KIX™

for the Apple II

OWNER'S MANUAL

Kyan Software, Inc.
San Francisco, California

Copyright

This users manual and the computer software (programs) described in it are copyrighted by Kyan Software Inc. with all rights reserved. Under the copyright laws, neither this manual nor the programs may be copied, in whole or part, without the written consent of Kyan Software Inc. The only legal copies are those required in the normal use of the software or as backup copies. This exception does not allow copies to be made for others, whether or not sold. Under the law, copying includes translations into another language or format.

Notice

Kyan Software reserves the right to make improvements to the products described in this manual at any time and without notice. Kyan Software cannot guarantee that you will receive notice of such revisions, even if you are a registered owner. You should periodically check with Kyan Software or your authorized Kyan Software dealer.

Kyan Software programs are sold only on the condition that the purchaser agrees to the terms contained in the Software License Agreement that begins on the following page. Please read this agreement before using the software.

1850 Union Street #183
San Francisco, CA 94123
(415) 626-2080

Software License Agreement

IMPORTANT: Kyan Software products are sold only on the condition that the purchaser agrees to the terms of the following license. PLEASE READ THIS AGREEMENT CAREFULLY. If you do not agree to the terms, return the unused package to Kyan Software or to your dealer immediately for a refund. If you agree to the terms contained in this License Agreement, complete the enclosed registration card and return it to Kyan Software.

When you purchase a Kyan Software product, you acknowledge that:

1. Kyan Software has a valuable proprietary interest in the computer programs and printed documentation (hereafter called "SOFTWARE"). What you have purchased is a non-transferable and non-exclusive license to use the SOFTWARE. Kyan Software retains ownership of the SOFTWARE.

2. You may not copy or reproduce the SOFTWARE for any purpose, other than to make backup or archive copies as provided for in Section 117 of the U.S. Federal Copyright Law, without the express permission of Kyan Software.

3. You may not copy, distribute, or otherwise make the SOFTWARE available to any third party without the express permission of Kyan Software.

4. This license is effective until terminated. You may terminate it at any time by destroying the SOFTWARE along with all copies, modifications and merged portions in any form. It will also terminate if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the SOFTWARE together with all copies, modifications and merged portions in any form.

Limited Warranty

Kyan Software warrants the diskette on which the Kyan software is furnished to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to you as evidenced by your proof of purchase.

Kyan also warrants that this software performs substantially in accordance with the published specification statement, the documentation, and authorized advertising. Kyan, if notified of significant errors within 90 days from the date of purchase, will at its option:

a) correct demonstrable and significant program or documentation errors within a reasonable period of time; or

b) provide the customer with functionally equivalent software; or

c) provide or authorize a refund.

Except for the limited warranty described in the preceding paragraphs, Kyan Software makes no other warranties, either express or implied, with respect to the software, its merchantability or its fitness for any particular purposes.

Some states do not allow the exclusion or limitation of implied warranties or liabilities for incidental or consequential damages, so the above limitations or exclusions may not apply to you.

This Agreement constitutes the entire agreement between the parties concerning the subject matter hereof.

Disclaimer of Warranty -- Apple Computer

Apple Computer Inc. makes no warranties, either express or implied, regarding the enclosed computer software package, its merchantability or its fitness for any particular purpose. The exclusion of implied warranties is not permitted by some states. The above exclusion may not apply to you. This warranty provides you with specific legal rights. There may be other rights that you may have which vary from state to state.

ProDOS is a copyrighted program of Apple Computer Inc. licensed to Kyan Software to distribute for use only in combination with Kyan Software products. Apple software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of Kyan Software products. When the Kyan Software product has completed execution, Apple software shall not be used by any other program.

WELCOME TO THE KYAN SOFTWARE FAMILY

KIX is a powerful extension of the ProDOS Operating System. The KIX software package consists of this user manual, a Quick Guide to KIX, and a single diskette.

We strongly recommend that you make and use backup copies of the KIX diskette. Keep your original KIX diskette in a safe location in case something happens to your copies.
(Remember ..... Murphy is alive and well, and he loves to mess with computers!)

Kyan Software has enclosed an owner registration card. Please fill in and return this card as soon as possible. Registered owners of Kyan Software products are eligible for technical support and periodic low-cost software upgrades. Registered owners can also subscribe to **"UPDATE ... Kyan"**, a bimonthly newsletter which contains programming tips, utility programs, and the latest information regarding upgrades and new product releases.

Copy Protection

Kyan Software products are not copy-protected. As a result, you are able to make backup copies and load your software onto a hard disk or into a RAM expansion card. We trust you. Please do not violate our trust by making or distributing illegal copies.

Technical Support

Kyan Software has a technical support staff ready to assist you with any problems you might encounter. If you have a problem, we request that you first consult this users manual. We have worked very hard to identify and include in this manual, the answers to questions and problems most frequently encountered.

If you have a problem which is not covered in the manual, our support staff is ready to help. If the problem is technical in nature, we can best help if you send us a written description of the problem. We will do our best to get back to you with an answer as quickly as possible.

If you question can be answered on the phone, then give us a call. Our technical staff is available to assist on Monday through Friday between the hours of 9 AM and 5 PM, West Coast Time. You may reach them by calling:

Technical Support: (415) 626-2080

Suggestion Box

Kyan Software likes to hear from you. Please write if you have suggestions, comments and, yes, even criticisms of our products. We do listen. Your suggestions and comments frequently lead to new products and/or product modifications.

We encourage you to write. To make it easier, we have included a form in the back of this manual. This form makes it easier for you to write and easier for us to understand and respond to your comments. Please let us hear from you.

Mailing Address: Kyan Software Inc.
1850 Union Street #183
San Francisco, CA 94123

Table of Contents

I. INTRODUCTION

Overview

KIX is a powerful disk and file management system that gives the programmer immediate and direct access to any file in the system. It extends the Apple ProDOS operating system so that it includes features like those supported by the UNIX operating system. KIX includes 25 of the most frequently used UNIX utilities which allow you to quickly and easily manipulate files, volumes, and directories. KIX makes ProDOS a more useful, efficient and productive operating system for the Apple II.

This chapter reviews the general purpose and function of operating systems and the concept of a hierarchical file structure. It also describes the KIX system and the family of KIX utilities. The chapter is organized in the following manner.

Operating Systems

An operating system is the manager which organizes and directs the flow of information through the computer. When information must be stored, it is the operating system that decides where and instructs the computer how, and when to store it. When information must be retrieved, it is again the operating system which accesses the data and delivers it to the computer. And, when information must be printed, it is the operating system which tells the printer what and how to print.

The operating system is responsible for managing all elements of the computer system hardware. It is the job of the operating system to manage these elements efficiently and to maximize the productivity of the hardware system. The hardware system consists of:

Central Processing Unit (CPU). The CPU is the "brains" of the computer. It receives and executes commands from the operating system and acts as the hub or focal point for all activities of the computer.

Temporary Storage. When the CPU processes data, it uses electronic memory for temporary storage of information. Random Access Memory or RAM is electronic memory. Information can be stored and retrieved from RAM many times faster than from magnetic memory. However, all information stored in RAM is lost when the computer is turned off.

Permanent Storage. The computer uses a floppy disk, hard disk, and/or magnetic tape for permanent storage of programs and information. The operating system controls the organization (format) of information stored on each type of magnetic memory device and directs the storage and retrieval of information.

A second form of permanent memory is Read-Only Memory or ROM. ROM represents software which has been permanently encoded in an integrated circuit. It is an integral part of the computer circuitry and allows information to be retrieved many times faster than from magnetic memory. Typically, ROM is the first memory accessed when the computer is turned on; it contains instructions to the CPU to load and run the operating system. ROM can also be used to permanently store specialized software routines, libraries, and even applications software.

Input/Output Ports. Input/Output ports are the interface between the computer and peripheral devices such as printers, modems, plotters (and sometimes keyboards and monitors). On the Apple //, these ports are called "slots". Interface cards are inserted into these slots, and, through these cards, the operating system communicates with peripherals.

Peripherals. Peripherals are the devices which allow interaction with the computer. They consist of input devices (e.g., keyboards, joystick, mice, modems, digitizers) and output devices (e.g., monitors, printers, plotters, modems). The operating system manages the flow of information between the computer and these peripheral devices.

The operating system manages the interaction and flow of information between all hardware components of the computer system. Its activities are, for the most part, automatic and transparent to the user.

The operating system also forms the foundation on which applications software (such as word processing, spreadsheets, and accounting programs) are developed and executed.

Finally, the operating system provides a fast, convenient and easy to-use interface for the user. The system provides all of the tools needed to build and maintain a well-organized, well-indexed and easily manipulated bank of information for the user.

File Structures

To effectively use an operating system, it is important to understand directories and file structures. User information is stored in memory in the form of **Files**. A file can be any size and can consist of virtually anything, from a single character to a novel.

Files are assigned names for easy storage and retrieval. These names are, appropriately enough, called **Filenames**. A filename must be a single word and can consist only of letters, numbers and periods. It cannot contain blank spaces, punctuation marks, or other keyboard characters. The first character of a filename must be a letter and the length must not exceed 15 characters.

Since many files can be stored on a disk, a method is needed for sorting and grouping files into categories. The operating system allows the user to store groups of files in units called **Directories**. A directory is analogous to a file cabinet drawer or a desk drawer which is used to store many individual files. Whenever you want to look at a file, you go the appropriate drawer or directory and retrieve the file.

Any number of directories that can be created and stored on a disk. However, to more efficiently organize the information stored on disks, it is sometimes useful to create directories within directories. These inner directories are called **Subdirectories.**

Directory and subdirectory names are subject to the same rules applied to filenames. They must not be longer than 15 characters and cannot contain blank spaces, punctuation marks (other than periods), or keyboard symbols (other than letters and numbers).

The system of storing files in directories and subdirectories is called a **hierarchical file system**. A hierarchical file system can be constructed to organize and hold massive amounts of information for rapid retrieval by the user and the computer.

Types of Files

The operating system recognizes several generic types of files.

Text Files are user files that contain numbers, letters, graphical symbols, application templates, or any other information intended to be read by the user and manipulated by the computer and an application program. Most files created by the user (e.g., word processing files, spreadsheets, data files) are text files.

Application Programs are sets of instructions which, through the operating system, direct the computer to perform specific tasks for the user. For example, word processors, spreadsheet programs, accounting software, are all application programs written and used to perform specific tasks. Application programs are not part of the operating system. However, they do run within the framework established by the system.

Utility Programs. The operating system stores information in files and directories. In order to create and maintain the file structure, tools are needed to perform such functions as copying and moving files, creating and deleting directories, etc. The operating system provides tools to do this, and these are called Utilities. ProDOS provides a small set of these utilities in the FILER. KIX extends the number of utilities available with a set of 25 UNIX-like utilities. A full implementation of the UNIX operating system includes more than 200 command utilities, many of which cannot be implemented on an 8-bit computer.

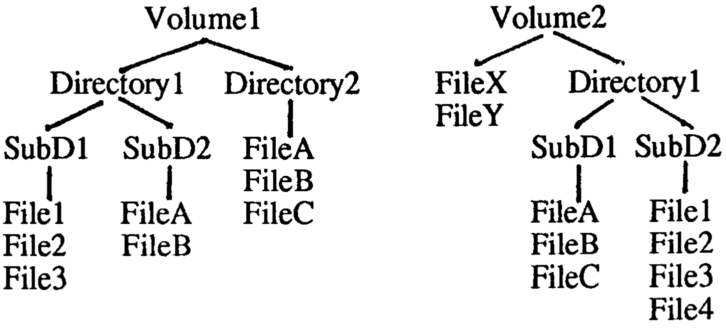
Pathnames

To store and retrieve files in the hierarchical system, the operating system needs a roadmap through the maze of directories and subdirectories. The device for supplying this roadmap is called a **Pathname.** A pathname specifies the exact path the operating system must take to reach a file.

A full pathname always starts with a slash ("/") and a volume name. This designates the **Root Directory** or **Volume Directory** and the beginning of the path to be followed.

Additional slashes are used in the pathname to separate the names (in descending order) of each directory and subdirectory encountered on the path to the file. The pathname ends with the name of the specific file being sought.

The following diagram illustrates the hierarchical file structure of a system that has two volumes (or disks) concurrently mounted on the system (i.e., Drive 1 and Drive 2).



Each disk has a Volume or Root Directory which contains a list of files and directories stored on the disk. The Volume Directory is the first and highest level in the hierarchical file structure.

In the example above, Volume1 has two directories in the Root Directory -- Directory1 and Directory2. Directory1 has two subdirectories -- SubD1 and SubD2. SubD1 contains three files and SubD2 contains two files. Directory2 contains three files and no subdirectories.

Volume2 has two files stored and one directory stored in the Root Directory. Directory1 contains two subdirectories: SubD1 contains three files; SubD2 contains four files.

The pathname to one of the files shown above would start with a slash ("/") and the name of the root directory, followed by slashes and the names of all directories between the root directory and the file. For example, the Pathname to File4 on Volume2 would be written:

/Volume2/Directory1/SubD2/File4

The pathname to FileC on the same volume would be written:

/Volume2/SubD1/FileC

The pathname to FileY on the same volume would be written:

/Volume2/FileY

Note here that there are no directories or subdirectories below the root directory.

