already in use.

BDALM = -205 - Bad alarm handle

The alarm handle specified as an argument for the svc_schalarm system call did not correspond to an active alarm.

BDDBK = -56 - Bad disk block number

An illegal disk block number was specified.

BDDVH = -57 - Bad device handle

A device handle, which did not specify a currently open device or file, was specified.

BDLNM = -99 - Bad logical name

The logical name specified contained illegal characters or was too long.

BDNAM = -18 - Bad process name

The process name specified contained illegal characters or was too long.

BDPID = -19 - Bad process ID

A process ID was specified which did not correspond to a possible process slot or to an existing process (if a reference to an existing process was required).

BDSPC = -29 - Bad device or file specification

An improperly formed device or file specification was specified.

BPIPE = -79 - Pipe error

An attempt was made to write to a pipe or to read from an empty pipe, the other end of which had been closed.

BUSY = -40 - File or device is busy

An attempt was made to perform some operation on a file or device which was busy for purposes of the attempted operation. For example, an attempt was made to delete or supersede a file which is currently being superseded.

CAASP = -184 - Close action already specified

An attempt was made to specify a close action for a file using the IOPAR_CLSACT I/O parameter when a close action had already been specified for the file. Only one close action can be specified for an open file.

CAERR = -185 - Close action error

An error occurred when attempting to perform the close action specified for a file.

CANCL = -151 - I/O operation canceled

An I/O operation was canceled after it queued but before it was started. No data was transferred.

CCMSS = -190 - Cannot change memory section size

An attempt was made to change the size of a memory section which has a constant size. Shared sections and sections which map device buffers usually have constant size.

CDAAD = -161 - LKE common data area already defined

A common data area being defined by an LKE is already defined.

CDAND = -162 - LKE common data area not defined

A common data area referenced by an LKE was not defined when the LKE was loaded. This error can only occur when loading an LKE.

CHARF = -16 - Illegal characteristic function

An illegal function was specified for a device or class characteristic.

CHARM = -17 - Required characteristic missing

A required device or class characteristic was not specified.

CHARN = -12 - Illegal characteristic name

The name specified for a device or class characteristic was illegal.

CHARS = -14 - Illegal characteristic value size

The value size specified for a device or class characteristic was illegal.

CHART = -15 - Illegal characteristic type

An illegal type was specified for a device or class characteristic.

CHARV = -13 - Illegal characteristic value

The value specified for a device or class characteristic was illegal.

CHNNA = -146 - DMA channel not available

The requested DMA channel is not available.

CLSAD = -109 - Device class already defined

The device class being added to the system is already defined.

CPDNR = -199 - Child process did not respond

The child process being created by a svc_iorun system call did not indicate that it had finished loading its program within the time-out period specified.

DATER = -46 - Data error

An unrecoverable data read error was encountered when attempting to input data from a device.

DATTR = -54 - Data truncated

Less data than expected was transferred to or from an I/O device. This error is only reported when this condition is due to an error condition. Reading less data than requested from a file because the end of file was reached does not produce this error, since this is a normal occurrence.

DEVER = -53 - Device error

An error was encountered when trying to access a device. This error generally indicates a failure of the device rather than the media, although this is not always true.

DEVFL = -35 - Device full

A mass storage device has no space available.

DEVIU = -31 - Device in use

A single user device was specified which was already in use by a different session process.

DFDEV = -37 - Different device for rename

An svc_iorename system service call specified a different device for the old and new file specifications.

DIRFL = -43 - Directory full

An attempt was made to create a new file when no additional space was available in the directory indicated (but not because the device was full). This error can only occur when attempting to create a new file in the root directory of an MS-DOS file system.

DIRNE = -44 - Directory not empty

An attempt was made to delete a non-empty directory.

DIRNF = -42 - Directory not found

A directory in the path specified for a file was not found.

DIRTD = -45 - Directory level too deep

A path to a file was specified with directories nested to more than the maximum directory nesting level. This limit is usually set to seven directory levels.

DIVER = -201 - Divide error

The child process being created by an svc_iorun system call terminated during initialization because of a divide error (divide by 0 or divide overflow).

DKCHG = -70 - Disk changed

The media for a removable media disk has been changed since a file was opened.

DKRMV = -149 - Disk removed

A removable media disk was removed since a file was opened.

DLOCK = -80 - Deadlock condition

An operation was attempted which would likely result in a deadlock condition.

DOSMC = -154 - DOS memory allocation data corrupted

The DOS memory block headers were corrupted, making it impossible to allocate or deallocate memory in a DOS program.

DOSPB = -180 - Permanent DOS process is busy

An attempt was made to load a DOS program into a session’s permanent DOS process when that process was not idle.

DPMIC = -207 - DPMI environment corrupted

The DPMI emulator was unable to perform the requested function because the per-process data which describes the DPMI environment for the process was not valid. This data is stored in user memory and can be corrupted by a misbehaved user program.

