

# **BOSS/IX (7.2)**

## **User Reference Manual**

**OCTOBER, 1986**  
**006210-000**

**M6210**

This manual is proprietary to and a trade secret of MAI Basic Four, Inc. ("MBF"). Possession and use of this software is restricted and subject to the terms of the standard MBF license agreement for the same. You shall not publish, reproduce or transmit, in whole or in part, this software in any form or by any means, whether electronic, mechanical, photographic, or otherwise. You shall not disclose, or fail to make your best efforts to prevent the inadvertent disclosure of, this software to any third party, except to your employees. If you fail to act in accordance with this notice and the standard MBF license agreement for this software, your license to this software will be immediately and automatically terminated, and you may expose yourself and others to civil and criminal liability.

©Copyright 1983 (Rev. 1986) by MAI Basic Four, Inc. All Rights Reserved. MAI and Basic Four are registered trademarks of MAI Basic Four, Inc.

MAI Basic Four, Inc., 14101 Myford Road, Tustin, CA 92680 (714) 731-5100

PAGE STATUS

Cover		Nov 1986
Title Page	i/ii	Nov 1986
Page Status	iii/iv	Nov 1986
Table of Contents	v through xii	Nov 1986
List of Figures	xiii/xiv	Nov 1986
List of Tables	xiii/xiv	Nov 1986
Preface	xv	Nov 1986
Section 1	1-1 through 1-6	Nov 1986
Section 2	2-1 through 2-30	Nov 1986
Section 3	3-1 through 3-18	Nov 1986
Section 4	4-1 through 4-188	Nov 1986
Section 5	5-1 through 5-218	Nov 1986
Section 6	6-1 through 6-62	Nov 1986
Appendix A	A-1 through A-4	Nov 1986
Appendix B	B-1 through B-10	Nov 1986
Index	I-1 through I-10	Nov 1986



## TABLE OF CONTENTS

	<b>Page</b>
<b>SECTION 1 - INTRODUCTION</b>	
BOSS/IX.....	1-1
Related Documentation.....	1-3
Conventions.....	1-3
<b>SECTION 2 - THE BOSS/IX FILE SYSTEM</b>	
File System Organization and Access.....	2-3
Files.....	2-3
File Organization.....	2-4
File Types.....	2-6
Filesystems.....	2-7
File Access.....	2-8
File and Directory Names.....	2-8
Path Names.....	2-11
Path Name Abbreviations.....	2-12
Directory Search Order.....	2-14
Accessing Additional Filesystems.....	2-15
File System Maintenance.....	2-19
File Security and Ownership.....	2-19
Disk and Diskette Maintenance.....	2-21
Disk and Filesystem Maintenance.....	2-24
Change File Size.....	2-25
File System Repair.....	2-26
BOSS/IX Utilities and Commands.....	2-26

## TABLE OF CONTENTS (cont'd)

	Page
<b>SECTION 3 - THE BOSS/IX PRINT SPOOLER</b>	
Introduction.....	3-1
The BOSS/IX Spooler.....	3-2
How the Spooler Works.....	3-3
Advantages of Using Spooling.....	3-4
Spooling Parameters.....	3-5
Default File Parameters.....	3-5
Print-Time Parameters.....	3-8
Default Files.....	3-9
Print Form Definitions.....	3-10
Print Form Parameters.....	3-11
Spooler Configuration Issues.....	3-12
Selecting Spooling.....	3-13
Bypassing Spooling.....	3-14
Automatic Form Feed.....	3-15
BOSS/IX Print Utilities and Commands.....	3-16
 <b>SECTION 4 - BOSS/IX UTILITIES</b>	
Introduction.....	4-1
Organization of this Section.....	4-1
Standard Utility Procedures.....	4-5
Accessing Utility Programs.....	4-5
Utility Prompts.....	4-10
Special Function Keys.....	4-14
File Selection Procedure.....	4-17
Pattern Matching Characters.....	4-23
Directory Utilities.....	4-25
Directory Display Utility.....	4-26
Create Directory Utility.....	4-29
Delete Directory Utility.....	4-31
Change Working Directory Utility.....	4-33
Change Directory Security Utility.....	4-35

## TABLE OF CONTENTS (cont'd)

	Page
SECTION 4 - BOSS/IX UTILITIES (cont'd)	
File Utilities.....	4-39
Create File Utility.....	4-41
Delete File Utility.....	4-44
Display File Information Utility.....	4-46
Rename File Utility.....	4-48
Copy File Utility.....	4-50
Move File Utility.....	4-53
Change File Size Utility.....	4-55
Filelist Maintenance Utility.....	4-58
Change File Security Utility.....	4-64
File Analysis and Repair Utility.....	4-66
Filesystem and Disk Utilities.....	4-71
Mount and Unmount Filesystem Utility.....	4-73
Format Diskette Utility.....	4-76
Copy Diskette Utility.....	4-78
Free Space Analysis Utility.....	4-81
Filesystem Error Analysis Utility.....	4-84
Printer Utilities.....	4-89
Submit Print Job Utility.....	4-90
Maintain Print Queue Utility.....	4-93
Change Print Form Utility.....	4-95
Printer Status Utility.....	4-97
Printer Parameters Utility.....	4-98
Save and Restore Utilities.....	4-109
Save to Cartridge Utility.....	4-110
Restore from Cartridge Utility.....	4-115
Report Save Cartridge Contents Utility....	4-119
Compare Save Cartridge Utility.....	4-121
Label Cartridge Utility.....	4-123
Save To Diskette Utility.....	4-125
Restore From Diskette Utility.....	4-130
Report Save Diskette Contents Utility....	4-134

## TABLE OF CONTENTS (cont'd)

	Page
<b>SECTION 4 - BOSS/IX UTILITIES (cont'd)</b>	
BASIC Program Utilities.....	4-135
BASIC Program Cross-reference Utility.....	4-136
BASIC Program Renumber Utility.....	4-139
BASIC Program String Search and Replace Utility.....	4-142
Compare BASIC Programs Utility.....	4-146
Merge BASIC Programs Utility.....	4-149
Dispaly BASIC Program Names Utility.....	4-152
Encrypt BASIC Program Utility.....	4-155
System Utilities.....	4-159
Date and Time Utility.....	4-160
Port Configuration Utility.....	4-164
Menu Editor Utility.....	4-173
Operator Information Utility.....	4-183

## SECTION 5 - COMMAND LANGUAGE

The Command Interpreter.....	5-1
Control Characters.....	5-3
The Command Language.....	5-4
General Syntax.....	5-6
Argument Syntax.....	5-6
Process Redirection.....	5-8
Background Processing.....	5-10
Pattern Matching.....	5-11
Additional Rules.....	5-14
Command Files.....	5-17
Creating Command Files.....	5-17
Executing Command Files.....	5-17
Variables in Command Files.....	5-18

## TABLE OF CONTENTS (cont'd)

	Page
SECTION 5 - COMMAND LANGUAGE (cont'd)	
Pound-Sign Commands.....	5-19
Formats and Uses of Pound-sign Commands.....	5-21
#a.....	5-22
#b.....	5-23
#d.....	5-25
#e.....	5-26
#h or #?.....	5-26
#l.....	5-27
#p.....	5-28
#q.....	5-29
#s.....	5-30
#x.....	5-31
#v.....	5-33
#!.....	5-34
BOSS/IX Commands.....	5-35
ad.....	5-41
addname.....	5-43
admin.....	5-45
advance.....	5-46
cat.....	5-47
cd.....	5-50
change.....	5-52
chown.....	5-55
command.....	5-56
copy.....	5-58
cpw.....	5-62
date.....	5-62
debe.....	5-65
delete.....	5-69
devfmt.....	5-71
diff.....	5-73
diskusage.....	5-75
do.....	5-77
echo.....	5-78
errlog.....	5-80
exec.....	5-82
filenodes.....	5-85
fschk.....	5-87

## TABLE OF CONTENTS (cont'd)

	Page
SECTION 5 - COMMAND LANGUAGE (cont')	
iconf.....	5-91
install.....	5-92
install_key.....	5-94
kill.....	5-95
kychk.....	5-96
login.....	5-98
lpmaint.....	5-100
lpq.....	5-104
lpr.....	5-106
lpstat.....	5-110
ls.....	5-111
makedev.....	5-114
makedir.....	5-116
makeeec.....	5-117
makefs.....	5-119
makesta.....	5-121
makettymntbl.....	5-122
makettyxlate.....	5-125
match.....	5-127
mcscompare.....	5-129
mcslabel.....	5-131
mcslist.....	5-133
mcsrestore.....	5-135
mcssave.....	5-139
message.....	5-143
mount.....	5-144
move.....	5-146
mtscompare.....	5-149
mtslabel.....	5-151
mtslist.....	5-153
mtsrestore.....	5-155
mtssave.....	5-159
p.....	5-163
pmask.....	5-166
pr.....	5-167
prompt.....	5-170
ps.....	5-172
pted.....	5-176
pwd.....	5-177

## TABLE OF CONTENTS (cont'd)

	Page
<b>SECTION 5 - COMMAND LANGUAGE (cont'd)</b>	
remark.....	5-178
resume.....	5-179
setmask.....	5-180
shutdown.....	5-182
sort.....	5-184
space.....	5-186
suspend.....	5-187
sync.....	5-188
sysinfo.....	5-189
ttymntbl.....	5-190
ttymodes.....	5-191
ttyxlate.....	5-199
unmount.....	5-200
usb.....	5-202
vconf.....	5-204
ved.....	5-208
wait.....	5-213
who.....	5-214
write.....	5-215

## SECTION 6 - PROBLEM SOLVING

Premature System Shutdown.....	6-3
What is a Premature Shutdown.....	6-3
What is a Memory Dump?.....	6-5
Recovery Procedure.....	6-7
Repairing the Filesystems.....	6-10
Filesystem Repair.....	6-11
Filesystem Repair by Utility.....	6-13
Filesystem Repair by Command.....	6-15
Remaining Errors.....	6-19
File Repair Procedures.....	6-20
Keyed File Repair.....	6-20
Lost File Recovery.....	6-21
Printer Problems.....	6-24
Symptoms and Fixes.....	6-24
Problem Areas to Check.....	6-27

## TABLE OF CONTENTS (cont'd)

	Page
SECTION 6 - PROBLEM SOLVING (cont'd)	
Startup Errors.....	6-30
Errors Causing Error Log Entries.....	6-30
Errors Causing Shutdown.....	6-34
Errors Causing a Crash.....	6-36
If the Problem Isn't Fixed.....	6-37
Diagnostic Programs.....	6-38
Fixed Disk Resident Diagnostics.....	6-39
Alternate Load.....	6-40
Start Printer Output.....	6-41
Running the Logic Tests.....	6-42
Load and Run the System Interaction Test..	6-44
MAI 2000 Disk Resident Diagnostics.....	6-47
Error Numbers.....	6-50
APPENDIX A - MINOR SYSTEM CONFIGURATION	
CHANGES.....	A-1
APPENDIX B - SYSTEM FILE FORMATS..... B-1	
Introduction.....	B-1
System Definition File Formats.....	B-1
Operator Definitions File.....	B-1
Port Definitions File.....	B-2
Printer Definitions File.....	B-3
Terminal Definitions File.....	B-5
Printer Parameter File Formats.....	B-5
Defaults File.....	B-6
Class File.....	B-7
Print Forms File Format.....	B-8
INDEX..... I-1	

## LIST OF FIGURES

	Page
2-1 File System "Tree" Structure.....	2-5
2-2 Disk and Diskette Structure.....	2-23
4-1 Sample Multiple Choice Prompt.....	4-10

## LIST OF TABLES

	Page
4-1 Menu Names for Utility Program Access.....	4-7
4-2 Utility Program Names.....	4-8
4-3 Special Function Keys.....	4-15
4-4 Control Sequences for Special Function Keys.....	4-17
4-5 Pattern Matching Characters.....	4-23
4-6 Permitted File Size Parameter Changes.....	4-55
5-1 Input/Output Redirection Symbols.....	5-9
5-2 Pattern Matching Characters.....	5-12
5-3 Pound-sign commands.....	5-20
5-4 Commands Summary by Function.....	5-36
5-5 "fschk" Exit Codes.....	5-89
5-7 "ved" Command Summary.....	5-211



## PREFACE

This BOSS/IX User Reference Manual contains reference information useful for the daily operation and maintenance of a system running the BOSS/IX (Level 7.2) operating system. The descriptions of the system are more complete and detailed than that given in the system user guides. On the other hand, technical information pertaining to the configuration of the system is excluded, being given in the BOSS/IX Technical Reference Manual.

The topics covered in this manual include:

Section 1 - Introduction to BOSS/IX

Section 2 - The BOSS/IX File System

Section 3 - The BOSS/IX Spooler

Section 4 - BOSS/IX Utilities Programs

Section 5 - BOSS/IX Command Language

Section 6 - Problem Solving

Appendix A - Minor Configuration Changes

Appendix B - System File Formats



## SECTION 1

### INTRODUCTION

The BOSS/IX operating system provides many features and capabilities beyond the needs of most users. General information and instructions for routine operation of a system operating under BOSS/IX are described in the User Guide for each system.

This manual, the BOSS/IX (7.2) User Reference Manual, provides information on those features of BOSS/IX needed by the system administrator, who requires more flexible and powerful control of the system. Topics covered here include:

- o Full description of the Utility Programs
- o Description of the Command Language
- o File System Maintenance
- o Description of Print Spooling
- o Problem Solving
- o Minor Configuration Changes

Advanced features of BOSS/IX are beyond the scope of this manual, and are discussed in the BOSS/IX Technical Reference Manual.

### BOSS/IX

BOSS/IX is a powerful, business-oriented operating system developed by MAI<sup>(R)</sup> Basic Four Inc. to run on its line of microcomputers. Primary features are:

- o Multi-User Support
- o Fixed-disk Based, for massive data storage and quick access
- o UNIX<sup>(R)</sup>-like file management and command interpreter
- o Extensive, menu-driven utility set
- o Business BASIC Level 7

---

UNIX is a registered trademark of Bell Laboratories

BOSS/IX is based on the UNOS™ operating system, the UNIX-like O.S. developed by Charles River Data Systems. On this base, MAI Basic Four Inc. has built an operating environment ideally suited to use in business applications.

The BOSS/IX environment allows for both a secure, menu-driven interface, providing easy access to the system for most users, while at the same time providing powerful tools for the system administrator.

Multiple user support means that several users can be logged on and using the system at the same time. The number of users that can operate varies with the system and configuration.

The UNIX-like file system allows efficient management of very large capacity fixed disks. The file system is a hierarchical arrangement of files and directories that encourages organizing files into related clusters and allows for random access to any point in the file system. A fixed disk can further be "partitioned," with each partition containing its own file system. Files on any partition can be accessed at any time by "mounting" that file system to the currently active file system.

The Utility Programs provide an interactive interface for all basic system maintenance functions. All required parameters are provided by the user in response to prompts and menus. Help screens are available at most prompts to provide brief descriptions of the option or requested information.

---

UNOS is a trademark of Charles River Data Systems

The Command Language provides a less friendly, but faster and more flexible interface for performing system maintenance functions. Extensive pattern matching, abbreviation (macro) and I/O redirection capabilities make the command interpreter extremely flexible and powerful. Command files can be easily created to form special routines by command streaming.