For practice, try locating the following files on the diagram:

/Volume1/Directory2/FileA

/Volume1/Directory1/SubD2/FileB

/Volume1/Directory1/SubD1/FileB

/Volume2/Directory1/SubD1/FileB

Note how, in the last examples, the names of the directory, subdirectory, and files are identical. The operating system allows you to do this because, even though most of the names are the same, the full pathname for each file is different (i.e., the volume directory for each is different).

Working with Directories and Pathnames

Working with pathnames and the hierarchical system can be difficult at first. The following provides you with some tips which make it a little easier.

1. ProDOS limits pathnames to a maximum of 64 characters and limits the number of names in the Volume Directory to 51.

2. A directory that contains other directories or files is called a **Parent Directory**. The term is relative (no pun intended!) because a directory may be the parent of one group of files, yet itself have a parent directory. Directory1 on Volume1 illustrates this principle. Directory1 is the parent of SubD1 and SubD2 while it has the parent Volume1.

3. It isn't always necessary to specify a full pathname when working with complex file structures. A **Prefix** or **Working Directory** can be used to store the pathname to the directory in which you are working. In KIX the "CD" or Change Directory command is used to set the system prefix. For example, if you are working with the files stored in

Subdirectory1 (SubD1), in Directory1, on Volume 2, you can set the system prefix or working directory to /Volume2/Directory1/SubD1/. Then to move from FileA to FileB, you only need enter the filename (i.e., FileB).

4. If the prefix is set to the current working directory, you can call other files in the directory without specifying a full pathname.

5. To call a file which is not in the current working directory, you must use the full pathname for the file. The following examples illustrate this principle.

   To move from File1 in SubD1 to FileB in SubD2, you must indicate the full pathname of FileB. The pathname would be **/Volume1/Directory1/SubD2/FileB.**

   To move from FileB in SubD2 to FileA in Directory2, the pathname would be **/Volume1/Directory2/FileA.**

Once you understand pathnames, you will begin to appreciate and use the many KIX commands that let you set the working directory (prefix), create new directories, move and copy files within or between directories, compare files, or manipulate the system itself.

The KIX Operating Environment

KIX is a ProDOS-based operating environment for the Apple II. KIX is modeled after the user environment found in the UNIX operating system. It provides the user with many of the utilities found only on large, multi-user UNIX systems. KIX is designed to make use of the Apple II easier and more efficient.

It is important to recognize that KIX is not really a new operating system for the Apple II. If you examine the KIX disk, you will find ProDOS in a prominent position. And, when you boot the KIX disk, you will see the ProDOS copyright screen, notifying you that ProDOS is being loaded into the computer.

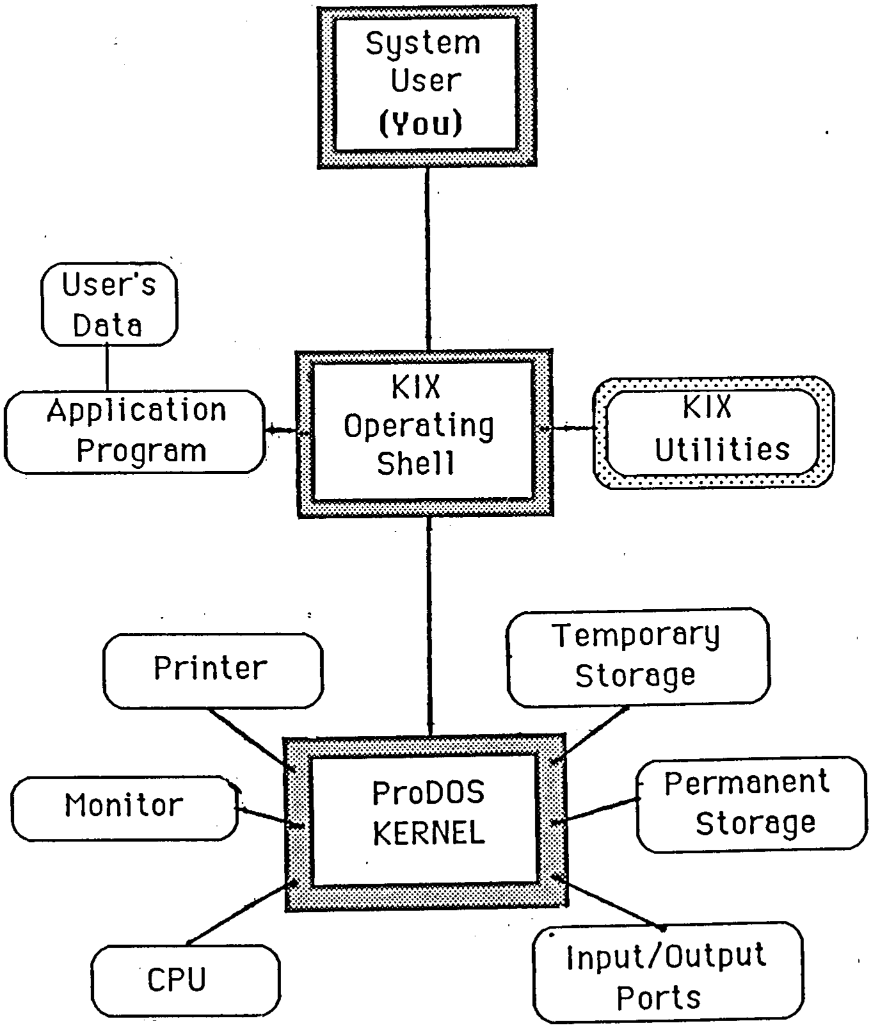
KIX is an operating system **Shell**. It establishes an interface between the user and the **ProDOS Kernel**. These two elements -- the Shell and the Kernel -- represent the outer and inner worlds of an operating system.

The ProDOS Kernel is the internal or invisible part of the operating system. It works automatically to manage the keyboard, disk drives, CPU, printers, and other hardware devices. The Kernel carries out instructions received from the KIX Shell.

The KIX Shell is the external or visible part of the operating system. It establishes the interface between the user, the user's files, and the computer hardware system. When the user enters a command, the shell interprets the command, calls the necessary utilities or application programs, and works with the ProDOS Kernel to carry out the user's instructions.

Since ProDOS is always present and performs all of the basic operating system functions, KIX is fully compatible with all ProDOS functions and ProDOS-based software.

KIX OPERATING ENVIRONMENT



KIX Shell

The KIX shell and user interface is created by the file named KIX.SYSTEM found on the KIX disk. When the KIX disk is booted, the ProDOS Kernel loads first and then KIX.SYSTEM. You can always tell when the KIX shell is active because you will see the percent (%) prompt. This symbol tells you that "KIX is ready to accept and execute your commands."

If you have exited KIX to the ProDOS environment, you will see the arrow prompt (">"). To return to the KIX environment from ProDOS, you must rerun the KIX.SYSTEM program.

BIN Directory

The disk structure for KIX is similar to UNIX in that all command utilities are stored in a directory named BIN. If you look at the KIX disk directory, you will see the following listing.

```
/KIX
   ProDOS            SYS
   KIX.SYSTEM        SYS
   BIN               DIR
```

When the KIX shell accepts a command from the user, it interprets the command and calls the utilities or programs required to execute the command. If a KIX utility is called, KIX automatically locates the utility in the BIN directory and runs it. If an application program is called, KIX follows the pathname specified and then loads and runs the program.

KIX Command Syntax

KIX consists of more than 25 UNIX-like commands or utilities. To call these utilities, it is necessary to communicate with the KIX shell using a standard format or syntax. The KIX command syntax takes the following form:

Utility_-Options_Arguments

The **Utility** portion of the KIX command syntax is the abbreviated name of the command utility you want to invoke (e.g., MV to rename a file, CP to copy a file). These utilities are described later in this manual.

The **Options** portion of the command syntax tells the utility program to perform the command in a specific way. Many KIX commands have options associated with them. For example, the "LS" command (List Directory) provides options for: standard or extended directories; alphabetized file names; listing of file protection status; and more. By using options, the user can customize the output of the KIX utility to meet specific needs. You can specify as many options as desired. If no options are specified, KIX assumes a set of default (i.e., predefined) options.

The **Argument** portion of the KIX command syntax is a pathname. The argument defines the file or directory name(s) associated with the command. For example, if a command is given to delete files on a disk, the argument would specify the pathname of the file or files (more than one can be specified) to be deleted. If a Move command is given, two arguments would be provided; the first would indicate the pathname of the file to be moved, and the second where the file is to be moved.

The following rules govern the syntax of KIX commands:

1. Either upper or lower case letters can be used to specify KIX utilities, options or arguments.

2. A single space must always separate the Utility from the Option(s) and the Options from the Arguments. (Note: In this manual a blank space is denoted by an underscore character).

3. Options and Arguments may be listed in any order.

4. Options are always preceded by a dash (-) so that KIX can distinguish between options and arguments.

5. There is no limit to the number of options which can be specified with a KIX command. However, a space and a dash must precede each option.

Redirection

Redirection is one of the more powerful features of the KIX environment. When certain commands are given to the KIX shell, the redirection optior allows you to save the results by printing a listing or writing a file to a disk. Redirection allows you to direct the output of the command to a specified slot number (e.g., printer, modem, or other peripheral) or to a pathname (i.e., save the output as a text file). The redirection option is especially useful for such things as printing a hard copy of a disk directory, merging text files, and printing file listings.

KIX Command Structure

KIX Command Utilities are divided into seven categories. Each category is described in a separate chapter of this manual. The categories and the utilities included in each are summarized below. (Note: command options are not listed).

Directory Management Commands

Directory management commands consists of five utilities used to create and manipulate directories. The command syntax and descriptions are:

Command Syntax	Description
CD_*pathname*	Change Working Directory
LS_*pathname*	List Directories and Files
MKDIR_*pathname*	Make a New Director
PWD	Print Working Directory
RMDIR_*pathname*	Delete a Directory

File Management Commands

File management commands consists of six utilities used to copy, rename or otherwise manipulate files. The command syntax and descriptions are:

Command Syntax	Description
CAT_*pathname*	List the Contents of a File
CHMOD_*pathname*	Change Protection Status
CP_*source* _*destination*	Copy a File
LPR_*pathname*	Print a File
MV_*old.name*_*new.name*	Move or Rename a File
RM_*pathname*	Delete a File or Directory

Volume Management Commands

Volume management commands consists of three utilities used to copy, format, and rename volumes. The command syntax and descriptions are:

Command Syntax	Description
CPV_(slot,drive)_(slot,drive)	Copy a Volume
FORMAT_(slot,drive)_*/name*	Format a Volume
MVV_(slot,drive)_*/name*	Rename a Volume

Special KIX Commands

Special KIX commands consists of five utilities used to compare files and volumes and to find strings.

Command Syntax	Description
CMP_*file.1*_*file.2*	Compare Two Files
CMP_(ss,sd)_(ds,dd)	Compare Two Volumes
FIND_*directory*_-*filename*	Locate a File in a Directory
GREP_*string*_*pathname*	Locate a String in a File(s)
SDIFF_*file.1*_*file.2*	Compare Two Text Files

Abbreviations and Wildcards

Abbreviations and wildcards greatly enhance the power and ease of use of KIX commands. The following are used in KIX.

Command Syntax	Description
. (Abbreviation)	Working Directory
.. (Abbreviation)	Parent Directory
? (Wildcard)	Single Character
* (Wildcard)	Character String
ECHO_*pathname*	List all pathnames affected by Wildcard usage

Firmware Calls

The KIX firmware calls consist of four utilities used to enable and disable the 80 column firmware, set the system time and date, and dump the contents of the screen to the printer. The command syntax and descriptions are:

Command Syntax	Description
C40	Disable 80 Column Card
C80	Enable 80 Column Card
Date_*yymmddhhmm*	Set/Read System Date/Time
SD	Print Screen Contents

KIX Utility Commands

KIX Utility commands consists of four utilities used to setup and configure KIX for the user's hardware and software. The command syntax and descriptions are:

Command Syntax	Description
CFG	Configure KIX
INSTALL	Install KIX
KIX	KIX Command Summary
QUIT	Exit KIX to ProDOS

II. GETTING STARTED

KIX is designed to run on any Apple II computer with at least two disk drives and 64K of memory. The software uses Apple Computer's ProDOS operating system which is included on the KIX disk. This chapter describes:

* Creating a backup copy of KIX
* Configuring KIX for your computer.
* Installing KIX in AppleWorks

Creating a Backup Copy

KIX consists of a single "flippy" diskette (i.e., software is copied on both sides of the disk). **Before you use KIX, you should make backup copy of both sides.**

It is easy to make these backup copies using the KIX "Copy Volume" (CPV) command.

1. Boot Side 1 (KIX Operating Shell) of the KIX disk in drive 1.
2. Place a blank disk in drive 2.
3. When the KIX prompt (%) appears, enter the following KIX command sequence:

```
% CPV_(6,1)_(6,2)  <RETURN>
```

   The KIX "Copy Volume Command" (CPV) will automatically format the disk and copy the contents of the drive 1 disk onto the drive 2 disk.

4. Flip the KIX disk over and repeat the procedure to make a backup copy of AppleWorks/KIX (Side 2).

Configuring KIX

KIX is designed to run on the full spectrum of Apple II systems. The software will work on an Apple II+ with 64K as well as on an enhanced Apple IIe with extra RAM and a hard disk drive. To insure optimum performance on your computer, it may be necessary to change some of the default configuration settings. The KIX configuration utility (CFG) is used for this purpose.