DQUOT = -96 - Disk quota exceeded

A process attempted to allocate more disk space than allowed by its disk quota.

DRFER = -90 - Directory block format error

When searching a directory for a file or for an empty slot, an illegal format was encountered.

DRRER = -91 - Directory block read error

When searching a directory for a file or for an empty slot, a read error occurred.

DRWER = -92 - Directory block write error

A write error occurred when attempting to update a directory block.

DTINT = -157 - Data transfer interrupted

A multi-block disk data transfer was terminated early because of a disk error. This error code is used internally by the disk optimization routines and should never be returned to a user program.

DUADF = -107 - Device unit already defined

The system device specified for an add unit function of the svc__ioclass system service call was already defined.

EOF = -1 - End of file

An attempt was made to read data past the end of a file. Note that this error is generally returned only when there is no data at all available before the end of file. An attempt to read past the end of a file when some data is available results in less data being read than was requested.

ERROR = -168 - Untranslatable/general error

This error code is reported for any error which is not covered by another error code. It is also used when mapping an error from a remote system and there is no direct mapping to a more specific error code.

EVNRS = -197 - Event is not reserved

The svc_schrelevent system call was issued to attempt to release an event which had not been reserved.

EVRES = -196 - Event is reserved

The svc_schresevent system call was issued to attempt to reserve an event which was already reserved.

EVSET = -198 - Event is set

An attempt was made to set an event using the svc_schsetevent system call which specified that the event should not be overwritten and the event was already set.

FBFER = -81 - FIB format error

An illegal format was encountered in a file information block (XOS file system).

FBPER = -82 - FIB pointer error

An invalid pointer was encountered in a file information block (XOS file system).

FBRER = -83 - FIB read error

An error occurred while reading a file information block (XOS file system).

FBWER = -84 - FIB write error

An error occurred while writing a file information block (XOS file system).

FILAD = -41 - File access denied

An attempt was made to access a file which the user is not privileged to access, which belongs to the user.

FILAF = -169 - File access failure

This error is returned for certain file access problems reported by foreign remote systems.

FILCF = -170 - File creation failure

This error is returned for certain file creation problems reported by foreign remote systems.

FILEX = -39 - File exists

An attempt was made to create a new file which specified that it should fail if the file existed and the file did exist, or an attempt was made to rename a file to the name of an existing file.

FILNF = -38 - File not found

An attempt was made to open a file which does not exist, or one which does exist that the user cannot access because it does not belong to the user or to a member of his user group.

FILRF = -172 - File rename failure

This error is reported for certain file rename problems reported by foreign remote systems.

FILXF = -171 - File extend failure

This error is reported for certain file allocation problems reported by foreign remote systems.

FSINC = -97 - File system is inconsistent

The file system was found to be internally inconsistent. This is generally a serious error which probably indicates that some data on the file system will not be accessible and that continued use of the file system will probably result in additional data loss. If this error occurs, the file system should be backed up immediately to recover as much data as possible and then it should be reformatted.

FTPER = -186 - FAT block pointer error

An invalid value was found in a FAT block when accessing a file on a DOS file structure.

FTRER = -187 - Error reading FAT block

An error occurred when reading a FAT block for a DOS file structure.

FTWER = -188 - Error writing FAT block

An error occurred when writing a FAT block for a DOS file structure

FUNC = -3 - Illegal function

A system service, which requires the specification of a function, was called with an illegal function specified.

FUNCM = -4 - Illegal function for current mode

The function specified in a system service call is illegal for the current mode of the device referenced.

HBFER = -85 - Home block format error

When attempting to mount a disk, an illegal format was found in the disk's home block (XOS file system) or boot block (DOS file system).

HBRER = -86 - Home block read error

When attempting to mount a disk, an error occurred when reading the disk's home block (XOS file system) or boot block (DOS file system).

IADEV = -69 - Illegal buffer address for device

The buffer specified for a data transfer had an illegal address. This generally occurs when a buffer which spans a page boundary is specified for a physical device transfer.

IATTR = -94 - Illegal file attribute change

An illegal file attribute change was specified, such as attempting to set or clear the directory attribute.

ICDEV = -68 - Illegal count for device

A transfer count was specified which was too large for the device.

IDEVC = -156 - Incorrect device class

The device class specified for the IOPAR_CLASS I/O parameter was incorrect.

IDFER = -47 - ID field error

An unrecoverable error was encountered when attempting to read the ID field of a disk.

IDREN = -159 - Invalid directory rename operation

An attempt was made to rename a directory into a directory which would result in an invalid directory tree.

IDSPC = -139 - Illegal destination file specification

The destination file specification for the svc_iodstname system call is not properly formed or is inconsistent with the source file specification and the search mask specified.

IFDEV = -67 - Illegal function for device

A function was specified for a device which is illegal for the device.

IIFF = -62 - Illegal image file format

An image file which was specified to be loaded for execution was not a properly formatted image file.