## **RELATED DOCUMENTATION**

- o MAI 2000 User Guide, M6201
- o MAI 3000 User Guide, M6202
- o BOSS/IX Technical Reference Manual, M6224
- o Business BASIC Level 7 Reference Manual, M6252
- o MAI 2000 Service Manual, M8079
- o MAI 3000 Service Manual, M8108
- o MAI 3000 Diagnostics and Error Logging, M6204

## **CONVENTIONS**

Certain notational and terminological conventions have been adopted to make it clear what is meant in the discussions and what actions you are intended to take in procedures.

### **Referencing Conventions**

#### File Names

File names, including directory and program names, are enclosed in double quotation marks when mentioned in text. E.g., "/etc/motd" is the message-of-the-day file. Quotation marks are not used when file names are listed apart from running text.

## Prompts

Prompts displayed by a program are displayed on a line separate from running text, bolded and indented. Note that a prompt may be shown on two lines in the documentation but use only one line on the terminal screen. This is made necessary by page width restrictions. Prompts shown in the text may also differ slightly from those actually displayed, but the sense is the same.

The command interpreter prompt is different for each user. Rather than use a specific prompt, in many cases we use a general command interpreter prompt:

@>

Understand that your prompt will be different.

## **Keyboard Data Entry**

### Keys

Keyboard keys are indicated by their name, usually as marked on the keyboard, enclosed in left and right angles. E.g., <RETURN> indicates the key that corresponds to the carriage return key on a typewriter but may be marked "RETURN" or "ENTER", or in any of several ways depending on the specific terminal. The "A" key is indicated as <A>.

## Pressing Keys

The key notation above is most frequently used when you are instructed to press a key. For instance, to execute most commands you must press <RETURN>.

In some cases two or more keys must be pressed simultaneously, as is necessary to enter control sequences. These sequences are indicated as the keys joined with a plus sign (+).

E.g., <CTRL>+<C> is the control sequence entered by pressing <C> while holding down <CTRL>.

If a key is to be pressed following the control sequence, it follows the sequence separated by a space, e.g.:

<CTRL>+<C> <Y>

This indicates that the <CTRL>+<Y> command sequence is pressed, and then <Y> is pressed. (This sequence is used to exit the text editor).

## Typing Data

Data that is to be typed at a prompt is displayed on an indented line separate from the text. E.g., to enter the menu system, type:

menu

This type of instruction is usually followed by instructions to press <RETURN>.

NOTES

## SECTION 2

### THE BOSS/IX FILE SYSTEM

One of the most important functions of a computer operating system is storing and retrieving of data. BOSS/IX provides a powerful filing system, called the Business File System (or BFS), which has been tailored to the needs of the business computer user.

The Business File System has four primary features:

- o The file system provides files that may be accessed randomly or sequentially, providing all the access methods necessary for business applications.
- o The file directory system allows the organization of files into logical groups, making access to files very quick.
- o The filing system has security provisions so that you can make certain that files are protected from unauthorized access.
- o File growth and location is handled by BFS. You generally do not need to be concerned about the precise location or size of files.

This section explains what the BOSS/IX file system is and how to use it to maintain files, directories and filesystems. The description of the file system includes the hierarchical, or "tree" structure that logically organizes files in directories.

The utilities and commands that maintain the file system are fully described in sections 4 and 5 of this manual. The discussion here is restricted to which utilities and commands are required and when with some brief examples.

A brief description of fixed disk and floppy diskette organization is included in the subsection, "File System Maintenance." The procedures required for creating filesystems on a floppy diskette or other fixed disk partitions are also outlined there.

A technical description of the file system is given in the BOSS/IX Technical Reference Manual (M6224), and should be referred to for more detailed information on disk and filesystem structures.

## FILE SYSTEM ORGANIZATION AND ACCESS

In this subsection we give a description of the BOSS/IX file system. BOSS/IX is a disk-based operating system; the primary file system is located on a fixed disk storage device.

On some BOSS/IX systems, removable floppy diskettes can be used as an extension of this fixed disk file system; files can be used on the floppy diskette directly. On all BOSS/IX systems, files can be written to and read from tape devices (MCS and MTS), but the files must be copied to the file system before they can be used.

Most of the management of the file system is handled by the applications being run on the system. The information provided here is intended to help you understand file organization and to allow you more control over how the system is used.

### Files

Files are used to collect information, usually in an organized manner, for use at a later time. Information, or "data", is then added to, deleted from or retrieved from the file.

Computers use files in the same way. The file organization is determined by the program or programs using the file. Programs are themselves files of data, or instructions, that the system understands.

These files are stored on the fixed disk, floppy diskette or tape, similarly to the way normal files are stored in filing cabinets and boxes. Fixed disks, floppy diskettes and tapes are all "magnetic storage devices"; information is written on them similarly to the way sounds are saved on a cassette tape.

## File Organization

Files are not usually stuffed into boxes in disorganized ways, but are labeled and sorted alphabetically, and grouped so that related files are kept close together for easy reference. Some older computers were not very organized in the way they stored files, but just kept them on a disk in whatever order they were created.

BOSS/IX provides a system for organizing the files on the disk, so that related files can be stored together and away from unrelated files. This is done by a system of "directories". A directory is a place where files can be collected. (It is actually a special type of file that contains file names and information required to locate the files on disk.)

Directories can also contain directories, called sub-directories, which in turn can contain files and/or directories, and so on. The resulting organization is hierarchical. A single starting directory usually contains several directories, and perhaps some files. These sub-directories contain files and/or more directories. This pattern is repeated until each file is located in some directory.

The hierarchical structure produced in this fashion is usually pictured as a "tree", with the starting directory at the root of the tree. Each subsequent directory provides a branching point in the tree. Branches terminate with files (other than directories), which can be thought of as leaves. The metaphor is not perfect; leaves, or files, can occur at any branching point, and a directory can be empty.

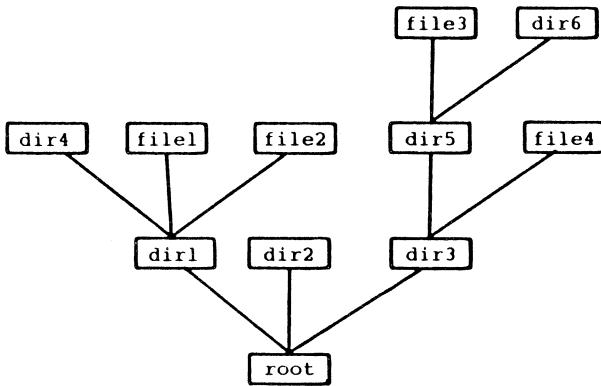


Figure 2-1. File System "Tree" Structure

The only part of this "tree" metaphor that is carried over into the remainder of the discussion is that the starting directory in each filesystem (defined below) is called the "root" directory of that filesystem. In addition, the default filesystem, the one activated (unless specified otherwise) when the system is booted, is called the "root" filesystem.

## File Types

BOSS/IX recognizes the following file types:

- o String
- o Direct
- o Sort
- o Indexed
- o Serial
- o BASIC Program
- o Device
- o Station
- o Eventcount
- o Directory

It is beyond the scope of this manual to describe the structures and use of these file types. Refer to the BOSS/IX Technical Reference Manual and the Business BASIC Level 7 Reference Manual for detailed information.

The primary file type is the String file type. A String file is a series of bytes on disk with no structure imposed on it other than that determined by the programs that read and write it. Files created by the text editor, VED, and files printed using the print spooler are of this type, as are also most system files.

Directory files are simply that, special files that are directories.

Direct and Sort files are often listed as KEY or KEYED files. Keyed, Indexed and Serial files are used as data files primarily by Business BASIC programs. Business BASIC programs are the only files of type BASIC Program.

Device, station and eventcount are special files used only by BOSS/IX for a variety of system functions. Device files, which are generally in the "/dev" directory, define devices such as terminals, printers, disk drives, and so on.

## **Filesystems**

A "filesystem" (one word) refers to an actual implementation by the BFS of the file and directory structure described above. Each filesystem has a unique "root" directory, named "/", which is created when the filesystem is created. Additional directories and files are created later through use of the filesystem.

Filesystems are individual things that exist on separate fixed disk partitions and floppy diskettes. Magnetic Cartridge Streamer (MCS) and Magnetic Tape Streamer (MTS) tapes are not filesystem devices.

When the system is booted, the root filesystem is activated so that all files and directories in it can be accessed. The files and directories in other filesystems cannot be accessed yet, but must be activated by process of "mounting."

When a filesystem is mounted, an association is made by the O.S. between a directory in an already activated filesystem (usually in the root filesystem) and the root directory of the filesystem to be activated. The second filesystem is then said to be "mounted" to the first filesystem. The directory in the first filesystem is called the "mount point" directory.

Once a filesystem has been activated by mounting, the files and directories in it are accessible in the same way as files in the root filesystem. Files and directories in a mounted filesystem are accessed through the mount point directory just as if the mounted filesystem were part of the root filesystem.

One caution is necessary here. If the mount point directory contains files or directories of its own, these files and directories are not available while a filesystem is mounted to it. All references to this directory are interpreted as references to the root directory of the mounted filesystem. For this reason, empty directories should be created for commonly mounted filesystem. The "/mnt" directory is provided for this purpose.

## **FILE ACCESS**

The above discussion described the file system structure in fairly abstract terms. The following paragraphs will be more concrete and specific about file access methods.

### **File and Directory Names**

Files and directories are identified and accessed by their names. A file or directory name is a string of up to 20 characters, and can include alphabetic characters, numbers and special characters, with only a few restrictions.

A period (.) can be used to begin a file name "extension." Unlike some operating systems, there is nothing special about the treatment of extensions. The period and extension are counted among the 20 characters allowed for a file name. Some programs use extensions to mark files used for special purposes, for example, ".bak" may be used for backup files, ".idx" may be used for index files, and so on.

A file name that begins with a period is treated as a "hidden" file and not listed by the "ls" command unless the "-all" option is used.

BOSS/IX distinguishes between upper and lower case alphabetic characters in file names. So, "afile" and "Afile" are different names. Be sure when specifying a file name that all characters are correct.

The following characters should be avoided in file and directory names:

space \* + - / ! @ ? { } # [ ]  
^ \ \$ " : < > ( ) & '

In addition, control characters and the null character (hexadecimal 00) should be avoided.

The characters listed above have special meaning to BOSS/IX, such as separating argument fields, redirection input and output and pattern matching. Avoiding these characters when naming files and directories will prevent difficulties in accessing files.

### An Example

The following sample (and much abbreviated) file system will be used in the subsequent description. It consists of two filesystems, the root filesystem and another filesystem located either on a fixed disk partition or on a floppy diskette.

Root Filesystem:

```
Root Directory (/)
  bin (directory)
    command (file)
    copy (file)
    ls (file)
    .
    .
    .

  usr (directory)
    john (directory)
      memos (directory)
        memo1
        memo2
        wrdproc (file)
        .
        .
        .

    fred (directory)
      memos (directory)
        memo1
        memo2
        wrdproc (file)
        .
        .
        .

    kim (directory)
    .
    .
    .

  util (directory)
    menu (file)
    fdelete (file)
    fcopy (file)
    smount (file)
    .
    .
    .

  accnting (directory, empty)
```

## Accounting Filesystem:

Root Directory (/)

```
payroll (directory)
  exempt (file)
  nexempt (file)
GNLDGR (directory)
cap-eqpt (directory)
```

## Path Names

A file system may have more than one file or directory with the same name. What uniquely identifies a file or directory is its path name.

A path name is a full list of the directories preceding the file or directory being accessed, separated by slashes (/), beginning at the root directory. The root directory is indicated by a slash at the very beginning of the path name.

For example, John's word processing program "wrdproc" is fully identified by the path name "/usr/john/wrdproc". The first slash indicates that the path starts at the root. The directories "usr" and "john" are then specified, separated by slashes. Finally, the program file name is given.

This word processing program is distinguished from Fred's by its path name. Fred's program is "/usr/fred/wrdproc". So, even though more than one file may have the same name, the path name will distinguish the files.

## **Path Name Abbreviations**

To simplify the specification of path names, BOSS/IX recognizes a few ways of specifying frequently accessed directories.

### Working and Alternate Directories

Working and alternate directories specify paths from the root directory to another directory. To specify a file, BOSS/IX allows you to omit that part of the directory path that is a working or alternate directory. This reduces the amount of typing required to specify a file.

Note that the working directory can be used for reading, writing and executing files, whereas alternate directories can only be used for reading and executing.

A file (or directory) is specified by giving the "partial path name" relative to the working (or alternate) directory. There is no initial slash, since this would be interpreted as the root directory. The full path is constructed by BOSS/IX by prefixing the working (or alternate) directory to the specified partial path name.

If John's working directory is "/usr/john" (as it usually would be when he logs on), he can access the file "memo2" by specifying "memos/memo2" instead of having to specify the full path name. Even shorter, if his working directory is "/usr/john/memos", he can specify "memo2" alone.

A separate set of directory paths is kept by BOSS/IX specifically for command execution. These can be viewed and modified using the "#x" command "PATH=" option as described in Section 5 of this manual.

In Business BASIC, working and alternate directory path names are maintained as a "prefix list". Refer to the Business BASIC Level 7 Reference Manual for more information.

### Defining Working and Alternate Directories

A working directory is defined for each user when he or she logs on to the system. This default working directory is called the user's "home directory." The home directory is specified by the Operator Information Utility when the user is defined, and is usually "/usr/XXX", where XXX is the user's logon (account) name.

Once the user is logged on, their working directory can be changed by the Change Working Directory Utility, the "cd" command, or any of several utilities that allow changing the working directory.

Alternate directories work the same way as the working directory, in the command language, but are not recognized by the utility programs. The only way to add alternate directories is by using the "ad" command.

### Dot Abbreviations

The double dot (..) is recognized by the O.S. as a reference to the directory immediately preceding the current working directory, the "parent directory." If the working directory is "/usr/john", ".." refers to "/usr" and "../fred" refers to "/usr/fred".

Similarly, a single dot (.) can be used to refer to the current working directory. This is seldom useful.

## Directory Search Order

BOSS/IX follows a determined order when searching for files, but the order differs based on whether the file is being accessed for data input and output or for execution as a program. The first term entered on a command line, immediately following a pipe or immediately following a semicolon in a compound command (as described in Section 4) is presumed to be a program file. All other file references are taken as data files.

A file specified with a full path name is accessed directly, without going through this search procedure.

When BOSS/IX searches for a data file that has been specified with a partial path name (lacking the initial slash), it searches the working directory first, and then it searches the alternate directories, if any, in the order in which they are specified. The first file found matching the specified partial path name is used.

If the data file is referenced for output (such as if output redirection has been used in the command language), the first file found in the search order matching the specified name is used. If no file with that name exists in either the working or an alternate directory, the file is created in the working directory.

The search for a program file is governed by the user's environment (refer to the #x command in Section 4). The usual order is "/bin", "/util", the home directory, the working directory (if different), and then the alternate directories in the order specified.

## **Accessing Additional Filesystems**

When the system is initially booted, only the root filesystem is accessible. Files in other filesystems, on either floppy diskettes or fixed disk partitions, can only be accessed when the filesystem has been "mounted" to the root filesystem or another, already mounted filesystem.