The CFG (configure) program lets you customize KIX for use on your system. It modifies the default values stored on the KIX boot program (KIX.SYSTEM) which is found on the system disk. To modify the default configuration values, type CFG when you see the system prompt (%). The following Configuration Menu will appear.

```
*************************************************

  Path: /KIX/BIN              KIX Configuration Utility
  -----------------------------------------------------
                      MAIN MENU

          1       Display/Change Startup Conditions

          2       Display/Change KIX Printer Settings

          3       Save this Version of KIX to disk.

          4       Quit

  -----------------------------------------------------
 Type Number or use arrows, then press <Return>   Press -? for HELP

*************************************************
```

The Configuration Utility is divided into two parts -- Startup Conditions and Printer Settings. The utility works just like AppleWorks with menu cards and highlight bars.

Select Item 1 if you need to change the KIX startup conditions. Select Item 2 if you need to change the Printer settings. Select Option 3 to install the changes you have made. Select Option 4 to cancel the configuration program and return to the system prompt.

Note: The KIX system disk contains two configuration utilities -- CFG40 and CFG80. When you call CFG, KIX checks to see if there is an 80 column card active in the system. If there is, CFG80 is loaded; if not, CFG40 is loaded. When CFG is executed on an Apple // or //+, CFG40 is automatically loaded.

Main Menu - 1: Startup Conditions

The KIX startup options and their default values are:

Option	Description	Value
1.	Copy KIX Files to /RAM	YES
2.	Enable 80 Column Card	YES
3.	Display/Specify System Date	YES
4.	Disable Lower Case Output	NO
5.	Use Linefeeds on Printer	NO
6.	Set Printer Interface Codes	----

Option 1. KIX and /RAM

KIX is compatible with most extended 80 column cards available for the Apple //. These cards contain 64K (or more) of additional RAM which can be used as a "RAMdisk".

RAMdisk is electronic memory that is set-up to act like a disk drive; it works just like another disk drive in the system. However, since the memory is electronic, data can be stored and retrieved much more quickly than with an external, magnetic memory disk drive.

To make it easier to use your computer's RAMdisk capability, KIX contains a feature which allows you to automatically load the KIX utilities into RAMdisk when KIX is booted.

If the RAMdisk auto-load option is active, KIX automatically creates a BIN directory in /RAM (the RAMdisk volume name) and copies the files in the KIX/BIN directory into /RAM. KIX will continue to move files into /RAM/BIN until there are no more KIX commands small enough to fit into the remaining space or until all KIX commands are loaded. With the KIX utilities in RAMdisk, the computer responds very quickly to KIX commands.

If the autoload option is active and you do not want the KIX utilities copied into RAM, simply press the <ESCAPE> key and the copy process will be terminated.

If you do not want KIX utilities to be automatically loaded when KIX is booted, you should change Option 1 to NO.

NOTE: ProDOS only recognizes 64K of RAMdisk. However, if you have more RAM available and want to use it for KIX commands and other files, you can prepare a preboot file using the installation program furnished by the manufacturer of your RAM card. Please see the Technical Reference Section (Section XI) for more information.

Option 2. 80 Column Cards

The second startup option instructs KIX to enable the 80 column card when KIX is booted on an Apple //e or //c. If you do not want this card enabled, you should set this option to "NO".

NOTE: The 80 column card will be automatically enabled only if KIX is booted on an Apple //e or //c. If it is booted on an Apple // or //+, this selection is ignored and KIX boots in 40 column mode. To activate the 80 column card on a // or //+, you must call the "C80" command at the KIX prompt.

Option 3. Display/Specify System Date on Startup

This option prompts you to enter the current date when KIX is booted. If a date is provided, all files created or modified during the session will be stamped with the current date. This option is useful to those who do not have a clock/calendar card installed in their computer.

If you have a clock card installed in your computer, or you don't want your files date-stamped, this option should be changed to NO.

NOTE: This option cannot be used to set or reset a clock/calendar card. The KIX "DATE" utility can be used for this function.

Option 4. Disable Lower Case Output

This option is available to users with an Apple // or //+ computer and VIDEX 80 column card or compatible. It enables or disables lower case output.

Option 5. Use Linefeeds on Printer

The default setting for this option should be changed if your printer does not automatically perform a linefeed after printing each line when using the CAT, LPR, or SD commands. If your printer doesn't advance, change this option setting to "YES".

Option 6. Set Printer Initialization Code

The default printer initialization code is:

<Control> I-80N

If your printer does not initialize properly or does strange things when you print, try changing this code to:

<Control> I-0N

If you still have problems, consult your printer owners manual for the correct initialization code and enter it under Option 6 of the configuration program.

Main Menu - 2: Printer Settings

KIX allows you to specify information about your printer and the printing format you want to use with the KIX commands. This information is stored in the KIX boot program (KIX.SYSTEM) and called when the LPR, SD, and CAT commands are used or when the redirection option is selected.

You can change the output slot number, printing margins, type size, and line spacing of the printer output. You can also specify unique control characters required for your printer. The KIX printer options and their default values are:

Option	Description	Value
1.	Printer Slot Number	1
2.	Characters per inch	10
3.	Specify Control Codes	----
4.	Line Spacing	Single
5.	Top Margin (inches) ..	1.0
6.	Bottom Margin (inches) .	1.0
7.	Left Margin (inches)	1.0
8.	Right Margin (inches)	1.0

To change a default value, type the option number. You will be then be prompted for a new value. Please note the following:

1. Margin settings are limited to half-inch increments and to a maximum of 2.0 inches.
2. The Printer Initialization Code cannot exceed 6 characters in length.
3. A maximum of 8 printer initialization characters can be specified.

NOTE: Before changing any values, check your owners manual to make certain that your printer supports the changes you want.

If you don't have a printer, specify slot 0 (zero) for the Printer slot number. Slot 0 tells the KIX interpreter that you don't have a printer and to ignore SD and LPR commands.

Main Menu - 3: Save this Version of KIX

When you are have finished configuring KIX for your computer, select option 3 from the Configuration Menu. CFG will store the changes in the KIX.SYSTEM file and give you instructions for powering down the computer and rebooting the system. By resetting the system in this manner, you are assured that the new configuration options are properly installed on all copies of the files (particularly those loaded in RAMdisk).

NOTE: Configuration changes are installed only on the volume which was booted to load KIX (i.e., if you have multiple copies of KIX on line in different volumes, the changes you specify will only be installed on the copy which is found on the volume which was booted).

Main Menu - 4: Quit

If you change your mind after calling the CFG program and want to quit, select Option 4. This option terminates the configuration program and returns you to the system prompt.

Installing KIX

The KIX "Install" program is used to copy KIX onto a hard disk (or 3.5 Unidisk) and to install KIX in AppleWorks. The Install program is simple to use and provides prompts for each step in the installation process. Before using the Install program, be sure that you have configured the KIX disk as described in the preceding section.

Boot side 1 of the KIX disk (your configured copy). Then insert side 2 of the KIX disk and call the Install program:

```
        % INSTALL                    <RETURN>
```

The Install program will load and the following menu will appear.

```
**************************************************

                KIX Installation Utility
----------------------------------------------------------------------
      Main Menu

         1. Introduction to KIX operating environment.
         2. Install KIX on another volume.
         3. Install KIX in AppleWorks.
         4. List information about volumes on line.
         5. Quit.

----------------------------------------------------------------------
Type Number or use arrows, then press <RETURN>    Press -? for HELP

**************************************************
```

Remove Side 2 and reinsert Side 1 of the KIX disk.

Option 1 - Introduction to KIX

Select Option 1 for a brief description of the KIX operating environment.

Option 2 - Install KIX on Another Volume

Select Option 2 if you have a hard disk drive or a Unidisk 3.5. By installing a copy of the KIX Operating Shell (Side 1) on one of these large volumes, you are assured that the KIX utility files are always accessible to thc KIX interpreter. You won't have to concern yourself with swapping disks.

When you select option 2, you will be asked to specify the device (i.e., the drive number) on which you want KIX installed. Once this is done, KIX will automatically create a BIN directory and copy itself onto the volume you selected. The installation is now complete.

Note: In case you are wondering about Side 2 of the KIX disk, all of the necessary KIX files are contained on Side 1. Side 2 contains the Install program and a special configuration of KIX which is used with AppleWorks. You do not have to copy Side 2 onto the hard disk.

Option 3 - Install KIX in AppleWorks

Option 3 installs KIX in AppleWorks. With this option, you simply press -K to save the AppleWorks desktop and open a KIX window. You now have access to all KIX commands. When you are finished with KIX, type QUIT, and the AppleWorks desktop will be restored.

Caution: Do not attempt to install KIX on your original AppleWorks source disk. Always use a backup copy!

The following procedure is followed to install KIX in AppleWorks. Before installing KIX on your AppleWorks disk, please read all of these instructions and the installation notes which follow.

1. Select Option 3.

2. At the prompt, enter the device (i.e., slot and drive number) on which the KIX utility files are stored. If you have copied KIX onto a hard disk or Unidisk 3.5, then specify the location of this drive (e.g., slot 7, drive 1). If you are using a floppy, insert your copy of the AppleWorks/KIX disk (Side 2) into any available disk drive and specify the location of the drive you selected.

Note: If you don't remember the slot and drive numbers, press <ESCAPE> and select Option 4 from the Install Menu. KIX will automatically scan all devices on line and list the numbers for you.

3. When prompted, confirm the name of the volume in the device you have specified (e.g., /KIX).

4. At the prompt, enter the device where your copy of the AppleWorks Startup disk is mounted. If you are using a hard disk drive and have installed the AppleWorks startup program (AplWorks.System) in a subdirectory, select item 7 from the device menu and then enter the name of the directory (e.g., /Hard1/AppleWorks or /Profile/AppleWorks).

5. When prompted, confirm the name of the volume in the device you specified (e.g., /AppleWorks)

The KIX Install program permanently modifies your copy of the AppleWorks Startup disk. Whenever you boot AppleWorks, you will see the KIX copyright notice appear briefly on the screen.

Installation Notes:

1. Many people use a desktop expander (i.e., Applied Engineering, CheckMate, or other RAM expansion card) which automatically loads AppleWorks into RAM and/or expands the desktop. These desktop expanders also require a small modification to the AppleWorks Startup disk. If you use one of these expanders, make sure that you install the expansion software on your AppleWorks Startup disk **before** you install KIX.

2. If you are going to use KIX with a RAM expansion card, you must use Desktop expansion software which is Version 4.3 or later. If you have an earlier version of this software, contact the manufacturer of the RAM card for updated software.

3. Due to the AppleWork's use of memory, KIX is incompatible with PinPoint, MacroWorks, and most other AppleWorks desktop accessories.

4. The KIX Install program cannot be run twice on the same copy of the AppleWorks Startup disk. If you make a mistake or need to rerun the Install program, always use a new copy of AppleWorks (and be sure the install the Desktop expansion software for your Ramcard first).

5. The KIX modification to the AppleWorks Startup disk in no way changes the copyright of the AppleWorks software.

NOTE: If you would like more information about how KIX works inside AppleWorks, please refer to the Technical Reference section (Section XI).

Option 4 - List information about volumes on line.

Select this option to list the volumes currently on line. If you don't remember the device numbers requested in Option 3, this option will help you determine them.

This option invokes a KIX "LS" command and lists the slot and drive number, volume name, and blocks free for each volume on line. You will learn more about the "LS" command in the next section of this manual.

Option 5 - Quit

Select Option 5 when you are through installing KIX in your system and are ready to Quit. You will returned to the KIX system prompt, ready to call your first KIX utility or application program.

III. USING KIX

With KIX you have a great deal of power at your fingertips. A few simple keystrokes allow you to perform a wide range of activities. This section describes:

- o using KIX with a floppy disk system,
- o using KIX with a hard disk system,
- o using KIX in AppleWorks,
- o uses for KIX with files, volumes and directories,
- o useful KIX commands, and
- o general notes for using KIX.

This section is intended to give you some general guidelines for using KIX commands. For simplicity, command options are not included. The complete specifications and list of options for each KIX command can be found in other sections of this manual. Also, a command summary can also be found on the enclosed "Quick Guide to KIX".

The best way to learn KIX is to use it. The process is like learning to drive a car. You should start slowly and try new things as you gain experience and confidence. Experiment with different commands. But, please be careful! It's like learning to drive in a powerful sports car; if you don't pay attention, you can end-up in trouble (or with a reformatted hard disk!). To play it safe, you should experiment using a duplicate copy of one of your disks. That way if you make a mistake, nothing is lost.

Good luck. Now start your engine!

Using KIX on Floppy Drive Systems

KIX can be used on either a single or multiple disk drive system. The instructions are basically the same for both; the primary difference is in the amount of disk swapping required.

KIX is simple to use with a floppy disk drive system. When you turn-on your computer, always boot the KIX Operating Shell disk (Side 1) first. The KIX utilities will load into RAMdisk (if available), and the KIX prompt will appear. From this prompt you can call a KIX command or launch an application program.

To launch an application program, insert the application disk in a drive and enter the full pathname for the system file. For example, to run AppleWorks you would use the following pathname:

```
% /AppleWorks/AplWorks.System    <RETURN>
```

When running an application, you can remove the KIX disk from the drive. This allows you to have your program disk in one drive and your data disk in another. When you quit the application, you will automatically return to the KIX prompt.

You can call a KIX command whenever you see the system prompt (%). Just use the standard KIX command syntax:

```
% Command.Name_-Options_Arguments
```

When the command utility is finished, the system prompt will reappear, waiting for the next command or application pathname.

If the command utility is stored in RAMdisk, you do not need to have the KIX disk on line. If it is not in RAMdisk, then you must be sure to insert the KIX disk (Side 1) in an available disk drive.