IIFRD = -63 - Illegal relocation data in image file

An image file which was specified to be loaded for execution contained improperly formatted relocation information.

IIFT = -61 - Illegal image file type

An image file which was specified to be loaded for execution was not of the proper type.

IINUM = -138 - Illegal interrupt number

An illegal interrupt number was specified by a device driver when attempting to initialize an interrupt vector.

ILLIN = -202 - Illegal instruction

The child process being created by an svc_iorun system call terminated during initialization because an illegal instruction was executed.

ILSEK = -78 - Illegal seek function

A seek operation was requested for a device which does not support seeks.

IMEMA = -144 - Illegal memory address

The memory address specified is illegal.

INCMO = -181 - Incomplete output operation

An output operation was not completed.

ININU -163 - Interrupt number in use

A device driver attempted to initialize an interrupt vector which was already in use.

INVST = -194 - Invalid segment type

An attempt was made to illegally move or otherwise modify a memory segment. This will occur if the segment was linked to an exec segment.

IOSAT = -158 - I/O saturation

An I/O operation was terminated because it was occurring at too high a rate. This generally indicates a defective device interface which is not clearing an interrupt request. The data transfer is terminated and the device is reset to keep it from hanging the rest of the system.

IPDIR = -192 - Illegal pointer in directory

An illegal value was found in the pointer in a directory to the first cluster of a file.

ISDIR = -76 - File is a directory

An operation was attempted on a file which is a directory, which is illegal for directories.

LASNA = -34 - Linear address space not available

The linear address space requested is not available.

LKEAL = -160 - LKE already loaded

The LKE being loaded is already loaded.

LOCK = -183 - File record lock violation

An attempt was made to lock a file record which was already locked by another user.

LSTER = -50 - Lost data error

Data was lost due to a device overrun or underrun.

MACFT = -24 - Memory address conflict

Memory is already allocated at the address which was specified for the allocation of memory. This error is most often associated with the svc_memchange and svc_memmap system service calls.

MAERR = -25 - Memory allocation error

User modifiable data which is required for memory allocation is inconsistent. This error can only be returned by the routines which emulate DOS memory allocation in virtual mode.

MATH = -257 - Math function error

An error was detected in one of the math library routines.

MEMLX = -208 - Memory limit exceeded

An attempt was made to allocate more memory than allowed for the process.

MPILK = -175 - Memory page is locked

An attempt was made to deallocate a memory page which was locked by an active I/O request.

MSNPR = -193 - Msect is not private

A attempt was made to convert a memory section that was not a simple private section to a shared section.

NACT = -72 - Device not active

The specified I/O request was not active on the device.

NCCLR = -126 = - Network connection cleared

The connection to a remote system was cleared by the remote system.

NCLST = -124 - Network connection lost

The connection to a remote system was terminated unexpectedly.

NCOMP = -142 - Not compatible

The requested operation was not compatible with the current state of the device, network, or system.

NCONG = -120 - Network congestion

Network communication failed because of excessive congestion in the network.

NCRFS = -127 - Network connection refused

An attempt to establish a connection with a remote system was refused by the remote system.

NDOSD = -155 - No DOS I/O data block available

A DOS system call could not be performed because no DOS I/O data block was available. These are dynamically allocated 256 byte blocks used to store parameters for certain DOS system calls.

NEMA = -23 - Not enough memory available

There is not enough free memory available in the system to satisfy a request for the allocation of additional memory or the request would cause the amount allocated to the process to exceed the amount the process is allowed. This error may be returned by any system service call which allocates memory to provide temporary workspace as well as by the memory system service calls.

NHSTA = -125 - Network host not available

The default network host for an XOS system configured as a workstation was not available.

NILAD = -116 - Illegal network address

An illegal network address was specified. This generally means that the external network complained about the format of the address.

NILPC = -118 - Illegal network protocol type

An illegal network protocol type was specified.

NILPR = -115 - Illegal network port number

An illegal network port number was specified.

NILRF = -117 - Illegal request format

This is a general error which means that the external network rejected some request because it was not formatted correctly.

NIYT = -256 - Not implemented yet

The operation attempted is not implemented in the current version of XOS.

NLKNA = -195 - Network link not available

An attempt was made to do I/O using a network interface which had been disabled.

NMBTS = -137 - Name buffer is too small

The name buffer (as specified with the IOPAR_FILSPEC I/O parameter) is not large enough to contain at least one file name when a repeated operation was specified.

NNAVL = -128 - Network not available

The connection to the network is not available.

NNOPC = -191 - No network protocol specified

A attempt was made to do I/O on a network device which requires specification of a underlying protocol and no such protocol was specified for the device. An example of this would be to attempt output on an Ethernet IP device for which no Ethertype value had been specified.

NNSER = -134 - Network name server error

A Domain Name Server indicated an unspecified error.

NNSNA = -131 - Network name server not available

A remote system cannot be accessed because no Domain Name Server could be found to resolve its Domain Name.