These additional filesystems may be located on fixed disk partitions or floppy diskettes. In either case, the procedure for mounting the filesystem is the same, only the device specified for mounting is different.

### **Mounting Filesystems**

The Mount and Unmount Filesystem Utility and "mount" and "umount" commands are the tools provided for mounting and unmounting filesystems. They are described in Sections 4 and 5, respectively. Both mount procedures require that you specify the filesystem device (fixed disk partition or floppy diskette drive), and the directory to serve as the mount point.

For instance, if the accounting filesystem described above exists on the fixed disk partition "/dev/acntng", the command:

```
@>mount /dev/acntng /mnt
```

mounts the filesystem on the fixed disk partition "/dev/acntng" to the "/mnt" directory in the root filesystem. The directory "/mnt" should be left empty for use as a mount point.

The floppy drives have consistent names, starting with "/dev/fd0" as the first floppy. This generic command will mount the filesystem on a floppy in this drive:

```
@>mount /dev/fd0 /mnt
```

Note the following restrictions on mounting:

1. Only one filesystem can be mounted to a directory at a time. An attempt to mount a second device to an already used directory results in a "directory busy" message
2. If the mount point directory is not empty, any files in that directory are inaccessible while a filesystem is mounted to it. For this reason, it is a good practice to create special, empty directories to be used exclusively as mount points.
3. A floppy drive can only support one mounted floppy at a time. If a floppy drive has been used to mount a filesystem, that filesystem must be unmounted before the floppy drive can be used again.

#### Unmounting Filesystems

When a mounted filesystem is no longer being accessed it should be unmounted. This frees certain system resources and protects the filesystem from accidental damage. A limited number of filesystems may be mounted at one time, so unmounting a filesystem not in use allows another required filesystem to be mounted.

The Mount and Unmount Filesystem Utility and "umount" command are used to unmount filesystems, as described in Sections 4 and 5. For both procedures, enter the name of the mount point directory to be unmounted. (The command version allows you to specify the mounted device instead of the directory.) For example, this command will unmount the filesystem mounted above:

```
@>umount /mnt
```

The "/mnt" directory is then available for mounting.

Note that all files in the filesystem must be closed and no user's working directory may include either the mountpoint directory or any directory in the filesystem. An attempt to unmount the filesystem will result in a "directory or device busy" message.

Floppy diskettes should always be unmounted before removing them from the drive. If you don't unmount a floppy before removing it from the drive, the system will detect this when you put another floppy in and try to use it. The system expects you to put the original mounted diskette back and will issue a warning if a different diskette is there.

### Accessing Files in Mounted Filesystems

Once a filesystem is mounted, the directories and files in it are accessed in the same way as files in the root filesystem. The root directory of the mounted filesystem is identified by the name of the mount point directory. The path name runs from the root directory of the root filesystem, through the mount point directory, and to the desired directory or file in the mounted filesystem.

In the above example, if the accounting filesystem is mounted to the "/mnt" directory, the "exempt" file is accessed by the path name "/mnt/payroll/exempt". The initial slash refers to the root directory of the root filesystem. "/mnt" now refers to the root directory of the accounting filesystem, and then the rest of the path is as usual.

#### Read-Only Mounting

BOSS/IX allows a filesystem to be mounted in "read-only" mode, allowing files in the filesystem to be read but not written. This option is only available with the "mount" command, not with the utility.

To mount a filesystem in read-only mode, use the "-r" option with the command.

## **FILE SYSTEM MAINTENANCE**

This subsection covers a variety of topics pertaining to the maintenance of the file system. Topics include:

- o File security and ownership
- o Disk and diskette maintenance
- o List of file system related utilities and commands

### **File Security and Ownership**

BOSS/IX incorporates several security features. The following paragraphs describe the security features incorporated into the file system.

### **File and Directory Ownership**

The file and directory related security features involve ownership rights at directory and file levels. When a file or directory is created, ownership is assigned to the user who created it.

BFS keeps track of ownership by operator number (assigned by the operator information utility).

When necessary, the system administrator can change file ownership using the Change File Security and Change Directory Security Utilities or the "chown" command.

Ownership also changes when a user creates, moves, copies or edits a file or directory. The user becomes the owner of the new file, but if an old version remains, or if a backup copy is made, it retains the old owner.

## File and Directory Access Rights

File and directory access modes are assigned for each of two classes of user: the owner and system administrator, and "the world", i.e., all other users. For each of these classes of users the modes permit or deny read, write and execute access to the file or directory. The meaning of an access mode differs depending on whether the access is to a file or to a directory.

When a file or directory is created, it is assigned a default set of access modes. These file access modes can be changed by the owner or the system administrator by using the Change File Security or Change Directory Security Utilities or the "filenames" command. The default filenames can be changed by the system administrator using the "setmask" command.

For files, the meanings of the access modes is straightforward. Read access allows users to read from the file, write access allows users to write to the file, and execute access allows users to execute the (program or command) file.

The meaning of the access modes for directories is less obvious. Read access allows listing of the directory's contents. Write access allows creating and deleting of files in the directory. Execute access allows accessing the files in the directory for processing, such as when running a program with the file as input. If execute permission to a directory is denied, all access to all files in the directory is denied, and the directory cannot be used as (part of) the working directory.

Directory access modes have no effect on the system administrator's access to the directory. File access modes do effect the system administrator's file access rights, but he can change the modes using the "filemodes" command.

It is important to note that even files with only read access, but in an unrestricted directory, may still be moved, copied, deleted and edited by non-owners. Doing any of these procedures also changes the file owner to the operator performing the command. To protect your files completely you must restrict access to the directory.

**DO NOT RESTRICT execute access to the following directories:**

/ /dev /etc /sys /tmp /usr

These directories contain files that must be accessed during normal system operations.

### **Disk and Diskette Maintenance**

This section begins with a brief description of how disks and diskettes are organized and how files are allocated space on them. Once this is understood, some aspects of disk and diskette maintenance are described.

#### Disk and Diskette Organization

A diskette is a thin disk of mylar, approximately five inches in diameter, enclosed in a protective cover. A fixed disk is a stack of rigid disks enclosed in an airproof case. In both cases, the surfaces of the disk platters are coated with a metallic oxide like that used on recording tape.

When a disk or floppy is in use, the disk is rotated at a high speed. Read/write heads move in and out along a radius on each platter surface. In the case of floppies, the heads actually contact the disk surface, but in fixed disks they do not. The heads locate at discrete positions along the radius while the disk rotates beneath them.

The circle defined by the rotation of a disk platter beneath a read/write head at a location is called a "track". Each platter surface has a number of concentric tracks. Each track is further divided into a set number of "sectors", which are the length on a track required to write 1 block (512 bytes) of data. A stack of tracks, or the set of tracks, one per surface, at the same head location, is called a "cylinder." A track is usually identified by the cylinder number and the head number.

There are nine sectors per track on BOSS/IX disks and diskettes. The number of tracks per platter surface varies with the device as does also the number of platters.

Figure 2-2 illustrates the structure of floppy diskettes and fixed disks in the terms described above.

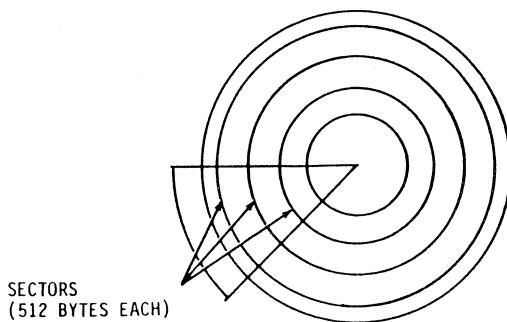
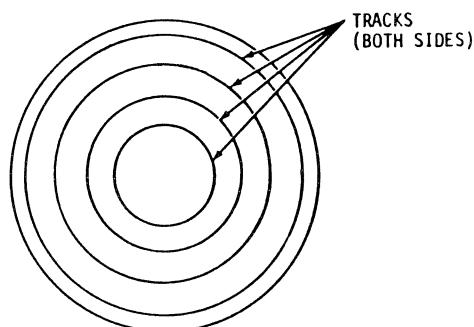
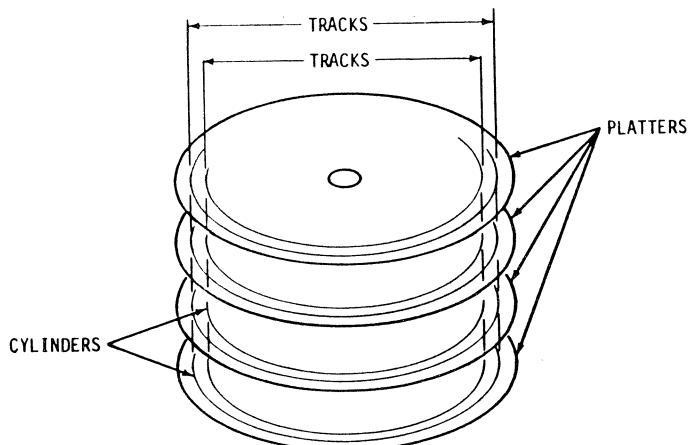


Figure 2-2. Disk and Diskette Structure

## Disk Partitions and Filesystems

The sectors on a single disk or floppy are identified sequentially from 0 to a large number, e.g., approximately 233580 for a 120 MB fixed disk.

On fixed disks, BOSS/IX maintains disk "partitions" as sequences of contiguous sectors. Partitions do not overlap, i.e. do not have any sectors in common, except that there is always one partition defined as the whole disk in addition to any smaller partitions. Partitioning is only supported on fixed disks. The only partition on floppy diskettes is the whole diskette.

Each partition may contain one filesystem, but some partitions do not. For instance, the partition on the first fixed disk that is used for data swapping, "/dev/swap", does not contain a filesystem, nor do floppies used by the backup and restore utilities. The root partition, boot partition, and partitions on additional fixed disks used for data and program storage do have filesystems.

## Creating Filesystems

A filesystem is created on a floppy diskette when it is formatted using the Format Diskette Utility. Refer to Section 4 for a description of the utility.

Instructions for creating partitions and filesystem on fixed disks are beyond the scope of this manual. Standard filesystems are created on the first fixed disk during the operating system installation procedure. Partitions and filesystems are created on additional fixed disks by a rather involved procedure involving several commands, and is potentially destructive to data. These procedures are described in the BOSS/IX Technical Reference Manual.

## **Disk and Filesystem Maintenance**

With the above discussion as background, the following paragraphs describe a few essential points of disk and filesystem maintenance.

### Disk Space

Filesystems are created with a capacity for a limited number of files, and disks have a limited amount of space for storing data. Both the number of files and amount of data stored on disks increases as the system is used. When either of these resources are exhausted a program requiring more space will fail to operate.

Accordingly, when you are using programs that create and write to files, it is necessary to keep track of the amount of file space available on the system. Being aware of the amount of available and used file space, and even the rate at which it is being used, allows you to prepare for exhausted resources before the situation becomes critical.

There is a utility program and a command that provide information on disk usage: the Filesystem Free Space Utility and the "space" command. Both of these programs are described in later sections of this manual.

These programs report filesystem usage in the following terms:

- o Blocks Free (1 block = 512 bytes)
- o Blocks Used
- o Total Blocks
- o Percentage of Blocks Used
- o File Descriptors Free
- o File Descriptors Used
- o Total File Descriptors
- o Percentage of File Descriptors Used

The numbers given for blocks reflects the actual disk space used, and the numbers given for file descriptors reflects the number of file entries used.

Both of these programs reflect only the usage of the filesystem device specified. The usage of a complete disk can only be determined by checking each filesystem device defined on that disk.

If disk space is running out, you should make more disk space available by either upgrading the disk capacity of your system or eliminating files from the affected filesystem. Files that are no longer in use should be deleted. Files that are not frequently used can be archived or moved to another filesystem with more space.

The amount of disk space used by the files in a directory is reported by the command "diskusage". The amount of disk space used by individual files is reported by the Directory Display Utility and the "ls -l" command.

### **Change File Size**

BFS files other than String and BASIC Program files are created with a maximum file size. In the course of use, these files may become full. In most cases these files contain valuable data, and so they cannot be deleted or initialized.

The Change File Size Utility is provided to allow you to enlarge these files. In general, you only need to change the maximum number of records for the file. The other parameters do not affect the amount of data a file may contain.

Refer to Section 4 for more information on the Change File Size Utility.

## **File System Repair**

There are many ways in which the file system can be damaged. Damage can occur to a filesystem or to individual files.

Refer to Section 6, "Problem Solving," for repair procedures for files and filesystems.

## **BOSS/IX UTILITIES AND COMMANDS**

The following is a list of BOSS/IX utilities and commands to perform the main file system maintenance functions.

**Utilities** (by title and program name):

Change Working Directory	workdir
Change File Size	fchange
Copy File	fcopy
Create Directory	dcreate
Create File	fcreate
Delete Directory	ddelete
Delete File	fdelete
Display Directory Contents	ddisplay
Filelist Maintenance	filelist
Format Diskette	sformat
Filesystem Free Space	freespace
Move File	fmove
Rename File	frename

**Commands** (program name and description):

cd	changes the working directory
copy	copies files and directories
delete	deletes files and directories
ls	lists directory contents
makedir	creates directories
makefs	creates a filesystem on a partition
move	moves files and directories
pwd	displays the working directory
space	reports filesystem free space

## NOTES

## NOTES

## SECTION 3

### THE BOSS/IX PRINT SPOOLER

#### INTRODUCTION

This section describes the features of the BOSS/IX print spooling facility. For descriptions of the utilities and commands governing the spooler, refer to Sections 4 and 5 of this manual.

Under most circumstances control of print jobs is handled by the application software that you run. The application program controls the selection of printers, forms, priorities, and so on. For the special features of printing from the application program, refer to the documentation for the program.

There are several situations in which control of printing must be handled more directly from the command interpreter or through the utilities. Spooler control may also be necessary from your own Business BASIC programs. This chapter contains the information you need to have this kind of printing control.

Simple procedures for performing standard printing functions using the BOSS/IX Printing Utilities are given in the User Guide accompanying your system. This section deals with spooler control features specifically.

A number of spooler configuration issues are discussed at the end of this section. These describe how spooling can be turned ON and OFF for a specific printer, and a few related issues.

## THE BOSS/IX SPOOLER

The basic job of spooler software is to queue print jobs, allowing several jobs to be submitted at one time and then printing them one at a time. This provides a great performance improvement over systems that allow a user to submit a print job but then do not accept any more jobs until the first one is complete.

In addition to the print job queuing capacity, the BOSS/IX spooler allows for a high degree of control over system printers. The print job characteristics (parameters) described below condition a print job by specifying the printer, a special form, the number of copies, the date to print and many more options.

A hierarchy of parameter default definitions is implemented to simplify the specification of the parameters. These defaults are contained in two parameter default files, "/etc/class" and "/etc/defaults". Parameters supplied from these definitions can be overridden by specifying special values for any parameter.

Spooling can be enabled or disabled for individual printers. This is done through the Port Configuration Utility when the printer is set up initially, but can be changed by the same utility at a later time.

Spooling can also be bypassed for specific print jobs, either through a specific spooler option or through process redirection. These possibilities allow a printer to be captured for dedicated use when necessary.