Note: If the KIX interpreter is unable to find the KIX utility that you specified, a "file not found" error message will be returned. When this occurs, check to see that you have the KIX volume in one of the disk drives. If you do and the error still occurs, try entering the full pathname for the KIX utility (e.g., /KIX/BIN/LS).

That's about all there is to using KIX. You can now skip to the "Using KIX with Files, Directories, and Volumes" to learn more about the uses of KIX.

Using KIX on a Hard Disk System

With a hard disk or Unidisk 3.5, you have full access to all the capabilities of KIX without any disk swapping. All of the KIX commands are stored in the BIN directory on your disk drive.

You can have KIX boot automatically from your hard disk by making KIX.SYSTEM the first system file in the volume directory. Otherwise, you can:

1. Launch KIX from Catalyst, MouseDesk, or other menu type program. You should follow the publisher's instructions for installing KIX in one of these programs.

2. Load and run KIX from the ProDOS Quit screen or ProDOS prompt by entering:

   /Volume.Name/KIX.System

3. Load and run KIX from BASIC.SYSTEM by entering:

   -/Volume.Name/KIX.System

When you launch KIX, the KIX command utilities will automatically load into RAMdisk (if the option is active), and the system prompt will appear. You can now call a KIX command or launch another application program. When you finished with KIX and type "Quit", you will be returned automatically to the launching program or menu.

You can call a KIX command whenever you see the system prompt (%). Just use the standard KIX command syntax:

% **Command.Name_-Options_Arguments**

When the command utility is finished, the system prompt will reappear, waiting for the next command or application pathname.

That's about all there is to using KIX. You can now skip to the "Using KIX with Files, Directories and Volumes" to learn more about the uses of KIX.

Using KIX with AppleWorks

If you have installed KIX in AppleWorks, you open a KIX window by pressing "-K". When the system prompt appears, you can enter any KIX command except those listed below in Table III-1. To return to AppleWorks, type "Quit"; the KIX window will be closed, and the AppleWorks desktop restored.

When using KIX with AppleWorks on a floppy disk system, you need to use the AppleWorks/KIX side of the disk (Side 2). This side is configured specifically for AppleWorks and contains all of the KIX commands which can be called from AppleWorks.

Note: Side 2 of the KIX disk also contains a file named "KIX.Desktop"; this file is used to save the AppleWorks desktop when you call KIX. Do not tamper with this file; doing so may result in the loss of your AppleWorks desktop.

It is not necessary to keep the KIX disk mounted when you are using AppleWorks. However, when you want to open the KIX window, you must remove one of the mounted disks (it doesn't matter which one) and insert the AppleWorks/KIX disk. When you are finished with KIX, remove the KIX disk and re-insert the AppleWorks disk.

Table III-1. KIX Commands Not Supported in AppleWorks

Command	Function
C40/C80	Enable/Disable 80 Column card
CFG/40/80	Configure KIX
CHMOD	Change File Protection Status
DATE	Set System Time and Date
FORMAT	Format a Disk
GREP	Search for a String
LPR	Print
SDIFF	Compare Files

Using KIX with Files, Directories and Volumes

Working with Files

When working with files, KIX can be used as follows:

Print A file can be printed using the **LPR** command or the CAT command.

LPR will print the file you specify, following the format defined in the configure (CFG) program.

CAT is a more powerful printing command. It allows you to (1) print more than one file at a time; (2) direct output to the screen, slot number, or disk file; and, (3) select from a long list of control and format options.

The following examples illustrate the use of both commands.

(Note: In all examples, the underscore character represents a single blank space.)

```
% LPR_/VolumeName/File1
```

Prints the file named File1 on the printer defined in the CFG program.

```
% CAT_/VolumeName/File1_>2
```

Prints File1 on the device in slot 2

Copy

A file can be copied using the **CP** command. List the pathname of the file to be copied, followed by the pathname of the new file.

```
% CP_/VolumeName/File1_/VolumeName/File2
```

Makes a copy of File1 and renames it File2.

Move

A file can be moved from one location to another using the **MV** command. List the pathname of the file to be moved, followed by the pathname of the new location (if you want, you can change the name of the file at the same time). When a file is moved, the original copy is automatically deleted.

```
% MV_/Volume1/File1_/Volume2/File1
```

Moves File1 from Volume1 to Volume2. File1 is automatically deleted from Volume1.

```
% MV_/Volume1/File1_/Volume2/File2
```

The same command as before except that the new file is called File2 instead of File1.

Delete

A file can be deleted using the **RM** command. List the pathname of the file to be deleted. More than one file can be deleted at a time by listing the pathname of each file after the command.

```
% RM_/Volume1/File1
```

Deletes File1 from Volume1.

```
% RM_Volume1/File1_Volume2/File3
```

Deletes File1 from Vol.1, File3 from Vol.2.

Rename A file can be renamed using the **MV** command (yes, the same command used to move a file). List the pathname of the file to be renamed, followed by the pathname with the new name.

```
% MV_/Volume1/File1_Volume1/File2
```

Renames File1 on Volume1 to File2.

Merge Two or more files can be merged into one using the **CAT** command. List the pathname of each file to be merged (concatenated), followed by instructions for where the merged file should be sent (i.e., printer, screen, or disk file).

```
% CAT_/Vol1/File1_/Vol2/File2_>/Vol2/File3
```

Merges File1 on Volume1 and File2 on Volume2 in a new file named File3 on Vol2.

```
% CAT_/Vol1/File1_/Vol2/File2_>1
```

Merges File1 and File2 and sends the resulting file to the device (e.g., printer) in slot 1.

Compare Two files can be compared with each other using the **CMP** command or the **SDIFF** command.

The **CMP** command finds the first difference between two files and reports the location of that difference. List the pathname of each file to be compared; output from the command is automatically sent to the screen. CMP can be used with any type of file.

The **SDIFF** command finds the first three differences between two text files and prints the lines containing the differences. SDIFF only works with text files.

```
% CMP_/Volume1/File1_/Volume2/File1
```

Compares two files with the same name on two different volumes. The location of the first difference is reported.

```
% SDIFF_/Volume1/Letter1_/Volume2/Letter2
```

Compares two letters with the same name on two different volumes. The lines containing the first three differences in the letters are printed.

Find The **FIND** command searchs through volumes and directories for a filename. List the pathname(s) to be searched and the name of the file being sought. The full pathnames will be printed for each file found with the name specified.

```
% FIND_/Volume1_-File1
```

Searches Volume1 and all directories in the volume for files with the name File1. The full pathname for each file named File1 will be printed on the screen.

Protect Files can be read-protected, write-protected, rename-protected, and/or delete-protected using the **CHMOD** command. File protection status is changed by listing the pathname of the file followed by the specified change in protection status: ± R for Read protection; ± W for Write protection; ± D for Deletion protection; and ± N for reName protection. The plus symbol deactivates protection; the minus symbol activates protection.

```
% CHMOD_/Volume1/File1_+R_-N_+D
```

Removes read and delete protection from File1 and adds rename protection.

Working with Directories

When working with directories, KIX can be used as follows:

Create A volume directory or subdirectory can be created using the **MKDIR** command. Just list the pathname of the directory to be created.

```
% MKDIR_/Volume1/Dir1
```

Creates a new directory on Volume1 called Dir1

Delete A volume directory or subdirectory can be deleted using the **RMDIR** command or the **RM** command.

The **RMDIR** command can be used only if the directory is empty (i.e., it contains no files or subdirectories). More than one directory can be deleted at the same time using this command. Just list the pathname of each directory to be deleted.

The **RM** command can be used to delete a directory and all of the files and subdirectories included in it. It is a very powerful command and should be used with caution. List the pathname of the directory to be deleted, followed by a "-R" option. The directories and all of its contents will be deleted.

```
% RMDIR_/Volume1/Dir1
```

Deletes the directory on Volume1 called Dir1 (assuming Dir1 is empty).

```
% RM_/Volume1/Dir1_-R
```

Deletes Directory1 on Volume1 and all of the files and subdirectories included in it.

List

The contents of a directory can be listed using the **LS** command. To generate a short listing of the directory (i.e., one which lists only the filename and filetype), use just the LS command without any options. To generate a long listing (i.e., one which lists filename, filetype, size, creation date, etc.), use the LS command with a "-L" option. (Section IV contains more information about short and long listings and other LS options). The listing can be directed to the screen, printer, or disk file using the redirection option.

```
% LS_/Volume1/Directory1
```

Prints a short listing to the screen of the files in Directory1 on Volume 1.

```
% LS_/Volume1/Directory1_-L_>1
```

Generates a long listing of the files in Directory1 and directs output to the device (e.g., printer) in slot 1.

Working with Volumes

When working with volumes, KIX can be used as follows:

Copy A volume can be copied in its entirety using the **CPV** command. Just list the slot and drive number of the source volume, followed by the slot and drive number of the destination volume. If the new volume is unformatted, it will be formatted automatically. If the new volume is not empty, you will be asked if you want the contents destroyed

```
% CPV_(6,1)_(6,2)
```

Completely replaces the contents of the disk in slot 6, drive 2, with the contents of disk in slot 6, drive 1.

Format A volume of any size can be formatted using the **FORMAT** command. List the slot and drive number of the volume, followed by the name of the new volume. If the disk is not empty, you will be asked if you want the contents destroyed.

```
% FORMAT_(6,1)_/NewVolume
```

Formats the disk in slot 6, drive 1, and names the new volume, NewVolume.

Rename A volume can be renamed using the **MVV** command. The volume does not have to be empty since MVV does not reformat the disk. List the slot and drive number, followed by the new volume name.

```
% MVV_(6,2)_/NewVolumeName
```

Renames the volume in Slot 6, Drive 2 to NewVolumeName.

Compare Two volumes can be compared using the **CMP** command (this is the same command used to compare two files). The volumes are compared on a byte by byte basis until a difference is located. The location of the differing byte is then reported on the screen. To compare two volumes, list the slot and drive number of the first volume, followed by the slot and drive number of the second volume.

```
% CMP_(6,1)_(6,2)
```

Compares the volume in slot 6, drive 1, with the volume in slot 6, drive 2. The difference is reported as the location of the differing byte.

List The contents of a volume can be listed using the **LS** command. To generate a short listing of the volume directory (i.e., one which lists only the filenames and filetype), use just the LS command without any options. To generate a long listing (i.e., one which lists filename, filetype, size, creation data, etc.), use the LS command with a "-L" option. (Section IV contains more information about short and long listings and other LS options). The listing can be directed to the screen, printer, or disk file using the redirection option.

```
% LS_/Volume1
```

Generates a short listing of the files and directories in the root directory of Volume1.

```
% LS_/Volume1_-L_>1
```

Generates a long listing of the files and directories in the root directory of Volume1 and redirects the output to the device (e.g., printer) in slot 1.

Other Types of KIX Functions

Find a String: The location of a character string in a text file can be determined using the **GREP** command. The search can encompass multiple files, directories and volumes. Each occurence of the search string is reported on the screen with the full pathname of the file containing the string and the line of text containing the string. The search is initiated by listing the search string followed by the pathname(s) of the files to be searched.

```
% GREP_'Kyan_Software'_/Volume1
```

Searches and reports every occurence of the string "Kyan Software" in files on Volume1.

Date-Stamp Files: The creation and modification date of files can be recorded on the file using the **DATE** command. By setting the system date, all files loaded during the session will have the current date recorded and listed in the directory. The date is entered by listing the month-day-year without blank spaces.

```
% DATE_070186
```

The system is set to July 1, 1986 and all files opened during this session will carry this date.

Print Screen Contents: The current contents of the screen can be printed using the SD command. The printer, slot location, and printing format is specified in the Configuration (CFG) program.

```
% SD
```

Prints contents of screen on printer.

KIX All the KIX commands plus options and syntax can be listed on the screen using the **KIX** command. This HELP program is intended to supplement the "Quick Guide to KIX".

```
% KIX
```

Lists the KIX commands and options on the screen.

C40
C80 The 80 column card in the Apple // can be deactivated using the **C40** command. The card can be reactivated using the **C80** command.

```
% C40
```

The screen shifts to 40 column mode.

```
% C80
```

The screen shifts to 80 column mode.

Wildcards: The power of KIX is significantly enhanced through the use of wildcards. KIX supports the "?" wildcard which is a substitute for a single character and the "*" wildcard which is a substitute for a string of characters.

The "?" wildcard is used to represent any single character. When used in a command line argument, you can capture a whole list of files, directories, or volumes with the use of a single wildcard character. For example, the pathname

/Volume?

is equivalent to /Volume1, /Volume2, /VolumeX, etc., and all other pathnames which consist of a "/Volume" followed by a single character.

The "*" wildcard is a more powerful character substitute and is used to represent any string of characters, including a null or empty string. For example, the pathname

```
/Volume*
```

is equivalent to all the files listed previously plus all the other files and directories which begin with "/Volume", regardless of the length of the pathname.

The use of wildcards with KIX commands is described more completely in Section VIII of this manual.

The preceding has described the general use of KIX commands and provided some sample applications. Following are additional examples of how KIX commands can be used.

Useful KIX Commands

This section provides many examples that will further your understanding of the KIX command structure.