NNSNC = -129 - Network name server not capable

A Domain Name Server refused a request with an indication that it was not capable of performing the request.

NNSRF = -130 - Network name server refused request

A Domain Name Server refused a request. This generally means that the system is not privileged enough to make the request.

NNSRQ = -132 - Network name server bad format request

A Domain Name Server refused a request with an indication that the request had a bad format.

NNSRS = -133 - Network name server bad format response

The response received from a Domain Name Server had a bad format.

NOBUF = -27 - No system buffer available

This error is reported if a buffer cannot be obtained for the system internal buffer pool. This error should not occur. If it does, it probably means that the memory on a small system is heavily overcommitted.

NODCB = -26 - No disk cache buffer available

This error is reported if no disk cache buffer was available for use when needed. Disk cache buffers are large buffers used internally by the kernel for many purposes, but primarily for buffering disk data. This error should not occur. If it does, the number of disk buffers specified in the system startup file should be increased.

NOERR - 0 - No error indicated

An error code value of 0 is not used by the system. Such a value usually indicates an error in the user program's error handling routines, typically that a normal return from a system service or library routine was taken to be an error.

NOIN = -59 - Input not allowed

Input was attempted from a device that was opened without the O$IN bit set or from a device which does not support input.

NOMEM = -140 - Memory not allocated

The memory section specified does not exist.

NOOUT = -58 - Output not allowed

Output was attempted to a device that was opened without the O$OUT bit set or to a device which does not support output.

NOPAP = -143 - Printer is out of paper

A printer reported that it was out of paper.

NORSP = -55 - Device did not respond

A device did not complete the requested operation in a reasonable amount of time.

NOSAD = -65 - No starting address specified in image file

An image file was specified to be loaded for execution which did not contain a starting address specification.

NOSTK = -66 - No stack specified in image file

An image file was specified to be loaded for execution which did not contain a stack address specification.

NPCIU = -119 - Network protocol type in use

The network protocol type specified was already in use by the interface.

NPERR = -112 - Network protocol error

This error reports a general network protocol error which does not fall into any more specific error category.

NPRIU = -114 - Network port in use

The network port specified is in use.

NPRNO = -113 - Network port not open

The network port specified is not open.

NRTER = -121 - Network routing error

The external network was unable to route a message to its destination.

NRTNA = -136 - Network router not available

No router is available when one is required to send a message to a remote system which is on a different sub-net.

NSCLS = -108 - No such device class

The device class specified for the svc_ioclass system service call did not exist.

NSDEV = -30 - No such device

A device which does not exist in the system was specified or a device which does exist but which the process is not privileged to use was specified.

NSEGA = -22 - No segment available

An attempt was made to create a segment when the process already has the maximum allowable number of segments.

NSLP = -182 - Not a session level process

A process which was not a session level process was specified for a function which required a session level process.

NSNOD = -122 - No such network node

The remote network node specified did not respond or is not known to the external network.

NSP = -20 - No such process

The requested process does not exist.

NSTYP = -145 - No such device type

The device type specified for the add unit function was not valid for the device class to which a unit was being added.

NTDEF = -98 - Not defined

The logical name or environment string specified was not defined.

NTDIR = -75 - File is not a directory

A directory specified in a path for a file is not a directory.

NTDSK = -95 - Device is not a disk

A non-disk device was specified for a function which requires a disk.

NTFIL = -93 - Device is not file structured

A device which is not file structured was specified for an operation which requires a file structured device.

NTIMP = -167 - Not implemented

The requested function is not implemented by the hardware.

NTLCL = -179 - Not local

An operation which is valid only for a local device was requested for a remote device.

NTLNG = -101 - Name is too long

The file, directory, or device name specified was too long.

NTRDY = -74 - Device not ready

The requested I/O operation could not be performed because the device was not ready, e.g., printer was off line, floppy disk was not inserted, etc. .

NTTIM = -123 - Network time-out

A response to a network message was not received in a reasonable length of time.

NTTRM = -77 - Device is not a terminal

A terminal specific operation was attempted on a device which was not a terminal.

NWPA = -204 - No watchpoint available

An attempt was made to set a hardware watchpoint when all available watchpoints were already in use. The 80386/80486 processors support a maximum of four hardware watchpoints.

NXERR = -135 - Network transmit error

An error occurred when attempting to transmit a network message.

PARMF = -10 - Illegal parameter function

An illegal function was specified for an I/O parameter.

PARMI = -6 - Illegal parameter index

The index value specified for an I/O parameter was illegal.

PARMM = -11 - Required parameter missing

A required I/O parameter was not specified.

PARMS = -8 - Illegal parameter size

The value size specified for an I/O parameter was illegal.

PARMT = -9 - Illegal parameter type

An illegal type was specified for an I/O parameter.

PARMV = -7 - Illegal parameter value

The value specified for an I/O parameter was illegal.