Several utilities and commands are provided to control and maintain the spooler, including the following:

<u>Utility</u>	<u>Command</u>
Submit Print Job	lpr
Maintain Print Queue	lpmaint
Printer Status	lpq, lpstat
Change Print Form	lpmaint -fc
Printer Parameters	None, but parameter files can be edited

These utilities and commands are fully described in Sections 4 and 5 of this manual.

Spooling is also supported from the Business BASIC, COBOL and C programming languages.

#### **HOW THE SPOOLER WORKS**

Spooling is handled in BOSS/IX by two programs; the spooler and the despooler.

When data (e.g., a file) is sent to a spooled printer an entry is added to the queue file ("*/etc/\_queues/lpq.que*"). If the job is submitted by another process (e.g., a program sending output to a spooled printer), a temporary print ready file is created in the "*/etc/\_qtemp*" directory. No file is created for String files that are already print ready (plain ASCII String files) that are submitted directly to the spooler by the Submit Print Job Utility or the "lpr" command; the file itself is used.

The despooler program continuously examines the print queue and, based on print job parameters, has the files printed on the specified printers.

The despooler program is "/sys/lpd". There are a few occasions, described later in this section and in Section 6 of this manual, Problem Solving, when this program needs to be executed or its process ID located.

The despooler performs these kinds of related tasks as they are required:

- o It allows the user to change forms as different jobs require.
- o It detects and reports certain printer errors such as running out of paper, the cable being accidentally unplugged, and the printer being left offline. The despooler waits until these problems are corrected.
- o It allows you to print multiple copies of your job.
- o It can notify you by means of a message on your terminal display screen when the job is finished

#### **ADVANTAGES OF USING SPOOLING**

Using spooling has several advantages over sending data directly to the printer:

1. It allows you to make your print jobs wait in "queues" for their turns to be printed. Thus, users on different workstations running different applications may send data to the same printer at the same time. The spooler handles the problem of printing each job in order.

2. A program often sends data to the printer faster than the printer can handle. This situation is handled efficiently without requiring you to be concerned.
3. Also users do not need to wait for the print operation to finish before moving on to another program. The spooling system manages which jobs are printed in which order and how jobs are printed.
4. If different print jobs require different forms, the spooler can keep track of the forms required and the form currently on the printer. When a form change is required, the user is notified before printing is continued.

## **SPOOLING PARAMETERS**

The following is a description of the parameters that can be specified for print jobs. Except as noted, these parameters can be specified when submitting a print job using the Submit Print Job Utility, the "lpr" command, or the "opts=" I/O option in Business BASIC.

The first group of parameters are given values in two files that are used by the spooler to assign default values to a print job. The second group of parameters are assigned at the time a job is submitted, but also have default values.

### **Default File Parameters**

These parameters are assigned values in the "/etc/class" and "/etc/defaults" files, described below. These files are then used to determine default values for print jobs submitted to the spooler.

Priority

An integer ranging from 0 to 9 to be assigned to the print job. The higher priority jobs are printed first.

Each printer is assigned a hold priority. A print job with a priority number equal to or less than the hold priority causes the print job to remain unprinted until the priority is raised above the hold priority.

Submitting a print job with a priority higher than that of a job already printing does not stop the printing of this job. The next job that is printed is selected according to priority.

#### **Form name**

This is the name of a form definition that specifies form attributes such as the form size and vertical tab settings. The form definition parameters are described below.

#### **Copies**

This is a number from 1 to 999 specifying the number of copies to be printed.

#### **Delete**

This is a "yes" or "no" parameter telling the system whether or not to delete the file upon completion of the print job. Normally this parameter is set to the default value "no". This option is ignored if the printer is not configured for spooling.

The default is that temporary files (created by the spooler in the "/etc/\_qtemp" directory) are deleted and other printable files are not deleted.

#### Requeue

This is a "yes" or "no" parameter telling the system whether or not to place this print job back in the queue again after completion. The job is requeued with a priority of 0. It is then held until the priority is raised to above the hold priority and the job is restarted (use the Maintain Print Queue Utility or "lpmaint" command). This option is ignored if the printer is not configured for spooling.

#### Alias

This is a job name (alias) that is used when the user is notified (see the "notify" option) that the job is completed or that there is a problem. If an alias is not specified the file name is used.

#### Notify

This is a "yes" or "no" parameter telling the system to notify the user when the print job is completed or in case of a problem (e.g., running out of paper). The user must be logged on in order to receive the message.

#### Spooling

This is a "yes" or "no" parameter telling the system whether to process the job through the spooling queue or to send it directly to the printer.

### **Despool wait**

This is a "yes" or "no" parameter telling the system whether to wait on printing this job until the entire job has been received or to begin printing as soon as possible.

This option maximizes the printer use, since the despooler may be trying to access data faster than an application provides it.

### **Print Time Parameters**

At the time a print job is submitted for printing, these additional parameters may be specified:

#### **Class**

This is a previously defined print job class name which set values for the above parameters.

#### **Date**

This parameter specifies the date the job should be printed. The job is held until the specified date, and is then processed by the despooler.

#### **Printer**

The name of the printer is specified at the time the print job is submitted.

### **Raw mode**

This is a "yes" or "no" parameter. If set to "yes", it suppresses processing control sequences. This is usually set to "no" except when printing graphics files to a graphics device.

### **Time**

This parameter specifies the time the job should be printed. The job is held until the specified time, and is then processed by the despooler.

### **vfu**

This parameter sets an absolute top of form when spooling is "off" (available for "lpr" and "opts=" only).

## **DEFAULTS FILES**

The first group of parameters described above are assigned values in the defaults files, "/etc/class" and "/etc/defaults". Both of these files are maintained by the Printer Parameters Utility, described in Section 4 of this manual.

The defaults file contains defaults entries for the system and for each printer. Entries may be added for users, but are not required.

The class file entries are all optional and are not tied to the system, a printer or a user.

The files are used in this order:

1. If a class is specified when the job is submitted, all values specified by that class definition are assigned to the print job.
2. If a class is not specified or some parameters are not assigned values by the specified class, values are supplied from the "defaults" file in the order:
  - a. the user's entry, if any
  - b. the printer entry
  - c. the system entry

Values are taken from these entries until all parameters have been assigned values.

These default values can be overridden by specifying a value through the Submit Print Job Utility and "lpr" commands for new jobs, or the Maintain Print Queue Utility or "lpaint" command for already submitted jobs.

#### **PRINT FORM DEFINITIONS**

The form descriptions are contained in the file named "/etc/forms". Each print job must have some kind of form specified. A standard form definition, "stanrd", is supplied and serves as the default form.

These file entries define such parameters as form length and width, pitch, and number of copies. You may include defaults for every kind of form you use.

Form definitions are maintained by the Printer Parameters Utility. Refer to Section 4 of this manual for details on this utility.

## **Print Form Parameters**

The form definition parameters are as follows:

### **Slew 1 through Slew 8**

"Slewing" is a feature that allows you to control which line a printer skips to. The slew "channels" can be defined in the form file or in a Business BASIC program. Only Business BASIC can make use of the defined channels using the printer mnemonics. Refer to the Business BASIC Reference manual for additional information on using this feature.

Slew 1 is not optional; it is always defined as line 1, the top of the page.

The other seven (2 - 8) slew channels are optional. The entry must specify the line number you want to skip to. The maximum line numbers for different printers are as follows (names are as used in the Port Configuration Utility):

DMP	= 192
LQP	= 255
TP80	= 102
Tritel	= 132
Whisper	= 132
ISP	= 132 (emulated)

### **Lines Per Inch**

This parameter can be set to either 6 or 8 lines per inch. Some printers cannot use the lines per inch feature. If there is a switch on the printer for one or the other setting, it overrides this parameter.

The options for supported printers are:

DMP	= 6 or 8
LQP	= 6 or 8
TP80	= 6 or 8
Tritel	= not supported
Whisper	= 6 or 8
ISP	= not supported

### Pitch

This refers to the number of character per inch (horizontally). The list below show the available settings for different printers.

DMP	= 10
LQP	= 10 and 12
TP80	= 10 and 16
Tritel	= 10 and 16
Whisper	= 10 and 16
ISP	= not supported

### Translation Tables

The translation tables are rather complicated features to accommodate international users and special situations. Refer to the BOSS/IX Technical Reference Manual if you need to alter or set up translations tables.

## **SPOLER CONFIGURATION ISSUES**

The following paragraphs address several configuration issues involving spooling.

## Selecting Spooling

When a printer is set up on your system, it can be configured to use spooling or not to. This selection is made using the Port Configuration Utility, described in Section 4 of this manual.

In general, spooling should be turned "ON" for system printers to allow all users to have access to the printers.

If spooling is turned "OFF", you can still place a job in the spool queue. This can be done in two ways:

1. Specifying a print class which specifies spooling is "ON". This can be done using the Submit Print Job Utility, the "lpr" command "class=" option, or using the "CLASS=" option in Business BASIC.
2. By using the "lpr" command or using the "OPTS=" I/O option in Business BASIC. The switch that must be used is "-spool", e.g.:

```
@>lpr list=p7 -spool /etc/forms
```

The file is added to the spool queue, but since spooling is not active the file is not printed. In order for the job to be printed, the spooler must be started by either:

1. Change the configuration of the printer, turning spooling "ON", and then shut down and reboot the system. Use the Port Configuration Utility to change the printer configuration.

2. Start the despooler directly. The program is "/sys/lpd", and can be executed at the command interpreter prompt, followed by the name of the printer to be despoiled. E.g.:

```
@>/sys/lpd p7 &
```

The ampersand (&) is not necessary, but starts the program in background mode, thus allowing you to perform other tasks while the spooler operates. In this case, spooling will be turned off again when the process is killed or you log off.

### **Bypassing Spooling**

With spooling enabled, you can choose to bypass the spooler for specific print jobs. This can be done in three ways:

1. Specify a print class that specifies that spooling is "OFF". This can be done when using the Submit Print Job Utility, the "lpr" command "class=" option, or the "CLASS=" option in Business BASIC.
2. By using the "lpr" command or using the "OPTS=" I/O option in Business BASIC. The switch that must be used is "-off", e.g.:

```
@>lpr list=p7 -off /etc/forms
```

3. By using output redirection in the BOSS/IX command language (refer to Section 5, Command Language, in this manual).

In these cases, the system will attempt to send the print job directly to the spooled printer without adding the job to the spool queue. It will work only if the printer is not currently in use. If another job is being printed, the attempt to print the job will fail and a "device busy" message will be returned.

### **Automatic Form Feed**

Unless it is told otherwise, the BOSS/IX spooler sends a form feed to the printer before printing a file. This automatic form feed can be turned off on a printer-by-printer basis.

To suppress the automatic formfeed for a printer, you must create a file named "/etc/XX.noformfeed", where XX is the name of the printer (lp, pl, p2, etc.). This command line will create the file for "lp":

```
@>echo > /etc/lp.noformfeed
```

The next time the spooler is started, it will see this file and suppress the automatic form feed for that printer.

The automatic form feed can be begun for the printer by deleting the file and restarting the spooler.

To start the spooler again, either:

1. shut down and reboot the system, or
2. find the process ID for the despooler (use the "ps -a" command), and kill it. If the printer configuration has spooling turned "ON", the despooler will be restarted automatically.

## **BOSS/IX PRINT UTILITIES AND COMMANDS**

The BOSS/IX utilities and commands available for maintaining printing are listed below. Refer to Sections 4 and 5 of this manual for detailed information on using these utilities and commands.

<b>UTILITY</b>	<b>COMMAND</b>
Submit Print Job	lpr
Maintain Print Queue	lpmaint
Change Print Form	lpmaint -fc
Printer Status	lpq, lpstat

Note that the Maintain Print Queue Utility also allows you to modify print jobs in the queue, as does the "lpmaint" command. In other words, the "lpmaint" command incorporates the functions of both the Maintain Print Queue and Change Print Form Utilities.

## NOTES

---

## NOTES

## SECTION 4

### BOSS/IX UTILITIES

#### INTRODUCTION

The BOSS/IX Utilities are general purpose programs that perform routine system maintenance functions. Parameters are supplied to the utility via an interactive interface, making it easy for the user to control the execution of the program without needing to remember a command format.

To further increase their ease of use, the utility programs are integrated into the BOSS/IX menu system. The menu system groups utilities by related function. The organization makes it easy to select a utility to perform the desired function by making selections at two or three menus, starting at a central menu. The utilities may also be executed directly by program name, thus circumventing the menu system.

#### ORGANIZATION OF THIS SECTION

This section describes all of the BOSS/IX utility programs. Several standard procedures are used throughout the utility set, and are all described once under "Standard Procedures." The remaining subsections describe the individual utilities. These subsections are organized according to the grouping of the utilities provided by the menu system.

For each utility, the function is described followed by a description of the prompts and the information required. The descriptions are brief, making them useful for quick reference when required.

Following a functional overview, the main menus and prompts are described. The prompts that are omitted are either described under "Routine Procedures" or are obvious. Although in some cases the descriptions reflect the procedural flow of the utilities, this is not strictly maintained.

The contents of the remainder of this section are summarized below.

#### **Routine Procedures**

- Accessing Utilities**
- Utility Prompts**
- Special Function Keys**
- Pattern Matching Characters**
- File List Specification**
- Printer Specification**
- Specify Utility Environment**

#### **Directory Utilities**

- Display Directory**
- Create Directory**
- Delete Directory**
- Change Working Directory**
- Directory Security**

#### **File Utilities**

- Create File**
- Delete File**
- Display File Information**
- Rename File**
- Copy File**
- Move File**
- Change File Size**
- Filelist Maintenance**
- File Security**
- File Repair**

**Filesystem and Disk Utilities**

- Mount and Unmount Filesystem
- Format Diskette
- Copy Diskette
- Free Space Analysis
- Filesystem Error Analysis

**Printer Utilities**

- Submit Print Job
- Maintain Print Job
- Change Print Form
- Printer Status
- Printer Parameters

**Save and Restore Utilities**

- Save to Mag Tape
- Restore from Mag Tape
- Report Save Mag Tape Contents
- Compare Save Mag Tape
- Label Mag Tape
- Save to Diskette
- Restore from Diskette
- Report Save Diskette Contents

**BASIC Program Utilities**

- List and Cross-Reference
- Renumber BASIC Program
- Search and Replace
- Compare BASIC Programs
- Merge BASIC Programs
- Display BASIC Program Names
- Encrypt BASIC Program

**System Utilities**

- Set Date and Time
- Port Configuration
- Edit Menus
- Operator Information

## NOTES

## **STANDARD UTILITY PROCEDURES**

This chapter describes standard procedures that are used throughout the BOSS/IX Utilities set. Included are:

- o Accessing utilities
- o Utility prompts and menus
- o Special function keys
- o File Selection Procedure
- o Pattern Matching Characters

### **Accessing Utility Programs**

The individual utility programs can each be accessed through the menu system and executed from the command interpreter. In either case, an individual user's right to run a particular utility is controlled by login security. In addition, some functions of utilities can only be run by the system administrator.

#### **Menu Access**

The BOSS/IX Utilities can all be accessed through the menu system. The Main Utility Menu (ut.main) provides options for each group of utilities, plus options for the Business BASIC programmer utilities and optional communications and intersystem transfer programs:

1. Directories
2. Files
3. Filesystem & disk
4. Printers
5. Save & restore
6. BASIC programs
7. Communications
8. System
9. Intersystem transfer

(Although Communications and Intersystem transfer are included in the menu, they are optional packages, and are not discussed in this manual.)