KIX Command	Action Taken
RMDIR_FILE.?	Delete all empty directories whose filenames begin with **FILE.** and have no extensions or a single letter extension.
LS_-L_*.S_>LSFILE	List, using the long format, all files which end in ".S". Save this listing as a text file "LSFILE".
LS_/_>1	List the volume names, slot and drive locations, and blocks free for each volume on line. Direct this list to the printer in slot 1.
LS_/ANYVOL	List the names of all files and directories stored in volume directory /ANYVOL.
LS_/ANYVOL/*	List all the files stored in volume /ANYVOL; expand all subdirectories stored in the volume directory ANYVOL.
LS_/ANYVOL/*_;SYS	List the names of all system files stored on the volume /ANYVOL; expand all subdirectories.

KIX Command	Action Taken
LPR_/AVOL/APL*TXT	Search the directory AVOL and print those files with names beginning with "APL" and ending with "TXT".
CAT_WORK/*.O_>FINAL.O	Concatenate all files in the directory "WORK", whose names end in ".O", and save the results in the file "FINAL.O".
CP_/AVOL/*_/Vol2/Dir1	Copy all files in volume "AVOL" into Dir1 on volume Vol2.
CP_/VOL1/FILENAME	Copy /VOL1/FILENAME into the Working Directory using FILENAME as the new name.
MV_FILE1_FILE2	Rename FILE1, FILE2.
MV_D1/F1_D2/F2_D3/F3 .	Move files F1, F2, F3 from their separate directories into the present Working Directory keeping the same filenames.
CHMOD_-w_-n_/V1/DIR1/*	Activate write and rename protection on all files in the directory /V1/DIR1.
FORMAT_(6,1)_/NEWVOL	Format the disk in slot 6, drive 1, and name the volume NEWVOL.
CPV_(6,1)_(6,2)	Copy the volume in slot 6, drive1 to the disk in slot 6, drive 2.

KIX Command	Action Taken
FIND_/_-MYFILE	Search all volumes for the file, "MYFILE". Unlike other volume arguments, / used with the FIND command extends the search to every directory in every mounted volume.
FIND_/V*_-LOSTFILE	Search all volumes whose name begins with "V" for the file named "LOSTFILE". The search extends to all directories in those volumes.
GREP_apple_*	Search all text files in the present Working Directory for the string "apple" or "APPLE" or any upper/lowercase combination of those characters.

General Notes on Using KIX

1. If you press the <ESCAPE> keys during the execution of any KIX command, you will be returned to the KIX prompt. The message "ESCAPE KEY PRESSED" and the name of the aborted command will be printed on the screen.

2. On an Apple IIe or IIc, you have an additional command line editing feature. When entering a KIX command, you can type <CONTROL>-X to erase the entry and start over again.

3. KIX command options and output redirection characters can be placed in any order on the input line (just make sure that pathnames are properly located). In addition, options can be specified in groups following a single - (minus) character . For example,

```
    % LS_>1_/VOLUME1/DIR1_-LP
```

prints a Long listing of the files included in directory /VOLUME1/DIR1. The listing is directed to the printer in slot 1 and the protection status of each file is printed.

4. If the KIX autoload feature is active when the KIX disk is booted, the KIX utilities will automatically be copied into RAMdisk. You can terminate the copy process at any time by pressing the <ESCAPE> key.

5. When you call a KIX command, the KIX interpreter automatically searches for the utility in up to three locations: first, it looks for the utility in the RAMdisk; if not found, it looks for the utility in the current working directory (i.e., the directory specified by the system prefix); if still not found, KIX looks in the BIN directory of the KIX disk. If still not found, the KIX interpreter will return a "File Not Found" error.

6. When KIX is booted, you will be prompted to enter the current date. If you make an error while entering the date, you can press <ESCAPE> to clear the entry and start over.

IV. DIRECTORY COMMANDS

Directory Management Commands

Directory management commands consists of five utilities used to create and manipulate directories. The command syntax and descriptions are:

Command Syntax	Description
CD_*pathname*	Change Working Directory
LS_*pathname*	List Directories and Files
MKDIR_*pathname*	Make a New Director
PWD	Print Working Directory
RMDIR_*pathname*	Delete a Directory

(Note: An underscore character (_) in the command syntax denotes a blank space).

CD: Change Working Directory

Syntax: CD_*pathname*

Options: none

Description: The Change Directory command lets you change the current prefix or working directory. It is very helpful in reducing the amount of typing required to enter file pathnames. When you are calling a file which is located in the current working directory, you do not need to enter a full pathname; simply enter the name of the file.

For example, typing

% CD_/MYDISK sets the system prefix to /MYDISK.

% CD_/ sets the system prefix to the Null location. All files must be specified with complete pathnames, including the volume name. In multiple drive systems, this allows you to move from one disk to the next.

% CD sets the system prefix to the 'home' volume of the present working directory. If the working directory is /KIX/XTRA/SubD, typing CD sets the working directory to /KIX. In other words, CD by itself returns the name of the volume the working directory is on.

As usual, when a CD command cannot locate the indicated volume, the INSERT DISK message is displayed. The system then waits until you insert the appropriate disk and press <RETURN> or press <ESC> to abort the command.

LS: List Directories or Files

Syntax: LS_pathname

Options:	-L	List Extended Directory.
	-P	List File Protection Status
	-F	List File Types
	-N	Do Not Alphabetize List of Filenames
	-R	List All Files in All Subdirectories
	;*type*	List only files of this *type*
	>X	Redirect output to slot or pathname "X"

Description: The LS command lists information about the directories, subdirectories, or files contained in the pathname you indicate. The LS command has many options which allow you to control the files selected and the information displayed.

LS has two basic forms: the long (or extended) list, which displays everything the system knows about the directory or file; and, the short list, which displays only the name and type of each file contained in the directory.

When you save a file, KIX records the following information about that file.

Length	(in blocks)
File Type	(Text, Binary, Directory, System, or AppleWorks -- WP, DB, or SP)
Date/Time	(created or modified)
File Length	(in bytes)
Protection Status	(Read, Write, Rename, or Delete allowed)

LS Option List

The LS command has a number of options.

-L List using the 'long' format. This includes all of the file information listed above except for the protection status of the files.

-P List the protection status of each file. To use this option, you must also select the -L (long format) option. The KIX command CHMOD enables you to protect files from reading, writing, renaming, and/or deleting. The symbols used in the display are:

±R	Read protection is set
±W	Write protection is set
±N	Rename protection is set
±D	Deletion protection is set

If the sign preceding the letter is minus (-), then the protection status is active; if plus (+), protection is inactive. For example, if the listing displays **-W-D** next to a file, you cannot write or delete the file but you can read or rename it.

-F Prints the character code abbreviation for the file type. The abbreviations are:

&	Text file (txt)
*	Binary file (bin)
/	Directory (dir)
@	System File (sys)
	AppleWorks Files
!	Word Processing (AWP)
$	Spread Sheet (ASP)
#	Data Base (ADB)

-N The LS command, by default, lists all files in alphabetical order. If you specify the -N option, LS will list files in the order in which they are stored in the directory. This option works for both short and long listing modes.

-R Lists all files under the pathname including those contained in subdirectories. If this option is not selected, LS lists only the name of the subdirectory and not the files contained in it.

File Type Option (*;type*)

You can be even more selective about the files you want listed by limiting the list to specific file types. The LS command accepts a *;type* option which indicates the type of files you want listed. For example, if you only want information about *text* files stored in DIR1 on disk VOL1, type the command:

```
LS_-L_-P_/VOL1/DIR1_;txt
```

The file types are indicated by the codes:

;txt	List only text files
;bin	List only binary files
;dir	List only directories
;sys	List only system files
;awp	List only AppleWorks Word Processing files
;asp	List only AppleWorks Spread Sheet files
;adb	List only Appleworks Data Base files

Redirecting Output (>X)

Although you usually direct the output of a LS command to the screen, you may sometimes want to print a hard copy or save the list in a disk file. The LS command has a redirection option which directs the output of the command to another device -- either a printer or disk drive.

To redirect output, conclude the LS statement with a ">" followed by the slot number of the printer or the pathname of the file which will store the output. If no destination is specified, output will be directed to the screen.

Note: When the screen becomes full, output stops and the message [More] appears at the bottom of the screen.

For example, the command %_LS_-L_-P_/VOL1_;bin_>1 prints a long listing of all binary files contained on Volume 1. This listing includes all protection status conditions of the binary files. It directs this information to the printer in slot 1.

Short Listings

If you omit the -L option, the LS command defaults to the 'short' list format. A short list simply includes the filenames and their respective types. The -F option which prints the abbreviation for the file type can be used with the short list command. However, the -P option, which lists the protection status of each file, cannot be used with this option.

Listing Volumes On-Line

LS uses a special output format when you specify the 'root' volume, i.e. the Null position in the file hierarchy.

```
LS /
```

prints the slot and drive location, name, and number of blocks free for each volume that is currently on the system. Finally, it prints the total number of blocks available on all volumes and the number of volumes currently 'mounted' on the system.

Examples

The output produced by the LS command depends upon the options you select and the pathname -- or lack of pathname -- you indicate. The normal rules for pathnames apply.

If you enter:	The following is displayed:
LS_-L_/	a long list of all volumes available to the system
LS_-L_/VOL1	a long list of all files contained in the volume directory VOL1
LS_-L_/VOL1/DIR1	a long list of all files contained in DIR1 on VOL1.

If LS cannot locate the specified volume, the "INSERT DISK" message is displayed. The system then waits for you to insert the correct disk and press <RETURN>, or press <ESC> to abort the process.

MKDIR: Make a New Directory

Syntax: MKDIR_*pathname*

Options: None

Description: The MKDIR command creates a directory with the name you specify. Three rules apply to making directories.

1. You cannot create a directory with the name of an existing directory.

2. If the directory you are creating is a subdirectory of another directory, you must use a full pathname.

3. You can create any number of directories with a single MKDIR command. Simply separate the pathname of each new directory with a space.

MKDIR cannot be used to name a Volume or Root directory. To create a volume, see the KIX command FORMAT or CPV.

The following examples illustrate the use of the MKDIR command. If the working directory is /MYDISK,

MKDIR_XTRA	creates the directory XTRA on MYDISK
MKDIR_XTRA/MORE	creates a subdirectory, MORE, in the directory XTRA on MYDISK
MKDIR_/NEWVOL/D1	tries to create a directory, D1, on disk NEWVOL.

The initial "/" indicates a new volume since the first "/" means the Null position in the file hierarchy. If MKDIR can not find the volume for the directory, it displays the message

```
INSERT DISK FOR /NEWVOL/D1
PRESS RETURN TO CONTINUE; ESC TO SKIP
```

If you press <RETURN>, MKDIR tries to make the directory again. If it still does not find NEWVOL, it repeats the INSERT DISK message.

If you press <ESC>, MKDIR skips the indicated directory. If you are creating a number of directories, it goes to the next one in the list.

MKDIR will create as many directories as you indicate. Each directory, however, must be separated by a space. MKDIR tries to create the directories in the order they are listed. All of the directories in the above examples could have been created with one MKDIR command:

```
MKDIR_XTRA_XTRA/MORE_/NEWVOL/D1
```

MKDIR would first create the directory, XTRA. Then it would create the subdirectory of XTRA, MORE. Finally, it would search the entire system for the volume, NEWVOL -- note the initial "/" that indicates the Null location. If it finds the disk, it creates the directory, D1. If it does not find the disk, it prints the INSERT DISK message.

Once you have created a directory, ycu can store related files within that directory.

PWD: Print Working Directory

Syntax: PWD

Options: None

Description: The PWD, Print Working Directory, command prints the name of the current default directory on the screen. It is an extremely useful command if you get lost and can't remember where you are in the file system's structure.

If you issue this command when you first boot the KIX disk, it will print:

```
/KIX
```

If you are working on a file and can't remember which directory you have saved it in, save the file, get the % prompt, and issue the PWD command. The pathname to the current working directory will be displayed.

Once you are in a directory, any unprefixed filenames are assumed to reside within that directory.

RMDIR: Remove a Directory

Syntax: RMDIR_*pathname*

Options: None.

Description: RMDIR destroys the directory or directories listed. Before a directory can be removed, however, it must be empty of all subdirectories and all files. If the directory is not empty, RMDIR prints a "FILE ACCESS ERROR" message. If any other problems arise, (e.g., the correct volume cannot be located), the INSERT DISK message is displayed.

You can delete any number of directories with a single RMDIR command. Simply separate the pathname of each directory to be removed by a blank space.

Note: To delete all the files and subdirectories in a directory, see the RM command (Section V).

V. FILE MANAGEMENT COMMANDS

File Management Commands

File management commands consists of six utilities used to copy, rename or otherwise manipulate files. The command syntax and descriptions are:

Command Syntax	Description
CAT_*pathname*	List the Contents of a File
CHMOD_*pathname*	Change Protection Status
CP_*source* _*destination*	Copy a File
LPR_*pathname*	Print a File
MV_*old.name* _*new.name*	Move or Rename a File
RM_*pathname*	Delete a File or Directory

CAT: Concatenate

Syntax: CAT_*pathname*

Options:

-N Number each line printed, starting at line 1
-B Do not number blank lines
-S Remove adjacent blank lines
-V Print control characters as ASCII equivalents (for example, print ASCII 3 as [^c])
-F Print name of file at top of first page
-P Stop printer output after each page and prompt for a new sheet of paper.

Description: The CAT command performs a number of functions. It lets you print the contents of a file, concatenate multiple files in a single file, or print files using different format options.

For example,

```
CAT_-N_-B_-S_/Vol1/Dir1/Myfile
```

prints the contents of the file named Myfile on the screen. It numbers the lines that contain data and removes blank lines.