PDADF = -106 - Physical device already defined

The physical device specified for an add unit function of the svc_ioclass system service call was already defined as a system device.

PDNAV = -105 - Physical device not available

The physical device specified for an add unit function of the svc_ioclass system service call could not be found.

PDTYP = -104 - Physical device type incorrect

The physical device type was invalid for the function requested.

PRIV = -21 - Privilege failure

An attempt was made to perform an operation which required a privilege which the process did not possess.

RANGE = -258 - Math function argument out of range

An argument to one of the math library routines was out of range.

RELTR = -64 - Relocation truncation in image file

An image file which was specified to be loaded for execution contained relocation information which resulted in a relocated value being truncated.

RNFER = -49 - Record not found error

The requested record was not found on the indicated track on a disk.

SBFER = -87 - Storage allocation block format error

When attempting to mount a disk, a format error was found in a storage allocation block (XOS file system) or a file allocation table (DOS file system). If this error occurs, serious consideration should be given to backing up the disk involved immediately, since it probably indicates a serious problem which could compromise file integrity.

SBRER = -88 - Storage allocation block read error

When attempting to mount a disk, a read error occurred while reading a storage allocation block (XOS file system) or a file allocation table (DOS file system). If this error occurs, serious consideration should be given to backing up the disk involved immediately, since it probably indicates a serious problem which could compromise file integrity.

SBWER = -89 - Storage allocation block write error

When attempting to allocate space or close a file, a write error occurred while writing a storage allocation block (XOS file system) or file allocation table (DOS file system) to the disk. If this error occurs, serious consideration should be given to backing up the disk involved immediately, since it probably indicates a serious problem which could compromise file integrity.

SEKER = -48 - Seek error

An unrecoverable seek error was encountered when attempting to position a disk to the desired track.

STKER = -200 - Stack error

The child process being created by an svc_iorun system call terminated during initialization because of a memory fault when accessing the user stack.

SVC = -2 - Illegal SVC function

An illegal system service function was specified.

TMALM = -206 - Too many alarms for process

An attempt was made to create more alarms than allowed for a process.

TMDDV = -178 - Too many device units for device

There are too many device units declared for this device.

TMDVC = -111 - Too many devices open for device class

There are no more allocatable devices available in the device class.

TMDVP = -36 - Too many devices open for process

An attempt was made by a single process to open more devices than the process device limit.

TMIOM = -173 - Too many I/O requests for memory page

A I/O operation would result in a single memory page being locked by more than 255 different requests. It is highly unlikely that this error will occur in normal system use.

TMIOP = -174 - Too many I/O request pointers

More than 8 contiguous groups of memory pages needed to be locked in memory for an I/O operation. A memory referenced by an I/O operation must be locked, including file specification strings, data buffers, I/O parameter lists, and string values of I/O parameters.

TMIOQ = -176 - Too many I/O requests queued

An svc_ioqueue system call would exceed the request queue limit for the I/O device.

TMPSS = -103 - Too many processes or shared segments in system

A attempt to create a child process was made when the system already contained the maximum allowed number of processes.

TRMNA = -210 - Terminal is not attached

An I/O operation was attempted on a terminal device which was not attached to a physical serial port or console display.

TMRNC = -148 - Too many requests for network connection

A function was requested which exceeded the multiplexing capacity of a network connection for the protocol being used.

TMRQB = -189 - Too many requests for buffer

Too many requests were made for access to a disk cache buffer. This error is extremely unlikely, since the maximum number of requests is 65534 for each buffer.

TMUDV = -177 - Too many users for device

A device has been opened too many times.

TMUSR = -102 - Too many users

An attempt was made to create a new user session which would exceed the maximum number of user sessions allowed.

UNXSI = -203 - Unexpected signal.

The child process being created by an svc_iorun system call terminated during initialization because of an unexpected signal.

VALUE = -5 - Illegal value

An illegal value was given as an argument to a system call.

VECNS = -209 - Signal vector not set up

An attempt was made to request some action (such as setting up an alarm) which required that a vector be set up and the vector was not set up.

WLDNA = -100 - Wildcard name not allowed

A wildcard file name was given where a fully specified file name was required.

WPRER = -52 - Write protect error

An attempt was made to write to a write protected device.

WRTER = -51 - Write fault error

A detectable error occurred while writing to a device.

XFRBK = -110 - Transfer blocked

A data transfer could not be completed.