When you select a group of utilities, another menu is displayed showing the utilities in that group. For example, the first option, Directories, leads to the Directory Utilities Menu (ut.dir):

1. Display
2. Create
3. Delete
4. Change working directory
5. Security

Enter the number of the desired utility to execute it.

The menu prompt also allows you to enter the name of a menu. The names of the menus used for utility access are shown in table 4-1.

Table 4-1. Menu Names for Utility Program Access

<u>Menu Title</u>	<u>Menu Name</u>
Initial Menu	top
Main Utility Menu	ut.main
Directory Utilities Menu	ut.dir
File Utilities Menu	ut.file
Filesystem & Disk Utilities Menu	ut.filesys
Printer Utilities Menu	ut.printer
Save and Restore Utilities	ut.save
System Utilities Menu	ut.system

#### Command Mode Access

#### Utility Programs

Users who have access to command mode can execute the utility programs directly as programs. No parameters are accepted on the command line other than the utility program name.

To execute a utility program, type the name of the desired utility immediately following the command interpreter prompt and press <RETURN>. For example, to execute the Directory Display Utility type:

```
@>ddisplay
```

and press <RETURN>. The command interpreter then executes the utility program. When the program is finished executing, the user is returned to command mode.

Table 4-2 lists utility program names.

Table 4-2. Utility Program Names

<u>Utility Title</u>	<u>Program</u>
<u>Name</u>	
<b>Directory Utilities</b>	
<b>Directory Display</b>	ddisplay
<b>Create Directory</b>	dcreate
<b>Delete Directory</b>	ddelete
<b>Change Working Directory</b>	workdir
<b>Change Directory Security</b>	dsecure
<b>File Utilities</b>	
<b>Create File</b>	fcreate
<b>Delete File</b>	fdelete
<b>File Information</b>	finfo
<b>Rename File</b>	frename
<b>Copy File</b>	fcopy
<b>Move File</b>	fmove
<b>Change File Size</b>	fchange
<b>Filelist Maintenance</b>	filelist
<b>Change File Security</b>	fsecure
<b>File Analysis and Repair</b>	frepair
<b>Filesystem and Disk Utilities</b>	
<b>Mount &amp; Unmount Filesystem</b>	smount
<b>Format Diskette</b>	sformat
<b>Copy Diskette</b>	copydisk
<b>Filesystem Free Space</b>	freespace
<b>Filesystem Error Analysis</b>	fsdbg
<b>Printer Utilities</b>	
<b>Submit Print Job</b>	psubmit
<b>Maintain Print Queue</b>	pqueue
<b>Change Print Form</b>	pformchg
<b>Printer Status</b>	pstatus
<b>Printer Parameters</b>	pparams

Table 4-2. Utility Program Names (cont'd)

<b>Save &amp; Restore Utilities</b>	
Save to Cartridge	msave
Restore from Cartridge	mrestore
Report Save Cartridge Contents	mlist
Compare Save Cartridge	mcompare
Label Cartridge	mlabel
Save to Diskette	dsave
Restore from Diskette	drestore
Report Save Diskette Contents	dsaverpt
 <b>BASIC Program Utilities</b>	
BASIC Program Cross-Reference	bxref
BASIC Program Renumber	brenumber
BASIC Program Search & Replace	bsearch
Compare BASIC Programs	bcompare
Merge BASIC Programs Utility	bmerge
Display BASIC Program Names	bdisplay
Encrypt BASIC Program	bencrypt
 <b>System Utilities</b>	
Date & Time	sdate
Port Configuration	configure
Menu Editor	menuedit
Operator Information	oprinfo

### Utility Menus

The menus listed in table 4-1 can also be executed from command mode. This is done by executing the menu driver program (menu) followed by the name of the desired menu. For example, to access the File Utilities Menu type:

```
@>menu ut.file
```

and press <RETURN>.

## **Utility Prompts**

Several different kinds of prompts are used throughout the utility set. The kinds of prompts that occur most commonly are described in the following paragraphs. Some prompts are in the form of menus. These menus are distinct from the menus generated by the menu driver.

Help text is available at each prompt in the utility programs providing immediate help with a prompt while you are using the system.

### **Multiple Choice Prompts**

The Multiple Choice Prompt is used frequently as a simple way to specify several parameters on a single screen. Current selections are indicated with an asterisk (\*). When the screen is first displayed, default options are indicated.

Figure 4-1 shows a sample multiple choice prompt.

**Figure 4-1. Sample Multiple Choice Prompt**

To change the selected option on any of the displayed parameter lists, press the number key of that list. DO NOT PRESS <RETURN>. Each time you press a number key, the asterisk in that list moves to the next option in the list.

When you have selected the desired parameters, press <RETURN>. The utility will proceed to the next step.

#### Menu Prompts

Menus are used in some utilities to allow you to choose one of several listed items. For example, the Printer Parameters Utility displays the following menu:

1. Maintain print control defaults
2. Maintain class definitions
3. Maintain form definitions

Type number of selection, <RETURN> to end:

As described in the displayed prompt, type the number corresponding to the desired option and then press <RETURN>. This completes the selection and the utility proceeds to the next step.

#### Toggle Prompts

The toggle prompt is similar to the multiple choice prompt except that each option in a list is numbered. Any number of items in the list may be selected. Currently selected items are marked with an asterisk (\*).

For example, the Change Directory Security Utility displays the following toggle prompt:

OWNER	OTHERS
* 1. read	* 4. read
* 2. write	* 5. write
* 3. execute	* 6. execute

Type number to change access parameters,  
<RETURN> to end:

In this example, all items are selected.

To deselect an item, press the number key corresponding to that item. The selection status of that item toggles from "ON" to "OFF" or from "OFF" to "ON", with the asterisks indicating the current status.

When you have finished selecting the menu items, press <RETURN>. The utility then proceeds to its next step.

### Input Prompts

Input prompts vary as widely as the information they request. Usually they ask for the name of a file or a directory, for example:

Type name of file, <RETURN> to end:

In response to the prompt, type in the requested information and press <RETURN>.

Remember that BOSS/IX is case sensitive in file names. A file name must be typed exactly as it exists in the file system.

Some prompts will accept pattern matching characters in file names. If the utility processes several files at a time, the prompt may return after you press <RETURN> following a file name to allow you to enter additional names.

Depending on the utility, pressing <RETURN> without any entered information has different effects. The effect is usually indicated in the prompt.

### Verification Prompts

A verification prompt asks you to verify that you really want the utility to continue with its procedure. This is done whenever the procedure may be destructive to data.

A typical verification prompt looks like this:

Do you want to delete this directory? (yes/no):

Check to make sure you have not made a mistake before replying.

If you do not want to continue with the procedure, type "N", "n", "NO" or "no" and then press <RETURN>. The utility stops the procedure immediately and either returns to an earlier step or terminates.

If everything is all right and you want to proceed with the procedure, type "Y", "y", "YES" or "yes" and then press <RETURN>. The utility then proceeds to its next step.

### **Special Function Keys**

The BOSS/IX Utilities recognize several special function keys available on standard MAI Basic Four terminals and assign them consistent functions. These keys and their standard functions are described in Table 4-3.

Table 4-3 Special Function Keys

<u>KEY</u>	<u>FUNCTION</u>
<RETURN>	<p>Signals the utility to accept the information typed at a prompt as input. If pressed without any information being entered, the program either:</p> <ol style="list-style-type: none"><li>1. Takes the default value, if a default is defined</li><li>2. Repeats the prompt if information is required and no default is defined</li><li>3. Ends a series of inputs and continues</li><li>4. Terminates the program.</li></ol> <p>The action is usually indicated in the prompt.</p>
<CTL-I>	Same as <RETURN>.
<CTL-II>	Returns the program to the immediately preceding prompt, allowing an earlier response to be changed. Responses to prompts remain unchanged, except for responses to verification prompts which must be entered again.
<CTL-III>	During the file selection procedure, accesses the environment menu.

Table 4-3. Special Function Keys (cont'd)

⟨CTL-IV⟩	Terminates the program immediately. No additional actions are taken by the program. The utility exits to the entry point, e.g., the menu system or command interpreter.
	Some utilities reject ⟨CTL-IV⟩ during processing. If data has been entered at an input prompt, ⟨CTL-IV⟩ is rejected.
⟨ESCAPE⟩	Terminates a utility during execution. At a prompt, ⟨ESCAPE⟩ is treated as ⟨CTL-IV⟩. Some utilities reject ⟨ESCAPE⟩ during processing.
⟨HELP⟩	Only on the EDT terminal, displays the help text for the current prompt. On all terminals the help key function is performed by ⟨CTRL⟩+⟨Z⟩.

#### Nonoperational Keys

A utility program might not use all of these special function keys, or might disable them during some part of its processing. In these cases, the program responds to pressing the key by sounding a single BEEP.

#### Function Key Equivalences

Table 4-4 describes CONTROL sequences that duplicate the keys described above. These can be used on terminals lacking the keys described.

**Table 4-4. Control Sequences for Special Function Keys**

<u>KEY</u>	<u>CONTROL SEQUENCE</u>
<RETURN>	<CTRL>+<M>
<CTL-I>	<CTRL>+<\>
<CTL-II>	<CTRL>+<]>
<CTL-III>	<CTRL>+<^>
<CTL-IV>	<CTRL>+<?>
<ESCAPE>	<CTRL>+<[>
<HELP>	<CTRL>+<Z>

### **File Selection Procedure**

Several utilities allow you to operate on a selected group of files. These files are usually specified by building a list of file names. Often this file selection process occurs if you choose the "selected files" option in a multiple choice prompt.

Note that the files selected are filtered according to the current file selection environment. This is described below.

The list of file names is initially empty, so the procedure begins by prompting for a file name to add:

Type a file or filelist name to add, <RETURN> to end:

You may specify either a file name or a filelist name. The file name may be either a full path name or a partial path name relative to the displayed working directory. A filelist name must include the ".f" extension but not the directory path. Type the name of the file or filelist and press <RETURN>.

Each file name you enter is added to a list below the prompt. For each filelist entered, all file names in the filelist are added to the list below the prompt.

After the file names are added to the list, the prompt is redisplayed.

You may press <CTL-III> at the prompt to access the file selection environment menu. Refer to the Change Environment description below.

When you are finished entering names to the list, press <RETURN> alone. The "Edit list of names" menu is then displayed:

**Edit list of names**

1. Add
2. Delete
3. Display
4. Sort
5. Change environment
6. Save

Type number of selection, <RETURN> to end:

**Add Function**

The Add option allows you to add more file names to the current list. The procedure is the same as described above.

### Delete Function

The Delete option allows you to delete file names from the current list. You may enter fully qualified or partially qualified file names or filelist names, as in the Add function.

The program checks the current list for each file name entered or found in the filelist and, if found, deletes that file name from the list. The prompt is then redisplayed.

You may press <CTL-III> at the prompt to access the file selection environment menu. Refer to the Change Environment description below.

Press <RETURN> alone when you are finished deleting file names. The "Edit list of names" menu is then redisplayed.

### Display Function

The Display option displays the file names in the current list. As many file names as can be displayed at one time are shown on the screen. For a long list, press <RETURN> to display the next screenful of names. When the last name has been displayed, press <RETURN> to return to the "Edit list of names" menu.

### Sort Function

The Sort option sorts the current list into alphabetic order by fully qualified pathname. When the list has been sorted the "edit list of names" menu is redisplayed.

## Change Environment

The Change environment option displays a menu of items controlling file selection. The items listed and their defaults differ for different utilities.

The environment menu can be accessed from the file name prompt of the Add and Delete functions allowing you to change file selection characteristics during the file specification procedure.

The options are described in the following paragraphs.

### Working Directory

The Working directory option allows you to change the current working directory. When you select this option, you are prompted for the new working directory. Specify the new directory either by a fully qualified path name or a partially qualified path-name relative to the currently displayed working directory.

### Starting Date/Ending Date

These options allow you to change the "Date modified range" for file selection. Only files that have been last modified on a date and time falling within this range are selected.

For the starting date and ending date, you are prompted to enter the new date in this format:

mm/dd/yy hh:mm xx

The "xx" is to be filled with "am" or "pm", the others are two-digit fields for month, day, year, hour and minute.

The default range is intended to include all files.

### Expansion

This option turns on or off the expansion of file names as entered. If the option is set to "yes", the working directory is prefixed to a partial path name and pattern matching characters are interpreted. If it is set to "no", the file name is added to the list as entered, without adding the working directory to partial path names or interpreting pattern matching characters.

Most utilities using the file selection routine do not include this environment option, and by default expand partial path names with the working directory and interpret pattern matching characters.

### Sub-directories

The Sub-directories option determines whether a specified directory will be searched recursively, that is whether or not files in sub-directories will also be included in the list of files being created.

Selecting this menu item toggles the selection between "yes", including files in sub-directories, and "no", including only files in the immediate directory.

## File Types

The File types option lists the file types recognized by the BOSS/IX operating system. This is a toggle prompt, as described earlier, with an asterisk marking each type of file to be included in the list being constructed. To change the file type selections, enter the number of the file type to be included or excluded. The default includes all file types.

## Save Function

The Save option allows you to save the list of files created into a filelist, making it available for later use. This option is not included in the Filelist Maintenance Utility menu since any filelist made using this utility saves the filelist by default.

When you select this option you are prompted to supply a filelist name. The rules for a filelist name are:

1. Must not include a directory path
2. Must have the ".f" extension
3. May be up to 10 characters long, including the extension

All filelists are stored in the "/util/f1" directory.

If a filelist with the specified name already exists, you are notified and asked whether or not to replace the current list:

The filelist already exists. Replace it?  
(yes/no):

If you answer "NO", you are asked for another name.

When the filelist has been saved, the "Edit list of names" menu is redisplayed.

### End File Selection

When you are finished with the file selection options, press <RETURN> alone at the "Edit list of names menu". The utility that called the procedure will then continue its procedure.

### **Pattern Matching Characters**

When you are specifying a file or filelist name you may make use of special pattern matching characters recognized by BOSS/IX. A full description of the pattern matching characters is contained in the Command Language section of this manual. Table 4-5 summarizes the pattern matching characters that can be used while specifying files.

If a pattern contains spaces, enclose the entire pattern in double quotation marks (" ") to assure correct interpretation of the pattern.

**Table 4-5. Pattern Matching Characters**

<u>Symbol</u>	<u>Pattern Matching Function</u>
*	Matches any number of characters
?	Matches any single character
#X	(Pound-sign followed by a character) Matches any number of occurrences of the specified character

Table 4-5. Pattern Matching Characters (cont'd)

- ! Specifies alternative matches
- {..} Groups patterns in order of precedence
- [x-y] Specifies a range of characters in ASCII order, where x is the first character in the range and y is the last character
- [^P] Where P is a pattern, matches anything that does not match P

## NOTES

## **DIRECTORY UTILITIES**

As described in Section 2, directories serve as branching points in the hierarchical file system structure. The Directory Utilities perform the functions necessary to maintain an orderly system of directories.