CAT prints text files as text. It prints AppleWorks word processing (AWP) files as they are displayed in the AppleWorks Word Processor. If the -V option is active, the AppleWorks formatting options will be displayed; printer options (e.g., ^ Underline Begin) will be displayed as carats.

CAT prints all files other than text or AWP files as 'hex' dumps.

CAT also supports concatenation of AWP files. Concatenating several files into one results in exact duplication of the files with the following exceptions:

1. The resulting file will be an ASCII text file (not an AWP file) and will not be paginated (however, if the file is converted back to an AWP file, AppleWorks will automatically repaginate it);

2. The AppleWorks "Zoom" function cannot be used to display printer format options in the resulting file.

3. The tab stops of the resulting file will be those of the first file in the list of files concatenated.

Normally, CAT prints to the screen. It will print 22 lines and prompt: **[more]** . To continue printing to the screen, press <RETURN>; to cancel printing, press <ESCAPE>.

Redirecting Output to a Printer

As an alternative to screen output, CAT supports the redirection of output to a printer. Placing a "**>1**" symbol at the end of the CAT command line will redirect output to the printer in slot 1. For example,

```
    CAT_Myfile_>1
```

instructs the computer to print the file named Myfile on the device (e.g., printer) in slot 1.

All CAT options can be used when redirecting output.

When printing hard copy, CAT uses an 80-column, 60 line-per-page format and follows the configuration settings you defined in the KIX configuration pro3ram (CFG).

Redirecting Output to a File

With the CAT command you can also redirect output to a file. Placing a ">*pathname*" instruction at the end of the command line will redirect output to the file specified by the pathname.

All files listed in the CAT argument are copied into the file indicated by the pathname. Each file is appended to the preceding file. The first source file determines the type and protection status of the resultant file. If one of the files being copied is not the same type as the first file, CAT will close the destination file and terminate the command.

The CAT command can be used to merge a number of small files into a larger unit. The following example merges three chapter files into one large file named Section which is contained in a directory named Book.

```
CAT_Chapt1_Chapt2_Chapt_>/Vol1/Book/Section
```

If CAT cannot find the files or volumes indicated, it displays the INSERT DISK message. Press <ESC> to skip the current file and continue. Otherwise, place the appropriate disk in the drive and press <RETURN> to continue.

Creation of Text File from the Keyboard

CAT allows you to create text files directory from the keyboard and/or type directly to the printer using the redirection option.

To print directly from the keyboard, use the following command syntax:

```
% CAT_>1
```

Following this command, lines of text entered through the keyboard will be directed to the printer in Slot 1. The maximum line length is 127 characters. Paging is tracked automatically and a new page command is issued after each 60 lines. To terminate direct output to the printer, you must enter <CNTL>-Z on a new line and press <RETURN>.

To create and save a text file directly from the keyboard, use the following command syntax:

```
% CAT_>DESTFILE
```

Following this command, each line of text entered through the keyboard will be saved in a Text file called "DESTFILE" when the <RETURN> key is pressed.

To close the file, enter <CNTL>-Z on a new line and press <RETURN>.

CHMOD: Change Protection Mode

Syntax: CHMOD_*pathname*

Options:

±R	Read Protection
±W	Write Protection
±D	Destruction Protection
±N	Rename Protection

Description: KIX gives every file 4 modes of protection. These protection modes determine which files can be Read, Written to, Deleted, or reNamed. The CHMOD (change protection mode) command is used to set these protection modes.

To change the protection status of a file, you must invoke the CHMOD command and indicate whether you want to activate or deactivate a protection mode. A + (plus) sign before the mode character deactivates the protection. A - (minus) sign before the mode character activates the protection.

Though ProDOS does not support Passwords and the locking of files, CHMOD gives you the next best thing. By activating read protection (-R) on a file, the casual user is unable to read that file. The only way someone can read the file is by invoking the CHMOD command and deactivating the read protection.

The following examples demonstrate use of the CHMOD command.

% CHMOD +R -W -D +N File1
allows a user to read (+R) or rename (+N) the file, File1. It prohibits the user from writing to (-W) or destroying (-D) the file.

% CHMOD +R +W -N File1
lets the user read (+R) or write to (+W) the file, but prohibits him/her from renaming (-N) it. The status of deletion protection (D) is unchanged.

CP: Copy

Syntax: CP_*Source.Pathname*_*Destination.Pathname*

Options:

-I Prompt user for permission to replace a file if one by the same name already exists in the destination directory.

-Q Prompt user for permission to copy a file.

; *type* Copy only files of the type specified.

Description: The Copy command produces a replica of the source file in the destination file.

If the *Destination.Pathname* argument is a directory, CP writes the source file into the directory with the same filename. If the *Destination.Pathname* argument is a filename, CP copies the file into the current working directory with the filename indicated.

CP permits multiple source files to be copied into a destination directory.

If the last pathname specified (i.e., the destination pathname) is not a directory, CP automatically copies all of the files listed into the working directory.

CP will not copy subdirectories. If a subdirectory is listed as a source file in the CP argument, it will be ignored. No error message will be generated.

-I Option

The "-I" option causes CP to prompt the user for permission to write over a file with the same name if one already exists in the destination directory. Responding NO to the KIX query causes CP to abort the copy. This option is intended to prevent you from accidentally destroying a file that you want to keep.

-Q Option

The "-Q" or Query option causes CP to prompt the user for permission to copy a source file. Responding NO to the query causes CP to move to the next file matching the criteria specified on the command line. This option is useful when a wildcard has been used to specify a list of files to be copied.

;*Type* Option

The ";*type*" option causes CP to copy only files of the file-type specified. This option is useful when a wildcard has been used to specify a directory or other grouping of source files.

Examples

The following examples illustrate use of the Copy command and options.

%_CP_File1_/Vol1/Dir
copies File1 (located in the current working directory) into the directory Dir located on volume /Vol1. The new file retains the name File1.

%_CP_-I_File1_Vol1/Dir1/SFile
copies File1 (located in the current working directory) into the directory Dir1 on volume /Vol1. The new file is named SFile. If the file SFile already exists in directory Dir1, the "-I" option will generate the following message:

```
REPLACE EXISTING "DIR1/SFILE"   (Y, N)
```

If you respond with Y, the existing file is overwritten. If you press N, the command is aborted.

%_CP_/Source.Vol/*_;BIN_/Destination.Vol_-Q_-I
copies all files of type BIN from the /Source.Vol directory to the /Destination.Vol directory, prompting the user for permission to copy each one. The following output could be expected on the screen:

```
Copy "/SOURCE.VOL/FILE1"? (Y/N)
Copy "/SOURCE.VOL/FILE2"? (Y/N)
Replace existing /DESTINATION.VOL/FILE2?  (Y/N)
```

File1, File2 and any other files subsequently listed would only be of type BINary. Responding NO to any of the queries causes CP to move to the next file matching the criteria specified in the command line.

LPR: Line Print

Syntax: LPR_*Pathname*

Options:

-F Print the name of the file at the top of the first page of output.

-P Stop printing at the end of each page so that a new piece of paper can be inserted.

Description: The LPR (line print) command prints the files specified in the Pathname argument. The files are printed using the margins, special printer codes, and other formatting options selected in the system-configuration program. See the CFG command in Section X for information about resetting the default printer settings.

LPR prints AppleWorks Word Processing files and text files as text; all other types of files are ignored. You can stop the printer output by pressing the <ESC> key. If LPR cannot find the file you specify, it will display the INSERT DISK message.

MV: Move

Syntax: MV_*Old.Name*_*New.Name*

Options:	-I	Prompt for new name of each file listed.
	-F	Ignore the protection status of files.
	-Q	Prompt user for permission to move file.
	; *type*	Move only files of type specified.

Description: The Move command allows you to change the pathname of a file. This has the effect of moving the file or renaming it. When a file is moved or renamed, the original file is deleted.

When moving files using the -I option, MV will prompt you for a new pathname for each file listed. If you enter <RETURN> without a pathname, MV will not move that file and will skip to the next file.

If the destination of the MV command is a directory, the files are automatically moved into the directory with their original filenames. The -I option cannot be used.

If you include more than two files in the *Old.Name* argument, the *New.Name* argument must be a directory. The reason is that, without including the -I option, the system has no way to identify file names. If they are copied into a directory, they retain their original filenames.

If *New.Name* is not a directory, MV will move the files into the working directory.

The -Q option is redundant when the -I option is active, and so KIX ignores it.

RM: Remove

Syntax: RM_*Pathname*

Options:

-I	Inquire before destruction of each file.
-F	Destroy file regardless of protection status.
-R	Empty the directory before deleting it.
; *type*	Remove only files of type specified.

Description: The Remove command deletes files and directories from a volume.

The -I option causes RM to prompt you before destruction of each file. It is intended to help you avoid accidental deletion of files. If you decide not to delete a file, type N and RM will skip to the next file in the list.

If you specify the -I option when deleting a directory, you will only be prompted regarding destruction of the first directory. If this directory contains subdirectories, they will be destroyed without warning.

The -F option instructs RM to ignore the protection status of each file. If the -F option is used when deleting a directory, the RM command ignores the protection status of all files in that directory and in any subdirectories.

Normally, you cannot delete a directory unless it is already empty. However, RM lets you delete non-empty directories by using the -R option. It automatically deletes all files in the source directory (including subdirectories and their files) and then the directory itself. If you are deleting non-directory files, the -R option has no effect.

The RM command also supports type selection. This feature provides an easy way to selectively delete files. When the -R option is active, the type selection field is ignored.

The following example illustrates use of RM. The command deletes all files and subdirectories contained in the directory TESTDIR.

```
% RM_-I_-F_-R_TESTDIR
```

Before the command is executed, the screen displays the message:

```
WARNING: TESTDIR WILL BE EMPTIED
REMOVE TESTDIR?  (Y/N)
```

If you select **Y**, RM deletes all the files in each subdirectory, then those in each directory, moving up through the file hierarchy until it deletes TESTDIR itself.

VI. VOLUME COMMANDS

Volume Management Commands

Volume management commands consists of three utilities used to copy, format, and rename volumes. The command syntax and descriptions are:

Command Syntax	Description
CPV_(slot,drive)_(slot,drive)	Copy a Volume
FORMAT_(slot,drive)_/*name*	Format a Volume
MVV_(slot,drive)_/*new.name*	Rename a Volume

CPV: Copy a Volume

Syntax: CPV_(ss,sd)_(ds,dd)

Arguments:

ss	source slot of original disk
sd	source drive of original disk
ds	destination slot of new volume
dd	destination drive of new volume

Description: The copy volume command duplicates the source volume in the destination volume.

CPV identifies the source disk and asks if the new volume should have the same volume name. If Yes, then CPV initiates the copy sequence. If No, CPV requests a new volume name. Press <RETURN> to terminate the CPV command.

If the CPV command determines that the destination disk is already named, it will ask you to verify the destruction of the existing destination volume. A Yes response causes CPV to format the disk and copy the volume. A No response terminates the CPV command.

FORMAT: Format A Blank Disk

Syntax: FORMAT_(s,d)_/NewVolumeName

Arguments:

s	Slot location of disk
d	Drive location

Description: The Format command lets you format a disk of any size.

To make formatting easier on single-drive systems, FORMAT tells you to insert the blank disk before formatting actually begins. Insert the disk and press <RETURN>; or press <ESC> to stop the process.

Before formatting begins, a check is made to determine if the disk in slot "s", drive "d" is already formatted. If it is, you are asked to verify its destruction. If you respond Yes, the disk will be reformatted and given the new volume name you have specified.

If you are formatting a "large" volume (i.e., a volume larger than 280 blocks), you will be prompted to confirm the command. This extra prompt is included to help you avoid accidentally reformatting a hard disk drive or a UniDisk 3.5.

MVV: Rename a Volume

Syntax: MVV_(s,d)_*/NewVolumeName*

Arguments:

s	slot
d	drive

Description: This command allows you to rename a volume. When the MVV command is entered, the following prompts appear:

```
Insert Volume to be renamed.

Type <RETURN> to proceed;  <ESC> to exit

Rename volume "/OldVolumeName"?  (Y/N)
```

All volumes, including RAMdisks, can be renamed using the MVV command.

VII. SPECIAL KIX COMMANDS

Special KIX Commands

Special KIX commands consist of five utilities used to compare files and volumes and to find strings. The command syntax and descriptions are:

Command Syntax	Description
CMP_*file.1_file.2*	Compare Two Files
CMP_(ss,sd)_(ds,dd)	Compare Two Volumes
FIND_*directory_-filename*	Locate a File in a Directory
GREP_*string_pathname*	Locate a String in a File(s)
SDIFF_*file.1_file.2*	Compare Two Text Files

CMP: Compare

Syntax: CMP_*File1*_*File2*
CMP_(ss,sd)_(ds,dd)

Options: None

Description: The CMP command is used to compare two files (of any type) or two volumes.

When CMP is used to compare two files, it searches until it finds the first difference between two files. Then, it reports the location of the difference as an offset from the start of the files in Block 1, Byte1. No further comparison is made after the first difference is discovered.

The following example shows the input and output of CMP when used to compare two files.

```
% CMP_File1_File2
FILES DO NOT MATCH: BLOCK 2, BYTE 457
```

CMP compares the two files and finds a difference in the second block of the files. It identifies the difference in the 457th byte. If CMP determines that the files are identical, the message **FILES MATCH** appears on the screen.

CMP also supports the comparison of volumes. It works in the same manner as a file comparison, comparing the two volumes on a byte by byte basis until a difference is located.