Numerical List of System Error Codes

Value	Name	Description
0	NOERR	Normal return
-1	EOF	End of file
-2	SVC	Illegal SVC function
-3	FUNC	Illegal function
-4	FUNCM	Illegal function for current mode
-5	VALUE	Illegal value
-6	PARMI	Illegal parameter index
-7	PARMV	Illegal parameter value
-8	PARMS	Illegal parameter value size
-9	PARMT	Illegal parameter type
-10	PARMF	Illegal parameter function
-11	PARMM	Required parameter missing
-12	CHARN	Illegal characteristic name
-13	CHARV	Illegal characteristic value
-14	CHARS	Illegal characteristic value size
-15	CHART	Illegal characteristic type
-16	CHARF	Illegal characteristic function
-17	CHARM	Required characteristic missing
-18	BDNAM	Bad process name
-19	BDPID	Bad process ID
-20	NSP	No such process
-21	PRIV	Not enough privilege
-22	NSEGA	No segment available
-23	NEMA	Not enough memory available

Value	Name	Description
-24	MACFT	Memory allocation conflict
-25	MAERR	Memory allocation error
-26	NODCB	No disk cache buffer available
-27	NOBUF	No system buffer available
-28	ACT	Device is active
-29	BDSPC	Bad device or file specification
-30	NSDEV	No such device
-31	DEVIU	Device in use
-32	DEVIO	Device is open
-33	DEVNO	Device not open
-34	LASNA	Linear address space not available
-35	DEVFL	Device is full
-36	TMDVP	Too many devices open for process
-37	DFDEV	Different device for rename
-38	FILNF	File not found
-39	FILEX	File exists
-40	BUSY	File or device is busy
-41	FILAD	File access denied
-42	DIRNF	Directory not found
-43	DIRFL	Directory full
-44	DIRNE	Directory not empty
-45	DIRTD	Directory level too deep
-46	DATER	Data error
-47	IDFER	ID field error
-48	SEKER	Seek error
-49	RNFER	Record not found error
-50	LSTER	Lost data error

Value	Name	Description
-51	WRTER	Write fault error
-52	WPRER	Write protect error
-53	DEVER	Device error
-54	DATTR	Data truncated
-55	NORSP	Device did not respond
-56	BDDBK	Bad disk block number
-57	BDDVH	Bad device handle
-58	NOOUT	Output not allowed
-59	NOIN	Input not allowed
-60	ADRER	Address error
-61	IIFT	Illegal image file type
-62	IIFF	Illegal image file format
-63	IIFRD	Illegal relocation data in image file
-64	RELTR	Relocation truncation in image file
-65	NOSAD	No starting address specified in image file
-66	NOSTK	No stack specified in image file
-67	IFDEV	Illegal function for device
-68	ICDEV	Illegal count for device
-69	IADEV	Illegal buffer address for device
-70	DKCHG	Disk changed
-71	RTOBG	Record too big
-72	NACT	Device not active
-73	FMTER	Format error
-74	NTRDY	Device not ready
-75	NTDIR	File is not a directory
-76	ISDIR	File is a directory
-77	NTTRM	Device is not a terminal

Value	Name	Description
-78	ILSEK	Illegal seek function
-79	BPIPE	Pipe error
-80	DLOCK	Deadlock error
-81	FBFER	FIB format error
-82	FBPER	FIB pointer error
-83	FBRER	FIB read error
-84	FBWER	FIB write error
-85	HMFER	HOM format error
-86	HMRER	HOM read error
-87	SBFER	SAT format error
-88	SBRER	SAT read error
-89	SBWER	SAT write error
-90	DRFER	Directory format error
-91	DRRER	Directory read error
-92	DRWER	Directory write error
-93	NTFIL	Not a file structured device
-94	IATTR	Illegal file attribute change
-95	NTDSK	Device is not a disk
-96	DQUOT	Disk quota exceeded
-97	FSINC	File system is inconsistent
-98	NTDEF	Not defined
-99	BDLNM	Bad logical name
-100	WLDNA	Wildcard name not allowed
-101	NTLNG	Name is too long
-102	TMUSR	Too many users
-103	TMPSS	Too many processes or shared segments in system

Value	Name	Description
-104	PDTYP	Physical device type incorrect
-105	PDNAV	Physical device not available
-106	PDADF	Physical device already defined
-107	DUADF	Device unit already defined
-108	NSCLS	No such device class
-109	CLSAD	Device class already defined
-110	XFRBK	Data transfer blocked
-111	TMDVC	Too many devices open for device class
-112	NPERR	Network protocol error
-113	NPRNO	Network port not open
-114	NPRIU	Network port in use
-115	NILPR	Illegal port number
-116	NILAD	Illegal network address
-117	NILRF	Illegal request format
-118	NILPC	Illegal network protocol type
-119	NPCIU	Network protocol type in use
-120	NCONG	Network congestion
-121	NRTER	Network routing error
-122	NSNOD	No such network node
-123	NTTIM	Network time out
-124	NCLST	Network connection lost
-125	NHSNA	Network host not available
-126	NCCLR	Network connection cleared
-127	NCRFS	Network connection refused
-128	NNAVL	Network not available
-129	NNSNC	Network name server not capable
-130	NNSRF	Network name server refused request