The Directory Utilities are:

- o **Directory Display Utility**, which reports the contents of a directory
- o **Create Directory Utility**, which creates a new directory
- o **Delete Directory Utility**, which deletes an empty directory
- o **Change Working Directory Utility**, which changes your current working directory
- o **Change Directory Security Utility**, which changes the read/write/execute access modes of a directory

## **Directory Display Utility**

### Program Name

/util/ddisplay

### Related Commands

/bin/ls

### Utility Description

The Directory Display Utility produces a report of the contents of the user's working directory. The report is recursive, reporting files in all sub-directories in addition to files in the specified directory.

Options provided by the utility are:

- o Change working directory
- o Selected file report
- o Include file attributes
- o Report to the terminal, a file or a printer

### Prompts

#### **Multiple Choice Prompt**

The utility begins with a multiple choice prompt. The current working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Directory

The Directory option determines whether the current working directory is to be displayed or another directory is to be displayed. If you select "other", you will be prompted for the new working directory when you leave the multiple choice prompt.

## File Selection

The File Selection option determines whether all files are to be reported or only selected files are to be reported. If you select "selected files", the file selection process (described under "Standard Procedures") is invoked to specify files when you leave the multiple choice prompt.

## Report Detail

The Report Detail option specifies the amount of detail in the report. The report can include file names only or file names and file attributes. The file attributes are:

- o File type
- o Key size
- o Record size
- o Maximum number of records
- o Number of records used
- o Number of blocks of disk space used

Data is given depending on the file type.

## Report Device

The Report Device option determines whether the report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "printer," you will be asked to select a printer from a menu of printer names when you leave the multiple choice prompt. If you select "file" you will be prompted for a file name when you leave the multiple choice prompt.

When all selections are made, press <RETURN>. The report is then produced. When the report is completed, the utility terminates.

## **Create Directory Utility**

### Program Name

/util/dcreate

### Related Commands

/bin/mkdir

### Utility Description

The Create Directory Utility makes a new directory in the file system structure. If a path is given including names of several directories that do not exist, all directories in the path are created.

### Prompts

Type working directory, <RETURN> for no change:

To change the working directory, enter either the fully qualified path name or the partially qualified path name of the directory. To leave the working directory as is, press <RETURN> alone.

Type name of directory to create, <RETURN> to end:

Enter the fully qualified path name or the partially qualified path name of the directory to create. The program creates any directory in the path that does not exist. In this way a whole chain of directories can be created at once.

When an acceptable directory name is given, the utility creates the directory, briefly displays the message "DIRECTORY CREATED", and repeats the previous prompt.

When you are finished creating directories, press <RETURN> alone at this prompt to exit the utility.

#### Error Messages

"file" is not a directory. This operation requires a directory.

The name entered is the name of an existing file.

## **Delete Directory Utility**

### Program Name

/util/ddelete

### Related Commands

/bin/delete

### Utility Description

The delete Directory Utility deletes directories from the file system. Only empty directories can be deleted by this utility. You must have "write" access to a directory to delete it.

### Prompts

Type working directory, <RETURN> for no change:

For convenience, you may change your working directory at the start of the utility. Type the full pathname or the partial pathname for the new directory.

Type name of directory to delete, <RETURN> to end:

Type the full or partial pathname of the directory to delete. When you are finished deleting directories, press <RETURN> to end the utility program.

### Error Messages

Cannot delete directory "directory". It is not empty.

The directory specified contains files or sub-directories. This utility can only delete empty directories.

File/directory "directory" is locked.

The directory is in use or you do not have "write" permission to the directory.

"directory" is not a directory. This operation requires a directory.

The specified file is not a directory.

## **Change Working Directory Utility**

### Program Name

/util/workdir

### Related Commands

cd	(change directory)
/bin/pwd	(print working directory)

### Utility Description

The Change Working Directory Utility changes your working directory.

The utility is effective only if you accessed the utility through the menu system. If you execute the program directly from the command interpreter, the change is reversed as soon as the program ends and you are returned to command mode.

### Prompts

Type working directory, <RETURN> for no change:

Your current working directory is displayed above the prompt. To change your working directory, type the full or partial path name of the new directory and press <RETURN>.

Press <RETURN> alone to leave the working directory unchanged.

If you entered a proper directory name, the program displays a message indicating that your working directory has been changed. Press <RETURN> to exit the utility.

The parent directory of the current working directory can be referred to as "...". For example, a change from "/usr/frank" to "/usr/fred" can be made by specifying "../fred" in response to the prompt.

### Errors

File/directory "directory" does not exist.

The specified directory does not exist. Check the directory path name.

"directory" is not a directory. This operation requires a directory.

The specified name is the name of a file, not a directory.

## Change Directory Security Utility

### Program Name

/util/dsecure

### Related Commands

/bin/filemodes	(change file access modes)
/bin/chown	(change owner)

### Utility Description

The Change Directory Security Utility changes the owner and read/write/execute access of a directory. Only the system administrator can change the owner of a directory. Only the system administrator and file owner can change the access modes. For other users, the owner name and access modes are displayed without the option of changing them.

### Prompts

Type working directory, <RETURN> for no change:

The first prompt asks for the working directory. To change your working directory, type the full or partial path name of the new directory and press <RETURN>.

Type directory name, <RETURN> to end:

Type the full or partial path name of the directory to process and press <RETURN>. Press <RETURN> alone to end the utility.

The remaining prompts differ for the system administrator, the directory owner, and all other users.

### System Administrator

The system administrator is shown a menu with two options, Directory owner and Directory access parameters. The current owner and access parameters are displayed with the menu. Enter the number of the item to change.

When parameters are set, press <RETURN>. The "Type directory name" prompt is then repeated.

### Directory Owner

If you select to change the directory owner, you are prompted for the name of the new owner. The owner must be a defined user. Type the owner's name and press <RETURN>.

### Directory Access Modes

If you select to change the directory access modes, a toggle menu is displayed including read, write and execute access for both the directory owner and others. Press the number of the access mode to change it. When all modes are set, press <RETURN>.

### Directory Owner

The owner of a directory is shown a toggle menu including read, write and execute access modes for both the owner and others. Press the number of the access mode to change it. When all modes are set, press <RETURN>.

### Others

Users other than the system administrator and the directory owner are shown the access parameters for the directory. After viewing, press <RETURN>.

## NOTES

## FILE UTILITIES

The File Utilities perform a number of functions necessary to the maintenance and manipulation of files in the file system. Although directories are actually a kind of file, they are not handled by this group of utilities but by the Directory Utilities.

The file utilities are:

- o **Create File Utility**, which creates files of string, direct, sort, indexed, serial, and BASIC program
- o **Delete File Utility**, which deletes files from a directory
- o **File Information Utility**, which reports file information parameters
- o **Rename File Utility**, which renames and optionally moves a file
- o **Copy File Utility**, which copies files between directories without erasing them from the original directories
- o **Move File Utility**, which moves files between directories, erasing them from the original directories
- o **Change File Size Utility**, which increases or decreases the file size parameters for a file
- o **Filelist Maintenance Utility**, which creates, deletes, edits and displays filelists

- o **Change File Security Utility**, which changes the access mode parameters for files
- o **File Analysis and Repair Utility**, which analyzes and optionally repairs structural errors in files

## **Create File Utility**

### Program Name

/util/fcreate

### Related Commands

None

### Utility Description

The Create File Utility creates files of type Direct, Indexed, Serial, Sort, String and BASIC Program. A few special operating system files, such as device and event count files, are not created by this utility.

The maximum size of a file is assigned when it is created, but disk space is allocated to the file by the operating system only as needed and defined by the initial and growth extents.

The initial extent determines the amount of disk space allocated when the file is first written to. The growth extent determines the amount of disk space allocated when the currently allocated space is exhausted, up to the maximum file size.

### Prompts

Type working directory, <RETURN> for no change:

The current working directory is displayed above the prompt. To change the working directory, type the full or partial path name of the directory and press <RETURN>.

Type file name, <RETURN> to end:

Type the full or partial path name of the file to be created. When you are finished creating files, press <RETURN> alone at this prompt to end the program.

#### File Type Menu

The six file types that this utility creates are shown in a menu. Enter the number of the desired file type.

Type key size (2-56):

For direct and sort files you must specify the key size. This is dependent upon the requirements of the application using the file, but must be within the range shown in the prompt.

Type record size (1-32767):

For direct, indexed and serial files you must specify the record size. This is dependent upon the requirements of the application using the file but must be within the range shown in the prompt.

Type number of records (1-8388607):

For indexed, serial, direct and sort files you must specify the maximum number of records in the file. The number entered must be within the range shown in the prompt.

Type file size (1-63 blocks):

For BASIC Program files you must specify the maximum file size in blocks (1 block = 512 bytes).

Type initial extent size (1-n [records,keys,blocks]):

The units (records, keys or blocks) in which the initial extent is requested varies with the file type. Enter a number within the indicated range for the initial extent. The default is the maximum shown in the prompt.

Type growth extent size (1-n [records,keys,blocks]):

The units (records, keys or blocks) in which the growth extent is requested varies with the file type. Enter a number within the indicated range for the initial extent. The default is "1".

Are the parameters correct?

The specified parameters are displayed in a menu. The parameters may be changed by entering the number of the parameter to change and responding to the prompt. When the parameters are all correct, press <RETURN> to define the file as specified.

A message is briefly displayed indicating that the file has been created, and then the "Type file name" prompt is repeated.

## **Delete File Utility**

### Program Name

/util/fdelete

### Related Commands

/bin/delete

### Utility Description

The Delete File Utility deletes one or more files or links to files.

Links are created by the "/bin/addname" command. A file is only deleted from the file system when all links to the file have been deleted.

If a file is open when this utility attempts to delete it, the file (or link) will be deleted when the file is closed.

When all specified files have been deleted, the program ends.

### Prompts

#### Multiple Choice Menu

The utility begins with a multiple choice prompt. Your working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will delete a single file or multiple, selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Verify

The Verify option determines whether you will be asked to verify (yes or no) to delete each file. If you select "yes" in the menu, a verification prompt will be displayed with the name of the file about to be deleted prior to its deletion.

## Report Device

The Report Device option determines whether the report of files deleted is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

## **Display File Information Utility**

### Program Name

/util/finfo

### Related Commands

/bin/ls (with options)

### Utility Description

The Display File Information Utility creates a report of file parameters including:

- o file name
- o file type
- o file size
- o file owner
- o file access parameters
- o date last modified
- o number of names (links)

### Prompts

### Multiple Choice Menu

The utility begins with a multiple choice menu. The current working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Directory

The Directory option determines whether the current working directory is to be used with partial path names or another directory is to be used. If you select "other" you will be prompted for the new working directory after you exit this multiple choice menu.

## File Selection

The File Selection option determines whether information is to be reported on a single file or on selected files. If you select "single file" you will be prompted for the name of the file to report after you exit this menu. If you select "selected files" you will be asked to build a list of files using the file selection procedure described under "Standard Procedures."

## Report Device

The Report Device option determines whether the report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name after exiting this multiple choice menu. If you select "printer", you will be asked to select the printer from a menu of printers after you exit the menu.

## **Rename File Utility**

### Program Name

/util/frename

### Related Commands

/bin/move

### Utility Description

The Rename File Utility changes the name of a file. The file may be moved at the same time by specifying the new directory path with the new name.

### Prompts

Type working directory, <RETURN> for no change:

The current working directory is displayed above this prompt. To change the working directory, enter the full or partial path name for the new working directory.

Type name of file to rename, <RETURN> to end:

Type the full or partial path name of the file to be renamed and press <RETURN>.

When you are finished renaming files, press <RETURN> alone to end the utility.

Type new file name:

Type the new name of the file, using the full or partial path name. If you specify a path name different from that used in the previous prompt, the file will be moved to the new directory as well as renamed.

A message is briefly displayed indicating that the file has been renamed, and the "Type name of file to rename" prompt is repeated.

## **Copy File Utility**

### Program Name

/util/fcopy

### Related Commands

/bin/copy	(creates a copy of a file)
/bin/addname	(creates a link to a file)

### Utility Description

The Copy File Utility creates a physical copy of a file on disk (or diskette). This is different from creating a link to a file (/bin/addname) which gives an additional name to a single copy of a file.

Files may be renamed as they are copied. The original copy of the file is unchanged by this utility.

### Prompts

#### **Multiple Choice Menu**

The utility begins with a multiple choice menu. The current working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The File Selection option determines whether a single file or selected files will be copied. If you select "single file" you will be prompted for the file name after you exit this multiple choice menu. If you select "selected files" you will be asked to build a list of files using the method described in "Standard Procedures after you leave this menu.

## Duplicates

A duplicate file condition occurs when a file already exists in the target directory with the same name as a file being copied into that directory.

The Duplicates option determines if the file in the target directory should be replaced automatically by the file being copied or if the user is to be asked to decide for each occurrence.

## Rename Files

The Rename Files option determines whether or not the user will be allowed rename files as they are copied. If you select "yes," you will be prompted for the new name before each file is copied.

## Report Device

The Report Device option determines whether the report of files copied is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name after exiting this multiple choice menu. If you select "printer", you will be asked to select the printer from a menu of printers after you exit this menu.

**File "filename" already exists. Replace it?**

**(yes/no):**

If you specified "ask user" for the Duplicate files option, this prompt is displayed each time a duplicate file condition occurs. Answer "yes" to replace the file, "no" to skip the file.

## **Move File Utility**

### Program Name

/util/fmove

### Related Commands

/bin/move

### Utility Description

The Move File Utility creates a physical copy of a file on disk (or diskette) and then deletes the original copy. Files may be renamed as they are moved.

### Prompts

#### Multiple Choice Menu

The utility begins with a multiple choice menu. The current working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

#### File Selection

The File Selection option determines whether a single file or selected files will be moved. If you select "single file" you will be prompted for the file name after you exit this multiple choice menu. If you select "selected files" you will be asked to build a list of files using the method described in "Standard Procedures" after you leave this menu.

## Duplicates

A duplicate file condition occurs when a file already exists in the target directory with the same name as a file being copied into that directory.

The Duplicates option determines if the file in the target directory should be replaced automatically by the file being copied or if the user is to be asked to decide for each occurrence.

## Rename Files

The Rename Files option determines whether or not the user will be allowed rename files as they are moved. If you select "yes," you will be prompted for the new name before each file is moved.

## Report Device

The Report Device option determines whether the report of files moved is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name after exiting this multiple choice menu. If you select "printer", you will be asked to select the printer from a menu of printers after you exit this menu.

File "filename" already exists. Replace it?  
(yes/no):

If you specified "ask user" for the Duplicate files option, this prompt is displayed each time a duplicate file condition occurs. Answer "yes" to replace the file, "no" to skip the file.

## Change File Size Utility

### Program Name

/util/fchange

### Related Commands

None

### Utility Description

The Change File Size Utility expands and contracts the size of a single file. The utility cannot contract the size of a file below the minimum size required for the data currently in the file. The file types for which changes can be made, and the allowable changes are summarized in table 4-6.

Table 4-6. Permitted File Size Parameter Changes

<u>File Type</u>	<u>Increase</u>	<u>Decrease</u>
Indexed	Record Size # Records Initial Ext Growth Ext	---
Serial	Record Size # Records Initial Ext Growth Ext	---
Direct	Key Size* Record Size # Records Initial Ext Growth Ext	---
		# Records Initial Ext Growth Ext

Table 4-1. Permitted File Size Parameter Changes  
(cont'd)

Sort	Key Size # Keys Initial Ext Growth Ext	---	# Keys Initial Ext Growth Ext
BASIC Program	Program Size Initial Ext Growth Ext	Program Size	Initial Ext Growth Ext

The key size of a direct file cannot be changed if the file has more than one key set or if the key is defined in the data record.