The syntax of the CMP command is different when comparing volumes. The command syntax must include the slot and drive identification of the volumes being compared.

```
CMP_(ss,sd)_(ds,dd)
```

ss	source slot of original disk
sd	source drive of original disk
ds	destination slot of new volume
dd	destination drive of new volume

FIND: Locate File

Syntax: Find_*Pathname*_*-Filename*

Options: None

Description: The Find command searches through the volume or directory indicated in the Pathname for the file defined in Filename. The command returns the full pathname of all files that match the filename to be located. Note the "-" (minus) sign preceding the filename; it is an essential part of the command syntax.

The pathname determines the extent of the search. For example,

```
%_FIND_/_-File1
```

searches the entire system, including all volumes currently mounted on the system, for File1. It prints the full pathname to every file named File1. The "/" indicates that the search should begin with the root or null location in the system and include all volumes.

```
%_FIND_/VOL1_-FILE1
```

searches the volume, VOL1, for any directories that contain File1. It prints the full pathname to the file whenever it locates a file with that name.

In another example, assume that two directories -- /AnyVol/Dir1 and /AnyVol/Dir2/SubD1 -- contain a file named File1. Then,

```
% FIND_/AnyVol_-File1
```

returns the following lines of output:

```
        /AnyVol/Dir1/File1
        /AnyVol/Dir2/SubD1/File1
```

If FIND cannot locate the specified volume name, it displays the INSERT DISK message. Press <ESC> to abort the command; or insert the appropriate disk and press <RETURN>.

GREP: Locate Strings in Text Files

Syntax: GREP_*String*_*Pathnames*

Options: -N Show line numbers

Description: The GREP command searches for a string of characters in the text files designated by Pathname. When GREP locates the string, it prints the Filename and the text of the line that contains the string. The search continues until all the files and/or directories designated by Pathname have been examined.

If the string being sought contains spaces, the string should be enclosed in single quotes (').

GREP uses the [more] convention described previously in those cases where the screen fills with text. Press <RETURN> to continue. Press <ESC> to exit the listing and return to the system prompt,.

If the "-N" option is selected, GREP will print the relative number of the line in the text file where the string has been found.

For example, the following command searches two files, MasterFile and TestFile for the string, 'Copyright (c)'.

```
% GREP_'Copyright_(c)'_MasterFile_TestFile

MASTERFILE: Copyright (c) 1986 KYAN SOFTWARE

MASTERFILE: COPYRIGHT (C) 1983 APPLE

TESTFILE:  Sample Program Copyright (c) 1985
```

Note that the search is not restricted by case. Any upper/lower case version of the string will be reported.

SDIFF: Source Difference

Syntax: SDiff_*File1*_*File2*

Options: -N Show Line Numbers

Description: Source Difference (SDiff) is a variation of the CMP command. It is used to compare text files and prints the text of the lines which contain differences. The comparison terminates after 3 differences are located.

If the files specified are not text files, SDIFF automatically terminates.

If the "-N" option is selected, the number of the line containing the difference is printed along with the text.

VIII. ABBREVIATIONS AND WILD CARDS

Abbreviations and Wildcards

Abbreviations and wildcards greatly enhance the power and ease of use of KIX commands. Directory abbreviations are used to reduce the amount of typing required to enter KIX commands. Wildcards are used to replace elements of filenames. The following are used in KIX.

Command Syntax	Description
.	Abbreviation for Working Directory
..	Abbreviation for Parent Directory
?	Wildcard for Single Character
*	Wildcard for Character String
ECHO_*pathname*	List all pathnames affected by wildcard usage.

Directory Abbreviations

KIX supports two directory abbreviations. These serve as substitutes for the current working directory and the parent directory.

Symbol	Abbreviation for
. (1 period)	Working Directory
.. (2 periods)	Parent Directory

Note that each dot or period represents one position above the current file in the pathname hierarchy.

The "." (1 period) abbreviation is used to abbreviate the working directory. In the following examples, assume that the working directory is "/Vol1/Dir1":

Command	Action Taken
CP_/Vol2/File1_. ^	copies File1 on Vol2 to the current working directory. The pathname of the new file is/Vol1/Dir1/File1.
MV_./File1_/Vol2 ^	moves File1 in the current working directory to the volume Vol2. The pathname of the new file is /Vol2/File1. (Note: The original file, /Vol1/Dir1/File1, is destroyed).

The ".." (2 periods) abbreviation represents the parent directory of the working directory (i.e., the working directory is a subdirectory of the parent directory). In effect, it moves the system one step up the pathname hierarchy from the working directory or two steps up from the current file. If the working directory pathname is /Vol1/Dir1/MyDir, then ".." is an abbreviation for /Vol1/Dir1.

You can use the ".." (2 period) abbreviation wherever a directory pathname is used. For example, assume that the working directory is /Vol1/Dir1/MyDir:

```
CD_../LastDir    changes the working directory from
  ^              /Vol1/Dir1/MyDir  to  /Vol/Dir1/LastDir.
```

If the working directory is the Volume or Root directory, the ".." abbreviation sets the system to the Null volume.

Wildcards

Wildcards are used to replace strings and characters in filenames. KIX supports two wildcards.

Symbol	Used to Represent
?	Any character in a filename
*	Any string in a filename

These two wildcards give you a great deal of power when you issue KIX commands.

? (Question Mark)

The character wildcard, "?", is used to represent any single character. An example illustrates the use of wildcards more clearly than an explanation. The command

```
LS_Letter?
```

produces a list of all files in the present working directory that have a filename beginning with the character string "Letter" and followed by another character. If the directory contains Letters, Letter1, Letter2,...Letter9, the command

```
LS_Letter?
```

lists all of these files. If you want to copy all of these files into another directory, a wildcard lets you copy them with a single command.

```
CP_/NewDirectory/Letter?
```

copies the files from NewDirectory into the present working directory . The names of the files remain the same.

In another use of the wildcard, ?, the statement

```
CAT_?AT
```

lists all the files in the present working directory that begin with a single character followed by the letters "AT" (e.g., FAT, HAT).

* (Asterisk)

The asterisk wildcard, *, is a more powerful character substitute. It can be used to represent any string of characters, including a null or empty string. Using this wildcard, you can quickly search through volumes and directories for files. For example,

```
CAT_M*.S
```

prints the contents of all files in the present working directory with a filename beginning with the letter "M" and ending with the suffix ".S".

As another example, the command

```
RM_/AnyVol/AnyDIR/M*.S
```

deletes any file located on the volume, AnyVol, in the directory, ANYDIR, whose filename begins with an "M" and ends with a ".S". If the following files existed on the disk, they would be removed.

```
/AnyVol/AnyDir/MouseFile.S
/AnyVol/AnyDir/ModelTest.S
/AnyVol/AnyDir/Memo.S
```

Echo_*pathname*

KIX provides a command which is intended to let you see exactly which pathnames and files are affected by the wildcard usage. With the ECHO command you can review the list of "echoed" pathnames before any commands are executed. ECHO is used to avoid accidents caused by wildcard usage.

ECHO lets you specify a pathname which contains a wildcard character. ECHO then prints the complete pathnames of all files which match the wildcard specification.

In a previous example, the ECHO command could be used to list the pathnames of all the files which would be deleted by the command:

```
RM_/AnyVol/AnyDIR/M*.S
```

By first entering:

```
ECHO_/AnyVol/AnyDIR/M*.S
```

you would see the following list of files printed on the screen.

```
/AnyVol/AnyDir/MouseFile.S
/AnyVol/AnyDir/ModelTest.S
/AnyVol/AnyDir/Memo.S
```

If files were listed which you did not want to delete, then you would know not to use the wildcard in this particular case.

The ECHO command is intended to help you get the most out of wildcards while minimizing the chances of an accident. If you practice the use of wildcards using this command,you will quickly become familiar with the use and power of KIX wildcards.

Using Wildcards

The use of wildcards is fairly logical. You can use them wherever and whenever the system knows what the wildcard means.

They can be used in source pathnames since the system has the information needed to determine what the wildcard stands for. However, wildcards cannot be used in destination pathnames since the system would not know what character or string to substitute for the wildcard.

Wildcards are also not valid in the source pathname of some KIX commands. For example, the Make Directory command (MKDIR) cannot use wildcards in the directory name since the system would not know what the name should actually be.

In general, wildcards can be used in the source pathname of any KIX command which accepts a list of pathnames as valid source file identifiers. A list of these commands follows.

Wildcard Usage with KIX Commands

Command	Wildcard Usage
RMDIR	Allowed
LS	Allowed
LPR	Allowed
CAT	Allowed
ECHO	Allowed
CP	See Note 1
MV	See Notes 1 and 2
RM	Allowed
CHMOD	Allowed
FIND	Allowed
GREP	See Note 3

Note 1: Wildcards are valid in the source pathname if the destination pathname is a directory.

Note 2: Wildcards are valid in the source pathname when the -I option is also selected.

Note 3: Wildcards are not permitted in the search pattern string. However, they are allowed in the list of filenames to be searched.

IX. FIRMWARE CALLS

The KIX firmware calls consist of four utilities used to enable and disable the 80 column firmware, set the system time and date, and dump the contents of the screen to the printer. The command syntax and descriptions are:

Command Syntax	Description
C40	Disable 80 Column Firmware
C80	Enable 80 Column Firmware
Date_*yymmddhhmm*	Set or Read the System Date and Time
SD	Print Screen Contents

C40: Disable 80 Column Display

Syntax: C40

Options: None

Description: This command disables the 80 column card, clears the screen, and returns the system to 40 column mode.

C80: Enable 80 Column Display

Syntax: C80

Options: None

Description: This command clears the screen, sets video to normal, and enables the 80 column display. If your computer does not have an 80-column card, the command is ignored.

C80 is compatible with most 80 column cards including Apple Computer, Applied Engineering, Videx, and Checkmate.

DATE: Set/Read System Time/Date

Syntax: Date_*yymmddhhmm*

Arguments: year-month-day-hour-minutes

yy	year	(85 through 99)
mm	month	(01 through 12)
dd	day	(01 through 28-31)
hh	hour	(00 through 23)
mm	minute	(00 through 59)

Description: The DATE command lets you read or set the ProDOS system time and date files -- even if you do not have a clock/calendar card.

To display the current date and time, enter DATE and press <RETURN>. For example,

```
%DATE
17-JAN-86 07:56
```

To set the date and time, enter **DATE** followed by an argument containing the year, month, day, hour, and minute in two-digit values. No spaces are allowed between values in the argument. However, you may include non-space or non-numeric characters to make the entry easier to read. The following examples set the same date-time values:

```
% DATE 8603241500

% DATE 86/03/24/15-00
```

Both commands set the calendar/clock to

```
24-MAR-86 15:00
```

If you set the day, DD, value to zero, the DATE command returns **NO DATE.** The system time, however, is always displayed.

The DATE command is compatible with the ThunderClock and other clock/calendar peripheral cards. If your system has one of these cards installed, the DATE command allows you to set it with the proper date and time. With these cards the DATE command also supports a DAY of the week field.

Using DATE to set the system clock also sets the clock card (assuming the card is set-enabled). You may also set the day-of-the-week feature on the card by including the 3-letter abbreviation for the day of the week after you have described the set-date command line. For example,

```
% DATE 8603241500MON
```

sets the system calendar/clock to

```
MON 24-MAR-96 15:00
```

When you set the DATE command, the system does not check to see if the current day-of-the-week is correct. If the system does not have a clock card, the day argument is simply ignored.

The day of month value is checked against the month specified in the DATE argument. If the specified month does not contain the number of days indicated, the value is not allowed. For example, 31-SEP is rejected. Also, 29-FEB is allowed only if the year is calculated to be a leap year.

SD: Screen Dump

Syntax: SD

Options: None

Description: The screen dump command lists the current screen display to the printer specified in the KIX configuration program (CFG). The printer output is initialized and conditioned by the CFG control codes and linefeed settings. If no printer is specified in the configuration files, the command is ignored.

X. KIX UTILITIES

KIX Utility Commands

KIX Utility commands consists of four utilities used to setup and configure KIX for the user's system. The command syntax and descriptions are:

Command Syntax	Description
CFG	KIX Configuration Program
INSTALL	KIX Installation Program
KIX	KIX Command Summary
QUIT	Exit KIX

CFG: Configure the KIX System

Syntax: CFG

Options: None

Description: The CFG (configure) command lets you customize KIX for use on your system. This command modifies the configuration of KIX which is installed on the system disk. To modify the default configuration values, type CFG at the system prompt (%) and the following Configuration Menu will appear.

```
Path: /KIX/BIN                         KIX Configuration Utility
-----------------------------------------------------------------
                          MAIN MENU

          1        Display/Change Startup Conditions

          2        Display/Change KIX Printer Settings

          3        Save this Version of KIX to disk.

          4        Quit

-----------------------------------------------------------------
Type Number or use arrows, then press <Return>     Press -? for HELP
```

For more information about the CFG utility, please refer to Section II of this manual ("Getting Started").

INSTALL

Syntax: Install

Options: None

Description: The Install program is used to install KIX as a desktop accessory on your AppleWorks boot disk. The program must be run on an Apple //e or //c. Please refer to Section II of this manual for more information about KIX/AppleWorks Desktop Utility.

Important Note: Be sure you install KIX on a backup copy of your AppleWorks boot disk; do not use your original disk.

KIX: KIX Command Menu

Syntax: KIX

Options: None

Description: The KIX command is a series of HELP screens which display the list of KIX commands and summarize the available options and required arguments.