Value	Name	Description
-131	NNSNA	Network name server not available
-132	NNSRQ	Network name server bad format request
-133	NNSRS	Network name server bad format response
-134	NNSER	Network name server error
-135	NXERR	Network transmit error
-136	NRTNA	Network router not available
-137	NMBTS	Name buffer is too small
-138	IINUM	Illegal interrupt number
-139	IDSPC	Illegal destination file specification
-140	NOMEM	No memory allocated
-141	ALDEF	Already allocated
-142	NCOMP	Not compatible
-143	NOPAP	Printer is out of paper
-144	IMEMA	Illegal memory address
-145	NSTYP	No such device type
-146	CHNNA	DMA channel not available
-148	TMRNC	Too many requests for network connection
-149	DKRMV	Disk removed
-150	ABORT	IO operation aborted
-151	CANCL	IO operation canceled
-154	DOSMC	DOS memory allocation data corrupted
-155	NDOSD	No DOS I/O data block available
-156	IDEVC	Incorrect device class
-157	DTINT	Data transfer interrupted
-158	IOSAT	IO saturation
-159	IDREN	Invalid directory rename
-160	LKEAL	LKE already loaded

Value	Name	Description
-161	CDAAD	LKE common data area already defined
-162	CDAND	LKE common data area not defined
-163	ININU	Interrupt number in use
-167	NTIMP	Not implemented
-168	ERROR	Unspecified general error
-169	FILAF	Cannot access file
-170	FILCF	Cannot create file
-171	FILXF	Cannot extend file
-172	FILRF	Cannot rename file
-173	TMIOM	Too many I/O requests for memory page
-174	TMIOP	Too many I/O request pointers
-175	MPILK	Memory page is locked
-176	TMIOQ	Too many I/O requests queued
-177	TMUDV	Too many users for device
-178	TMDDV	Too many device units for device
-179	NTLCL	Not local
-180	DOSPB	Permanent DOS process is busy
-181	INCMO	Incomplete output operation
-182	NSLP	Not a session level process
-183	LOCK	File record lock violation
-184	CAASP	Close action already specified
-185	CAERR	Close action error
-186	FTPER	FAT block pointer error
-187	FTRER	Error reading FAT block
-188	FTWER	Error writing FAT block
-189	TMRQB	Too many requests for buffer
-190	CCMSS	Cannot change memory section size

Value	Name	Description
-191	NNOPC	No network protocol specified
-192	IPDIR	Illegal pointer in directory
-193	MSNPR	Msect is not private
-194	INVST	Invalid segment type
-195	NLKNA	Network link not available
-196	EVRES	Event is reserved
-197	EVNRS	Event is not reserved
-198	EVSET	Event is set
-199	CPDNR	Child process did not respond
-200	STKER	Stack error
-201	DIVER	Divide error
-202	ILLIN	Illegal instruction
-203	UNXSI	Unexpected software interrupt
-204	NWPA	No watchpoint available
-205	BDALM	Bad alarm handle
-206	TMALM	Too many alarms for process
-207	DPMIC	DPMI environment corrupted
-208	MEMLX	Memory limit exceeded
-209	VECNS	Signal vector not set up
-210	TRMNA	Terminal is not attached
-256	NIYT	Not implemented yet
-257	MATH	Math function error
-258	RANGE	Math function argument out of range

Appendix A

Command Comparison

This appendix provides command comparison tables between XOS and other operating systems. The information is provided for reference as a translation table by those users acquainted with DOS, UNIX, or VMS.

DOS to XOS Command Comparision

DOS	XOS
@	@
;	N/A
?	N/A
APPEND	LOGICAL
ASSIGN	LOGICAL
ATTRIB	ATTRIB
BACKUP	N/A
BREAK	N/A
BUFFERS	CLSCHAR DISK
CALL	CALL
CHCP	N/A
CHDIR, CD	CHDIR, CD
CHKDSK	CHKDSK
CHOICE	N/A
CLS	CLS
COMMAND	COMMAND
COMP	N/A
COPY	COPY
COUNTRY	N/A
CTTY	N/A
DATE	DATE, DAYTIME
DEBUG	XDT
DEL, DELETE, ERASE	DEL, DELETE, ERASE
DELTREE	DEL ...\
DEVICE	N/A

DOS	XOS
DEVICEHIGH	N/A
DIR	DIR
DISKCOMP	N/A
HELP	N/A
DISKCOPY	N/A
DOS	N/A
DOSKEY	ALIAS
DOSSHELL	N/A
DRIVPARM	N/A
ECHO	ECHO
EDIT	VID
EDLIN	VID
EMM386	CLSCHAR PROCESS
ERASE, DEL, DELETE	ERASE, DEL, DELETE
EXE2BIN	N/A
EXIT	EXIT
EXPAND	N/A
FASTOPEN	N/A
FC	N/A
FCBS	N/A
FDISK	N/A
FILES	CLSCHAR PROCESS
FIND	N/A
FOR	FOR
FORMAT	N/A
GOTO	GOTO
GRAFTABL	N/A