#### Prompts

Type working directory, <RETURN> for no change:

The current working directory is displayed above the prompt. To change the working directory, type the full or partial path name of the new directory and press <RETURN>.

Type file name, <RETURN> to end:

Type the full or partial path name of the file to view or modify and press <RETURN>. When you are finished processing files, press <RETURN> alone to end the utility.

#### File Size Parameters Menu

The current size parameters for the specified file are displayed, and the parameters that are available to be changed are numbered. Type the number of the item to change and press <RETURN>.

When you select a file size parameter to change, you are prompted for the new size. The allowable sizes are indicated in the prompt, e.g.:

Type maximum number of records (1-8388607),  
<RETURN> for no change:

When you have finished changing parameters, press <RETURN> alone at the menu prompt to return to the file name prompt.

## **Filelist Maintenance Utility**

### Program Name

/util/filelist

### Related Commands

/bin/ved text editor to create/edit file  
/bin/delete delete file  
/bin/p display file

### Utility Description

The Filelist Maintenance Utility creates, modifies, deletes and displays the contents of filelists.

A filelist is a special file that contains a list of file names. The file names can be fully qualified or partially qualified path names, or absolute file names without any path information. File names stored with partial or no path information are expanded by the program using the filelist.

All filelists are stored in the directory "/util/f1". Filelist names must have the ".f" extension.

In addition to this utility, filelists are created by the file selection routine used by several utilities when the SAVE option is selected (refer to "Standard Procedures"). Filelists can also be created and edited by the BOSS/IX text editor, ved.

## Prompts

### Menu of Maintenance Functions

The opening menu for this utility allows you to Create a new filelist, Modify an existing filelist, Delete a filelist or Display the contents of a filelist. The remaining prompts are described according to these four functions.

To exit the utility, press <RETURN> alone at this menu.

#### Create Filelist Function

Type filelist name ending with ".f", <RETURN> to end:

Type the name for the filelist to create and press <RETURN>. The filelist name must end with the ".f" extension and may be up to 10 characters long including the extension.

When you are finished creating filelists, press <RETURN> alone to return to the menu of maintenance functions.

Type a file or filelist name to add, <RETURN> when done:

If the filelist name you entered is new, you are prompted to start adding names to the list. For a description of the rest of the procedure, refer to the description of the File Selection procedure under "Standard Procedures."

Note that the default environment setting for this utility does not expand file path names, pattern matching characters or filelist names, but adds names exactly as entered. This environmental default can be changed as described under "Standard Procedures."

The filelist already exists. Do you want to modify it? (yes/no):

If the filelist name entered at the previous prompt is the name of an already existing filelist, this prompt notifies you that it exists and allows you to modify it or enter another name.

To modify the existing filelist, enter "yes". The Edit list of names menu is then displayed. For a description of the rest of the procedure, refer to the description of the File Selection procedure under "Standard Procedures."

#### Modify Filelist Function

Type filelist name, <RETURN> to end:

Type the name of an existing filelist to modify and press <RETURN>. The filelist name must end with the ".f" extension and may be up to 10 characters long including the extension.

When you are finished modifying filelists, press <RETURN> alone to return to the menu of maintenance functions.

## Edit List of Names Menu

If the filelist name entered is found, the Edit list of names menu is displayed, unless the list is empty. If the list is empty, you are immediately prompted for names to add. For a description of the rest of the procedure, refer to the description of the File Selection procedure above under "Standard Procedures."

The filelist does not exist. Do you want to create it? (yes/no):

If the filelist name entered is not found, you are asked if you want to create it. If you answer "yes", the utility switches to the create function and you are prompted for a name to add. For a description of the rest of the procedure, refer to the description of the File Selection procedure under "Standard Procedures."

If you answer "no", the utility repeats the Type filelist name prompt.

## Delete Filelist Function

Type filelist name, <RETURN> to end:

Type the name of a filelist to delete and press <RETURN>.

When you are finished deleting filelists, press <RETURN> alone to return to the menu of maintenance functions.

Delete the filelist? (yes/no):

Answer "yes" to delete the specified filelist, "no" to keep the filelist. The Type filelist name prompt is repeated after you press <RETURN>.

### Display Filelist Function

#### Multiple Choice Menu

The Display Filelist function begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

#### Range

The Range options determines whether all filelists will be included in the report or only selected filelists. If you specify "selected", you will be prompted for filelists one at a time after you exit this multiple choice menu.

## Detail

The detail option determines whether only filelist names will be displayed, or the filelist names together with their contents.

## Report Device

The Report Device option determines whether the report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name after exiting this multiple choice menu. If you select "printer", you will be asked to select the printer from a menu of printers after you exit this menu.

Type filelist name, <RETURN> to continue:

Type the name of a filelist to report and press <RETURN>. When you are finished reporting filelists, press <RETURN> alone to return to the maintenance functions menu.

## Change File Security Utility

### Program Name

/util/fsecure

### Related Commands

/bin/filemodes

### Utility Description

The Change File Security Utility changes the owner and read/write/execute access to a file. Only the system administrator can change the owner of a file. Only the system administrator and file owner can change the access modes. For other users, the owner name and access modes are displayed without the option of changing them.

### Prompts

Type working directory, <RETURN> for no change:

The current working directory is displayed above the prompt. To change your working directory, type the full or partial path name of the new directory and press <RETURN>.

Type file name, <RETURN> to end:

Type the full or partial path name of the file to process and press <RETURN>. Press <RETURN> alone to end the utility.

The remaining prompts differ for the system administrator, the file owner, and all other users.

## System Administrator

The system administrator is shown a menu with two options, File owner and File access parameters. The current owner and access parameters are displayed with the menu. Enter the number of the item to change.

When parameters are set, press <RETURN>. The "Type file name" prompt is then repeated.

### **File Owner**

If you select to change the file owner, you are prompted for the name of the new owner. The owner must be a defined user. Type the owner's name and press <RETURN>.

### **File Access Modes**

If you select to change the file access modes, a toggle menu is displayed including read, write and execute access for both the file owner and others. Press the number of the access mode to change it. When all modes are set, press <RETURN>.

### File Owner

The owner of a file is shown a toggle menu including read, write and execute access modes for both the owner and others. Press the number of the access mode to change it. When all modes are set, press <RETURN>.

### Others

Users other than the system administrator and the file owner are shown the access parameters for the file. After viewing, press <RETURN>.

## **File Analysis and Repair Utility**

### Program Name

/util/frepair

### Related Commands

/bin/kychk

### Utility Description

The File Analysis and Repair Utility analyzes and optionally repairs Keyed (Direct and Sort), BASIC Program, Serial, Indexed and String type files, and generates a report of the final condition of the file. Only the system administrator is allowed to run the repair option.

If the repair option is being used, only one user should be logged on.

The repair option may require free disk space equal to twice the disk size of the damaged file. Use the Filesystem Free Space Utility to check for sufficient free space before repairing large files.

### Prompts

#### **Multiple Choice Menu**

The utility begins with a multiple choice menu. The current working directory is displayed at the top of the screen. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Repair Mode

The Repair Mode option determines whether the files will be analyzed only or analyzed and repaired. Although any user can set the option to "analyze and repair," the repair will be performed only for the system administrator.

## File Selection

The File Selection option determines whether the analysis is to be run on a single file or on selected files. If you select "single file" you will be prompted for the name of the file to analyze after you exit this menu. If you select "selected files" you will be asked to build a list of files using the file selection procedure described under "Standard Procedures."

## Report Device

The Report Device option determines whether the file analysis report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name after exiting this multiple choice menu. If you select "printer", you will be asked to select the printer from a menu of printers after you exit the menu.

**WARNING: Files or data might be lost. Proceed?  
(yes/no):**

If you select the "analyze & repair" option, this prompt asks for verification. You should have made backup copies of the files before attempting to repair them.

#### **Error Messages**

**Only the system administrator can repair a file.**

This message is displayed if a user other than the system administrator attempts to repair a file. The number shown is used by the program for internal purposes. Press <RETURN> to acknowledge the message and continue.

## NOTES

## NOTES

## **FILESYSTEM AND DISK UTILITIES**

This chapter describes the Filesystem and Disk Utilities. These utilities contain basic procedures required for the use and maintenance of the BOSS/IX file system.

As described in Section 2, the BOSS/IX file system is organized into structures called filesystems. Filesystems are created on floppy diskettes and on fixed disk partitions. One filesystem is singled out during the system boot as the "root" filesystem (for a normal boot, this is the filesystem on the root partition, "/dev/root", on the first hard disk). Other filesystems must be "mounted" to this filesystem before files in them can be accessed.

Floppy Diskettes require procedures not applicable to fixed disk partitions, namely formatting and copying. Utilities to perform these functions are also included in this group of utilities.

The Filesystem and Disk Utilities are:

- o **Mount & Unmount Filesystem Utility**, which attaches filesystems to the root filesystem, enabling access to the files they contain, detaches filesystems from the root filesystem, and displays currently mounted filesystems
- o **Format Diskette Utility**, which formats and creates a filesystem on floppy diskettes
- o **Copy Diskette Utility**, which makes an image copy of one diskette on another diskette

- o **Filesystem Free Space Utility**, which reports the amount of disk space available in a filesystem
- o **Filesystem Error Analysis Utility**, which analyzes and optionally repairs filesystem errors

## **Mount and Unmount Filesystem Utility**

### Program Name

/util/smount

### Related Commands

/bin/mount  
/bin/umount

### Utility Description

The Mount and Unmount Filesystem Utility mounts filesystems to the root filesystem (or an already mounted filesystem), unmounts filesystems and produces a report of currently mounted filesystems.

### Prompts

#### **Menu of Utility Functions**

The opening menu for this utility allows you to choose to Mount a filesystem, Unmount a filesystem or Display mounted filesystems. The remaining prompts are described according to these three functions.

To exit the utility, press <RETURN> alone at this menu.

## Mount a Filesystem Function

Type directory name, <RETURN> to end:

The current working directory is displayed above the prompt. The prompt is asking for the directory to which the filesystem is to be mounted. Files and directories in the mounted filesystem will be accessed through this directory. Since files in this directory are inaccessible while a filesystem is mounted to it, an empty directory should be used.

Type the full or partial path name of the mount point directory and press <RETURN>.

When you are finished mounting directories, press <RETURN> alone at this prompt to return to the functions menu.

## Menu of Filesystem Devices

A menu of filesystem devices is displayed. The menu includes every buffered disk and disquette device file in the "/dev" directory, and are not all mountable filesystems. An error message will be displayed if an unmountable device is selected.

Select the option containing the desired filesystem and press <RETURN>.

When the filesystem is successfully mounted, a message is displayed showing the filesystem device and the mount point directory. Press <RETURN> to acknowledge the message and return to the directory name prompt.

### Unmount a Filesystem Function

Type directory name, <RETURN> to end:

The current working directory is displayed above the prompt. Type the full or partial pathname of the mount point directory of the filesystem to be unmounted and press <RETURN>.

When the filesystem has been unmounted, a message is displayed indicating the directory that was unmounted. Press <RETURN> to acknowledge the message and return to the directory name prompt.

When you are finished unmounting directories, press <RETURN> alone to return to the functions menu.

### Display Mounted Filesystems Function

The Display Mounted Filesystems function requires no interaction. When this function is selected, a list of directories and the filesystem devices mounted on them is displayed on the terminal screen. When you are finished viewing the display, press <RETURN> to end the utility or, in the case of a very long display, to display the next screen.

After the last screen is displayed, the utility returns to the functions menu.

## **Format Diskette Utility**

### Program Name

/util/sformat

### Related Commands

/bin/devfmt format diskette  
/bin/makefs create filesystem

### Utility Description

The Format Diskette Utility formats a blank floppy diskette and creates an empty filesystem on the diskette. When the utility is finished, the floppy can be mounted to the root filesystem to receive files or used in the Backup to Diskette Utility.

A "verify" option checks the diskette for bad blocks, defects that may cause loss of data if the diskette is used. If any bad blocks are found during this procedure, the filesystem will not be created and the diskette should not be used.

### Prompts

Insert a diskette into drive.

WARNING: Format will destroy all data on the diskette.

Type <RETURN> to proceed:

Formatting fully initializes the diskette, thus destroying any data on it. Insert the diskette to be formatted into the floppy drive and press <RETURN>.

Verify the format? (yes/no):

The verification process checks the diskette for defects in the diskette surface that could cause loss of data. If any blocks are found to be defective, the bad blocks are listed and no filesystem will be created.

While the diskette is being formatted, the program displays:

Format in progress.  
PLEASE WAIT

If the formatting is successful, and no bad blocks found during the verification pass, the program then displays:

Creating Filesystem  
PLEASE WAIT

When the procedure is complete, the following prompt is displayed:

Format complete with no errors.  
Remove the diskette. Then type <RETURN> to proceed:

Remove the diskette and press <RETURN>.

The program then repeats the "Insert a diskette" prompt.

## **Copy Diskette Utility**

### Program Name

/util/copydisk

### Related Commands

None

### Utility Description

The Copy Diskette Utility makes an exact image copy of a diskette on another diskette. The utility uses only one diskette drive for the procedure. Data is first copied from the source diskette to the fixed disk, and then from the fixed disk to the destination diskette. The utility requires that there be more than 100 blocks of free space on the fixed disk in order to run (use the Free Space Analysis Utility to check the amount of free space).

When the utility is executed, it displays a line indicating the number of times the source and destination diskettes will need to be switched during the procedure. This is usually only 1 time, but is more if there is limited space on the fixed disk.

Note that the destination floppy must be formatted prior to running this utility.

### Prompts

Insert source diskette. Type <RETURN> to proceed:

Insert the diskette to be copied and press <RETURN>.

**Insert destination diskette.** Type <RETURN> to proceed:

Remove the source diskette from the drive and insert a formatted destination diskette, then press <RETURN>.

**WARNING:** All information on the destination diskette will be lost.

Proceed with copy? (yes/no):

This verification prompt is a chance to check and change before proceeding. A NO response repeats the "Insert destination diskette" prompt. A YES response causes the procedure to proceed.

Copy another diskette?

1. Yes, recopy same diskette
2. Yes, copy another diskette
3. No

Type number of selection, <RETURN> to end:

When the copy is complete, this menu is displayed. Type the number of the desired option.

Option 1 returns to the "Insert destination diskette" prompt.

Option 2 returns to the "Insert source diskette" prompt.

Option 3 or <RETURN> exits the utility.

## Notes

1. Once the destination diskette has been inserted and verified, the program compares the first 8 blocks of the destination and source diskettes. If they are identical, this prompt is displayed:

**WARNING: First 8 blocks of source and destination are identical.**

**Proceed with copy? (yes/no):**

Check to make sure that the diskettes were switched so you are not copying over the source diskette. Then answer the prompt.

A NO response repeats the "Insert destination diskette" prompt. A YES response causes the procedure to proceed.