The cursor control keys and space bar are used to scroll through the help screens. Pressing <RETURN> or the <Space Bar> will advance the text one full page.

To quit the KIX Help program, simply press <ESCAPE>.

QUIT: Exit KIX

Syntax: Quit

Options: None

Description: The QUIT command executes the ProDOS dispatcher code which was found on bank 2 of main memory when the system was first powered up. This will return you to the ProDOS prompt or to any start-up program which was used to launch KIX (e.g., Catalyst, Mouse Desktop).

Entering QUIT at the system prompt invokes the prompt:

```
EXIT KIX?  (Y/N).
```

Typing "N" voids the command and returns you to the system prompt. Typing "Y" calls the ProDOS Dispatcher Code.

XI. TECHNICAL REFERENCE

A. ProDOS Dispatcher (Quit) Code

When the "ProDOS" file is loaded, it places its "Interpreter Dispatcher Code" in bank 2 of bank switchable main memory from locations $d100 through $d3ff ($300 or 768 bytes). This "Dispatcher" code is executed whenever ProDOS executes a "Quit" command through the MLI.

The "Quit" starts by relocating the code in bank 2 from locations $d100 - $d3ff to locations $1000 - $12ff in main memory. A JMP to $1000 is then executed to invoke the dispatcher code.

When KIX is booted, the KIX.System file copies the KIX shell into the dispatcher area of memory. Thus each KIX command and Pascal program (run in the KIX environment) is able to return to the KIX shell by executing a "Quit" call to the ProDOS MLI.

This same technique is used by other application programs (such as Catalyst and MouseDesk) to regain control of the system from ProDOS when an application is complete. They also replace the Dispatcher Code with their own. Thus there is a natural conflict between KIX and these programs.

To avoid conflict with these types of programs, KIX takes the following additional steps to preserve the Dispatcher code of these applications.

The KIX.System file, when executed, copies the existing contents of bank 2 main memory from $d100 - $d3ff (the ProDOS Dispatcher area) into the KIX QUIT command (i.e., it saves the dispatcher code currently in the bank 2 memory in the

QUIT command). Only then does it install the KIX shell code onto the Dispatcher bank. Thus, when you are finished with KIX and want to exit, you simply execute the QUIT command and you are returned to the launching application or the ProDOS prompt.

This feature is not important to most users of KIX but the convenience it offers will be enjoyed by Catalyst users and the growing number of KIX "power" users.

When the KIX shell is invoked via a ProDOS "Quit" command, it is relocated to $1000. The shell then performs the following actions:

a. Enables main memory bank 1.
b. Closes all open files.
c. Marks the ProDOS Memory Bit Map used for pages 0 thru 7, and pages $bc, $bd, and $be.
d. Relocates the actual shell code to main memory $bc00 thru $be7f.
e. Executes the shell via a JMP $BC00.

B. Memory Usage

The KIX shell occupies main memory from $bc00 thru $beff when active. KIX also uses the following areas of memory:

Locations	Description
$200 - $27f	Keyboard input buffer
$280 - $2bf	Scratch space for pathname parsing by KIX Shell
$2F8 - $2FE	Printer card initialization codes (6 characters and a null)
$2FF	Boolean True (hi bit set) if a Linefeed must follow a carriage return.
$300 - $309	The ProDOS pathname of the BIN directory in which to search for KIX commands. For example, if KIX is booted on a volume named "/KIX", memory at $300 - $309 would appear as follows:

Address	Contains	Comment
$300	9	Length of prefix in use.
$301-$309	/KIX/BIN/	Prefix to use on commands.
$30a-$337	?	Undefined/Reserve

The memory from $340 through $3e0 is safe during execution of Kyan Pascal (Version 2.0) programs.

C. Apple IIe/IIc RAMdisk

On startup, ProDOS defines auxiliary memory in an Apple //c or 128K //e as a 127 block disk volume, named /RAM. KIX includes a feature which allows you to copy KIX files from the KIX BIN directory into a BIN directory on the /RAM volume. Taking advantage of this feature makes the access and response time of KIX commands much faster.

KIX determines if the system should attempt to access the /RAM volume by scanning the ProDOS device list (starting at $bf32). If a device is found in slot 3 (either drive), a test is made for /Ram.

D. Memory Expansion Cards

Many people use an extended memory card with their system. You to configure a portion of this card to be an oversized "RAM" volume (i.e., larger than the 64K supported by ProDOS) using software provided by the manufacturer. You can use this oversized RAM to store KIX commands. The advantage to doing so is that the execution time of commands is drastically reduced.

To set up an oversized RAM volume, you must run a "preboot" program. This program configures and initializes the /RAM volume and then passes control to the KIX.System file. The KIX.System file then creates a BIN directory in /RAM and copies the KIX files from the disk. KIX.System will continue to copy files into /RAM/BIN until there is no more space available, or until all of the files in the KIX/BIN directory have been loaded.

Side 1 of the KIX disk contains two files which are used in conjunction with the software provided by the expansion card manufacturer to create a preboot program.

o AE.PREBOOT is used with Applied Engineering expansions cards.

o CM.PREBOOT is used with CheckMate expansion cards.

The following instructions describe how to create a preboot program for your /RAM volume. These instructions assume that you have a copy of the /RAM software provided by the RAMcard manufacturer and a copy of a ProDOS Users Disk.

1. FORMAT a new disk and and name it "PREBOOT".

2. Copy ProDOS and BASIC.SYSTEM onto the disk from your ProDOS Users Disk.

3. Copy the /RAM configuration program onto the disk from the disk provided by the RAMcard manufacturer ("ProDrive" if you have an Applied Engineering card; "MultiDrive.Pro" if you have a CheckMate card).

4. Copy the appropriate preboot program (Applied Engineering or CheckMate) from the KIX disk to the PREBOOT disk. When you copy this file, change the name of it to "STARTUP".

5. Remove the PREBOOT disk from the drive and boot your copy of the ProDOS Users Disk. Select AppleSoft BASIC from the menu. When the BASIC prompt appears, remove the ProDOS disk and re-insert your PREBOOT disk.

6. Enter the command: **LOAD_STARTUP**

7. Enter the command: **LIST_10**

8. Line 10 of the STARTUP program will now appear:

   ```
   10_PFX$="/HARD1/KIX.SYSTEM"
   ```

   This line of the program must be changed to conform with your system configuration and the location of the KIX.System file. For example, if you have KIX stored on a floppy disk, change the pathname "/HARD1/KIX.SYSTEM" to the pathname used for the floppy (e.g., "/KIX/KIX.SYSTEM"). If you have KIX located on a Profile hard disk, change it to "/PROFILE/KIX.SYSTEM".

9. When the pathname has been changed, press <RETURN> and save the change by entering the command: **SAVE_STARTUP**

That's it! Now, when you boot the PREBOOT disk, a large RAM volume will automatically be set-up and the KIX commands copied into it. When the KIX prompt appears, you can load and run any application program (including AppleWorks). The KIX commands will remain in RAM until you turn off the computer.

E. Notes on KIX and AppleWorks

After you install KIX in the AplWorks.System file, the KIX copyright notice will briefly appear before the usual AppleWorks startup screens.

When you press -K, KIX will preserve the desktop memory by saving some in the Aux Memory Language Card area. The remainder (about 42,500 bytes) will be written to a disk file ("KIX.Desktop"). If you have setup a RAM volume for KIX (see Section D), KIX will try to save this intermediate file on the RAM volume. If the system has no RAM volume, or there isn't enough room for the file, KIX will write KIX.Desktop in the volume containing KIX's BIN directory.

Note: On a floppy disk, saving the intermediate file, "KIX.Desktop", will take about 25 seconds; on a hard disk, about 7 seconds; and, in RAM, about 4 seconds.

When the KIX window appears in AppleWorks, you can execute any program listed in the KIX BIN directory. If you have a BIN directory in RAM, you can also run any program contained in this directory (/RAM/BIN).

Note: KIX will not recognize programs stored in sub-directories in the BIN directory nor will it run System files (i.e., all files must be binary).

It is possible to edit, compile, and run Kyan Pascal programs from the AppleWorks/KIX window. However, please remember that you can use any directory for your source files, but that the BIN directory (containing PC, AS, LIB, STDLIB.S, etc.) must be the destination for any work you want to compile/assemble and test-execute.

Final Note: Be careful not to remove or destroy the KIX.Desktop file. If you do, you may lose important data, and you will not be able to return to AppleWorks.

kyan

KIX™

"Adds the Power of UNIX to ProDOS and AppleWorks"

Kyan Software Inc.
San Francisco, Calif.

KIX™

Adds the Power of UNIX to ProDOS and AppleWorks!

KIX is a New Operating Environment for the Apple.

It is modeled after the UNIX™ operating system and provides you with tools found only on large, multi-user UNIX systems. KIX makes the Apple faster, easier and more efficient to use.

KIX Expands the Power of ProDOS™.

It lets you do things which were difficult or even impossible to do before with the ProDOS FILER. And, KIX is compatible with virtually all ProDOS-based software.

KIX Expands the Power of AppleWorks™.

Install KIX on your AppleWorks disk and you can jump to the KIX environment at the touch of a key. Now, you don't have to exit AppleWorks or work through cumbersome menus to perform simple file management functions. And, when you are finished, simply press Apple-K to return to AppleWorks.

KIX Gives You the Power to:

Control Directories: (print, create, delete or change directories);

Manipulate Files: (copy, move, delete, rename, print or change protection status of files);

Work with Volumes: (copy, delete, or rename volumes; format disks);

Compare Files and Volumes: (compare and list the differences between multiple files and/or volumes);

Search Files and Directories; (search directories for files and/or search files for character strings);

Make Hardware/Firmware Calls; (set system time/date and 40/80 column video outputs; and dump screen contents to a printer).

In addition, KIX contains full support for wildcards and redirection of output to printers and other peripheral devices.

UNIX is the operating system of the future. You can learn UNIX today and enjoy much of its power on your Apple II computer! Give it a try!

Hardware Requirements: Any Apple II with 64K of memory and two disk drives; RAMcard recommended.

KIX is a tradmark of Kyan Software. ProDOS and AppleWorks are trademarks of Apple Computer. UNIX is a trademark of Bell Laboratories.

kyan

Quick Guide to KIX™

Directories

Command Syntax	Description	Options and Arguments	
CD *pathname*	Change Working Directory	*pathname* of New Directory	
LS	List Directories and Files	-L	List Extended Directory
		-F	List File Types
		-P	List Protection Status
		-N	Do Not Alphabetize List
		-R	List All Files in Directory
		:*type*	List Only Files of this *type*
		>X	Direct Output to Slot or Path
MKDIR *pathname*	Create New Directory	*pathname* of New Directory	
PWD	Print the Working Directory	None	
RMDIR *pathname*	Delete a Directory	*pathname* of Directory	

Files

Command Syntax	Description	Options and Arguments	
CAT *pathname*	List Contents of a File	-N	Print Line Numbers
		-B	Skip Blank Lines
		-S	Remove Blank Lines
		-V	Print ASCII Values
		> N	Direct Listing to Slot N
		>*pathname*	Save Listing
CHMOD *pathname*	Change Protection Status	± R	Read Access
		± W	Write Access
		± D	Destruction Access
		± N	Rename Access
CP *source* *dest.*	Copy a File	-I	Prompt Before Replace
		-Q	Prompt Before Copy
LPR *pathname*	Print a File		None
MV *old.name* _*new.name*	Move or Rename a File	-I	Prompt for New Name
		-F	Ignore File Protection
		-Q	Prompt Before Transfer
RM *pathname*	Delete a File or Directory	-I	Prompt before Deletion
		-F	Ignore Protection Status
		-R	Empty and Delete Directory

KIX is a Trademark of Kyan Software, Inc.

Volumes

Command Syntax	Description	Options and Arguments	

CPV_(ss,sd)
_(ds,dd)
	Copy a Volume	(ss,sd)
		(ds,dd)
			Source Slot, Drive
			Destination Slot, Drive

FORMAT_(s,d) _ */volumename*	Format a Volume	(s,d)	Slot, Drive Number
MVV_*old.name* _*new.name*	Rename a Volume	-I	Prompt for New Name

Special Commands

Command Syntax	Description	Options and Arguments
CMP *file.1*_*file.2*	Compare Two Files	None
CMP_(ss,sd)_(ds,dd)	Compare Two Volumes	None
FIND_*directory* _-*filename*	Locate a File in a Directory(s)	None
GREP_*string* _*pathnames*	Locate a String in a File(s)	-N Show Line Numbers
SDIFF_*file.1*_*file.2*	Compare Two Text Files	-N Show Line Numbers

Abbreviations and Wild Cards

Command Syntax	Description	Options and Arguments
.	Abbreviation for Working Directory	None
..	Abbreviation for Parent Directory	None
?	Wildcard for a Single Character	None
*	Wildcard for a Character String	None
ECHO_*pathname*	List Wildcard Pathnames	None

Firmware Calls

Command Syntax	Description	Options and Arguments
C40	Disable the 80 Column Firmware	None
C80	Enable the 80 Column Firmware	None
DATE_yymmddhhmm	Set or Read System Date/Time	None
SD	Print Contents of Screen	None

KIX Utilities

Command Syntax	Description	Options and Arguments
CFG	Configure Printer and KIX Boot	None
INSTALL	Install KIX in AppleWorks	None
KIX	Show the KIX HELP Screen	None
QUIT	Exit the KIX Environment	None