DOS	XOS
GRAPHICS	N/A
IF	IF
INSTALL	N/A
JOIN	N/A
KEYB	N/A
LABEL	LABEL
LASTDRIVE	N/A
LOADHIGH	N/A
MEM	N/A
MIRROR	N/A
MKDIR, MD	MKDIR, MD
MODE	DEVCHAR, MODE
MORE	MORE
MOVE	MOVE
NLSFUNC	N/A
PATH	PATH, LOGICAL
PAUSE	PAUSE
PRINT	N/A
PROMPT	PROMPT
QBASIC	N/A
RECOVER	N/A
REM	REM
RENAME	RENAME
REPLACE	N/A
RESTORE	N/A
RMDIR, RD	RMDIR, RD
SET	SET, SETENV

DOS	XOS
SETVER	N/A
SHARE	N/A
SHELL	N/A
SHIFT	SHIFT
SMARTDRV	CLSCHAR DISK
SORT	N/A
STACKS	N/A
SUBST	LOGICAL
SWITCHES	N/A
SYS	N/A
TIME	DAYTIME, TIME
TREE	N/A
TYPE	TYPE
UNDELETE	N/A
UNFORMAT	N/A
VER	VER
VERIFY	N/A
VOL	VOL
XCOPY	COPY

UNIX to XOS Command Comparison

UNIX	XOS
ALIAS (C shell)	ALIAS
AR, TAR	N/A
CAT	TYPE
CHMOD	ATTRIB
CD	CHDIR, CD
CP	COPY
CSH, SH	SHELL
GREP, EGREP, FGREP	N/A
HISTORY	HISTORY
LS	DIR
MKDIR	MKDIR, MD
MV	MOVE, RENAME
PS	SHOW, SYSDIS
PWD	CHDIR, CD
RM	DELETE, DEL, ERASE
RMDIR	RD, RMDIR
SETENV	SET, SETENV
SDB, ADB	XDT
VI, ED, EDIT, EX, SED	VID

VMS to XOS Command Comparison

VMS	XOS
@	SHELL
BACKUP	N/A
CLEAR	CLS
COPY	COPY
CREATE/DIR	MKDIR, MD
DEBUG	XDT
DELETE	DELETE, DEL, RMDIR, RD
DIRECTORY	DIR
EDT, EDIT	VID
RENAME	RENAME
SEARCH	N/A
SET FILE	ATTRIB
SET LOGICAL, SET SYMBOL SHOW LOGICAL, SHOW SYMBOL	LOGICAL
SHOW	CLSCHAR, DEVCHAR, SHOW
SHOW DEFAULT, SET DEFAULT	CHDIR, CD
SHOW SYSTEM	SYSDIS
STOP PROCESS	KILLPROC
TYPE	TYPE

Appendix B

Technical Support

Contacting XOS Systems

When a program attempts an illegal action, an error message is generated and the action fails. However, in certain cases, this action has unanticipated consequences and may cause the system to stop operating for any of a number of reasons. XOS operating system failures are explained in a fatal system error box. An example of an XOS fatal system error box is found on the next page. Copy the page, and transcribe the error box onto the page before calling XOS Technical Support.

The important information to write down when you see this box is:

Everything in the top half of the box, up to Data and Stack.

Before you call

Before calling technical support, you need to gather the following information:

System Configuration:

System RAM

Hard disk brand and type, model number, or parameters

Display adapter brand and type

Other peripherals attached to the system

What programs are running (active) on the system

Problem description:

Error message or number

Version of the software causing the problem

Steps needed to reproduce the problem

Fatal system error information

Call Us

XOS Systems can be reached at (520) 795-6000 from 9am to 5pm MST, Monday to Friday (except holidays). The telephone number provided also functions as a message system, should the lines be busy, or if no technicians are available. Alternatively, problem reports can be faxed to (520) 795-0158.

Fatal System Error Box

<figure B-1>

XOS bug/enhancement report

<figure B-2>

Trouble Symptom List

This section only assumes trouble with XOS. If you suspect a hardware problem, diagnose and correct that problem first.

Hardware compatibility problems can also occur; these are usually best resolved by:

Determining what part of the hardware is causing the trouble

Checking the connections, jumpers, switches, etc. for that equipment

Either modifying the hardware settings or the corresponding XOS configuration information

If the problem cannot be resolved, call XOS technical support for further information and assistance.

The following general steps may be helpful in diagnosing trouble conditions:

Was the hardware configuration changed? If yes, was XOS reconfigured for the new hardware?

Does the problem disappear when a certain peripheral or piece of hardware is removed from the system? If yes, is there a hardware/hardware or hardware/software incompatibility?

Is the problem reproducible, or is it intermittent? Can you trace it to a specific command, program, or set of conditions?

Commonly Asked Questions

Why won't XOS run on my 8088 or 80286 machine?

XOS uses functions only provided by the Intel (and 100% compatible) 80386sx/dx and subsequent CPU chips for memory use and multitasking. Because these functions were not available in earlier chips, XOS cannot be run on those machines.

<...>

Index

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X