2. If you have to switch diskettes more than once during the procedure and insert the wrong diskette during a switch after the first, one of these messages will be displayed:

**This diskette is not the original source diskette.**

**Type <RETURN> to proceed:**

**This diskette is not the original destination diskette.**

**Type <RETURN> to proceed:**

## **Free Space Analysis Utility**

### Program Name

/util/freespace

### Related Commands

/bin/space

### Utility Description

The Free Space Analysis Utility produces a report of the number of free blocks and file descriptors available on a filesystem device. The report can be either a "summary" or "full" report, and can be output to the terminal screen, a printer or a file.

### Prompts

#### Menu of Filesystem Devices

The program begins with a menu of filesystem devices. Not all devices listed can be analyzed (e.g., "swap" which does not have a filesystem built on it), and the program will give an error message if one of these is selected. If "wd0" or "wd1" is selected, the filesystem on the first partition on that disk will be analyzed.

Type the number of the desired device and press <RETURN>. Press <RETURN> alone to end the utility.

## Multiple Choice Menu

When an analyzable device is selected, a multiple choice prompt is displayed. The options are described below. Refer to "Standard Procedures" for instructions on using a multiple choice prompt.

### Report Type

A summary report includes the following:

- o the total number of blocks in the filesystem
- o the number of blocks free
- o the percentage of the blocks used and free
- o the maximum number of file descriptors
- o the number of file descriptors free
- o the percentage of the maximum number of file descriptors used and free

A full report includes the information in the summary report plus the following:

- o the first block number of each contiguous chunk of free space
- o the number of blocks available in each contiguous chunk of free space

## Report Device

This option gives the choice of having the report displayed on the terminal screen, output to a printer or written to a file.

If you choose to output the report to a printer, you will be asked to select the printer after exiting the multiple choice prompt.

If you choose to write the report to a file, you will be asked for the report file name after exiting the multiple choice prompt.

Type <RETURN> to proceed:

This prompt is displayed at the bottom of each report screen if the report is displayed on the terminal screen. Press <RETURN> to display the next screen. The program exits after the last screen has been displayed.

## **Filesystem Error Analysis Utility**

### Program Name

/util/fsdbg

### Related Commands

/bin/fschk

### Utility Description

The Filesystem Error Analysis Utility analyzes a filesystem device for structural errors, and optionally attempts to repair any errors found.

The utility can be run as an automatic procedure by any user, or as a user directed procedure by the system administrator. The user directed procedure allows the system administrator to examine critical filesystem structure data.

Only the system administrator may select to repair the filesystem, in either automatic or user directed modes.

In order to use the user directed option, a detailed understanding of the BOSS/IX filesystem structure is required. Without due caution, it is possible to do more harm than good with this option when in repair mode.

## Prompts

### Filesystem Devices Menu

The utility begins with a menu of filesystem devices. Not all of the devices listed can be analyzed (e.g., "swap" which does not have a filesystem built on it), and the program will repeat the filesystem device menu if one of these is selected. If "wd0" or "wd1" is selected, the filesystem on the first partition of that disk will be analyzed.

Type the number of the desired device and press <RETURN>. The selected device is then displayed in the top left corner of the screen.

Press <RETURN> alone to end the utility.

### Perform automatic filesystem analysis? (yes/no):

The response to this prompt determines whether the utility will run in automatic or user directed mode. Answer YES to perform the automatic analysis. Answer NO to perform a user directed analysis.

### Repair the filesystem? (yes/no):

The response to this prompt determines whether the utility will attempt to repair errors it finds if being run in automatic mode, or if the user will be allowed to change filesystem parameter values if run in user directed mode. Answer YES to allow repairs/changes. Answer NO to disallow repairs/changes.

**WARNING: Files or data might be lost. Proceed?  
(yes/no):**

File repair procedures may destroy data. It is recommended that the filesystem be backed up before a repair is attempted in either the automatic or user directed modes. Answer YES to continue with the procedure. Answer NO to repeat the "Perform automatic filesystem analysis" prompt.

Inode 235 contains invalid block number XXXX.  
**Delete block from inode? (yes/no):**

This prompt is displayed if the program discovers a reference in an inode to a duplicate or illegal block number during the automatic procedure in repair mode.

If you answer NO, this prompt is displayed:

**Delete the inode? (yes/no):**

You must delete either the block or the inode, or abort the utility. If you answer NO to this prompt, the "Delete block from inode?" prompt is repeated.

#### **NOTE**

In general, it is better to delete the block than the reference. In either case, data may be lost.

When the automatic filesystem analysis (and repair) is complete, a message is briefly displayed showing the number of errors found (and repaired). The filesystem devices menu is then repeated.

## Filesystem Parameters Menu

If you answered NO when asked whether to perform the automatic filesystem analysis, a menu with 29 options (number is subject to change) is displayed. A description of these options is beyond the scope of this document. Refer to the BOSS/IX Technical Reference Manual for a description of the BOSS/IX filesystem and these options.

To exit the menu, press <RETURN> alone. The filesystem devices menu is then repeated.

## NOTES

## PRINTER UTILITIES

This chapter describes the Printer Utilities. These utilities control BOSS/IX print spooler operations.

As described in Section 2, BOSS/IX provides a sophisticated spooling feature that efficiently manages the use of system printers. The utilities described here provide simple tools for printer maintenance routines.

The Printer Utilities are:

- o Submit Print Job, which submits a string file to the spooler for printing
- o Maintain Print Queue, which reports the current status of print jobs in the spool queue and allows certain characteristic of jobs in the queue
- o Change Print Form, which informs the Spooler when a form change has been made
- o Printer Status, which reports the current status of all system printers
- o Printer Parameters, which maintains print control defaults, class definitions and form definitions

The utilities that modify print jobs, namely Maintain Print Queue and Change Print Form, require spooling to be turned "ON" for the printer involved. Spooling can be turned ON or OFF for individual printers by using the Port Configuration Utility. The Submit Print Job Utility operates with spooling turned "OFF", but print job parameters are not used.

## **Submit Print Job Utility**

### Program Name

/util/psubmit

### Related Commands

/bin/lpr

### Utility Description

The Submit Print Job utility provides a interactive method for placing a file in the print queue. The utility can only submit String files and can only submit them to printers that have spooling turned ON.

Print job parameters are supplied from the Print Class ("./etc/class") and Defaults ("./etc/defaults") files, with the defaults taken in order of the user, the printer and the system. (The class and defaults files are maintained by the Printer Parameters Utility, described below.) These default parameters can be modified before the print job is submitted.

### Prompts

Type working directory, <RETURN> for no change:

The current working directory is displayed at the top of the screen. Type the full or partial pathname of the working directory and press <RETURN>.

Type class name, <RETURN> to continue:

Type the name of the print class and press <RETURN>, or press <RETURN> alone to leave the class unspecified. If the class is not specified, default parameters will be taken from the Defaults file.

Type printer name, <RETURN> for default:

At this point, a screen of parameters is displayed with the fields all empty.

Type the name of a printer and press <RETURN>. Only printers which have spooling turned ON are accepted. The default printer is usually "lp".

Type file name:

Type the full or partial path name of the String file to print and press <RETURN>. If a file of a type other than String is entered, an Invalid File Type message is displayed.

The print job parameter fields are now filled with defaults based on the information entered. If a class was specified that has spooling turned OFF, only the Printer and File fields are displayed; otherwise, there are 15 parameter fields, as described above.

Type number if item to change, <RETURN> to submit the job:

The print job parameters are displayed in a menu with their current values. All parameters have default values except the Date and Time to print parameters. Select a parameter to change and enter the new value in response to the prompt. When the parameters are all correct, press <RETURN> to submit the print job.

When the print job has been submitted, the print job number is displayed. Press <RETURN> to acknowledge the message and terminate the utility.

## **Maintain Print Queue Utility**

### Program Name

/util/pqueue

### Related Commands

/bin/lpq

### Utility Description

The Maintain Print Queue Utility reports and allows modification of print jobs for a spooled printer. Only the system administrator and the user who submitted a print job are allowed to modify a job.

### Prompts

Type printer name, <RETURN> for default printer:

Type the name of the printer to report and maintain. Only printers which have spooling turned ON can be processed by this utility; others printers will be rejected.

Type number of job to process, <RETURN> to update status:

Printer status statistics and information of the first nine jobs in the print queue are displayed. To modify the characteristics of a print job, type the number of the job and press <RETURN>. If there are more than nine jobs in the queue, press <RETURN> for the next page of job listings.

The job is printing. Do you want to stop it?  
(yes/no):

This prompt is displayed if you select a job for which the status is WORKING. The job must be stopped before it can be modified. Answer YES to stop the job and proceed with the modification process. Answer NO to return to the print job listing.

#### Job Action Menu

Type number of job action, <RETURN> to end:

The menu gives the follow job action options:

Kill - stops the job and deletes it from the print queue.

Stop - stops the job, but leaves it in the print queue. The job will be skipped until it is restarted.

Modify - allows modification of print job parameters, except for the file and the file alias.

Restart - starts a print job that has been stopped.

When the action is completed, the prompt is repeated. Press <RETURN> alone to end the utility.

## Change Print Form Utility

### Program Name

/util/pformchg

### Related Commands

/bin/lpmaint printer -fc

### Utility Description

The Change Print Form Utility notifies the spooler that the required print form has been changed for a printer. The utility begins by displaying status information for all printers. Only a printer with status "formchange" can be processed by the utility.

When a formchange has been signaled, the job waiting for the change is resumed. No checking is possible to verify that the correct form has been mounted; it is up to the operator to make sure that the correct form has been mounted and that it is aligned properly.

### Prompts

Type printer name whose form was changed, <RETURN> for default:

The status of system printers is displayed above this prompt. Change the form on the printer to process before answering this prompt. Type the name of the printer for which the form has been changed and press <RETURN>. Only printers displayed with status "formchange" are accepted.

Do you want to align the form? (yes/no):

If you answer YES, a form alignment pattern is printed, and the utility displays "the sample form has been printed." This prompt is then repeated. Make necessary adjustments to the form alignment, and repeat this process until the form is aligned properly. When it is set, answer NO.

A message is displayed stating that printing is being resumed on the specified printer. Press <RETURN> to acknowledge the message. The utility then ends.

## **Printer Status Utility**

### Program Name

/util/pstatus

### Related Commands

/bin/lpstat

### Utility Description

This utility only displays the status of printers on the system. No changes to any printer are allowed.

### Prompts

Type <RETURN> to continue, <CTL-II> to update status:

Press <RETURN> to exit the utility. Press <CTL-II> to update and redisplay the printer status display.

## **Printer Parameters Utility**

### Program Name

/util/params

### Related Commands

None, but see notes.

### Utility Description

The Printer Parameters Utility maintains definitions for print control defaults, print classes and forms. The three functions are selected from a menu displayed when the utility is first accessed. The definitions are used by the print spooler to specify parameter values for print jobs.

The three functions making up this utility are divided into create, modify, delete and report sub-functions. All users are allowed to use the report functions, but only the system administrator can create, modify or delete definitions.

### Prompts

The prompts differ for each of the utility functions.

## Maintain Print Control Defaults Function

This function creates, modifies, deletes and reports print default definitions. These defaults are defined for users, printers and the system. The defaults for the system are provided with the system software. Defaults for users and printers can be defined in addition to the system defaults.

The print control fields are:

- o Name - printer or user name
- o Priority - the default priority (0 - 9)
- o Form Name - the default form name
- o Copies - the default number of copies
- o Delete - deletes spooled file after printing
- o Requeue - requeues file after printing
- o Notify - notifies user after printing
- o Spooling - spools the file
- o Despool wait - delays printing spooled file until entire file is received.

One defaults definition, "system" is required for this function to operate.

### Create/Modify Defaults Definition

These functions create and modify print job default definitions. They function identically except for the first prompt.

Type operator or printer name, <RETURN> to end:

Type the name of the operator or printer for which the is being created/modified. If an existing default definition name is entered for the create function, or a non-existent name is entered for the modify function, a verification prompt allows you to switch functions and continue with the procedure.

Type number of item to change, <RETURN> when all are correct:

The current default parameters for the user or printer are displayed. Default parameters are provided for new definitions. Select the number of the parameter to change.

For the Name, Priority, Form name, and Copies fields (options 1 through 4), the cursor moves to the first character of the field. Type the new value and press <RETURN>.

The Delete, Requeue, Notify, Spooling and Despool wait fields (options 5 through 9) rotate between "yes", "no" and empty as values. Enter the number of the parameter to switch to the next value. The empty value allows the value to be determined by a definition lower in priority.

When all values are correct, press <RETURN> to return to the menu.

#### **Delete Defaults Definition**

This option deletes the defaults definition for a user or printer.

Type operator or printer name, <RETURN> to end:

Type the name of the defaults entry to delete and press <RETURN>. The "system" default definition cannot be deleted.

Do you want to delete this default definition?  
(yes/no):

The parameters for the specified operator or printer are displayed and you are prompted to verify that it is to be deleted. Answer YES to delete the displayed definition.

#### Report Defaults Definition

This option reports the defaults definitions for all users and printers.

#### Report Device Menu

Select to have the report displayed on the terminal screen, written to a file or output to a printer.

If you select "printer", you will be asked to select a printer name from a menu. If you select "file", you will be asked to specify a file name.

End of report. Type <RETURN> to continue:

When the report has been produced, this prompt is displayed. It is displayed at the last page of a report displayed on the terminal screen.

## Maintain Class Definitions Function

This function maintains print class parameter definitions. The parameters are the same as the print control defaults except that the class name is not tied to a user or printer.

### **Create/Modify Class Definition**

These functions create and modify print class definitions. They operate identically except for the first prompt.

Type class name, <RETURN> to end:

Type the name of the class definition to be created/modified. If an existing class definition name is entered for the create function, or a non-existent name is entered for the modify function, a verification prompt allows you to switch functions and continue with the procedure.

Type number of item to change, <RETURN> when all are correct:

The current class definition parameters for the specified class are displayed. Default parameters are provided for new definitions. Select the number of the parameter to change.

For the Class name, Priority, Form name, and Copies fields (options 1 through 4), the cursor moves to the first character of the field. Type the new value and press <RETURN>.

The Delete, Requeue, Notify, Spooling and Despool wait fields (options 5 through 9) rotate between "yes", "no" and empty as values. Enter the number of the parameter to switch to the next value. The empty value allows the value to be determined by a definition lower in priority.

When all values are correct, press <RETURN> to return to the menu.

### Delete Class Definition

This option deletes a print class definition.

Type class name, <RETURN> to end:

Type the name of the class definition to delete and press <RETURN>.

Do you want to delete this class definition?  
(yes/no):

The parameters for the specified class definition are displayed and you are prompted to verify that it is to be deleted. Answer YES to delete the displayed definition.

### Report Class Definition

This option reports all class definitions.

## Report Device Menu

Select to have the report displayed on the terminal screen, written to a file or output to a printer.

If you select "printer", you will be asked to select a printer name from a menu. If you select "file", you will be asked to specify a file name.

End of report. Type <RETURN> to continue:

When the report has been produced, this prompt is displayed. It is displayed at the last page of a report displayed on the terminal screen.

## Maintain Form Definition Function

This function maintains printer form definitions. Form definitions contain information such as form length and width, number of lines per inch, number of characters per inch, special translation tables, and other information that determines how files are printed.

The function opens with a menu allowing the user to create, modify, delete and report form definitions.

One form definition, "stanrd", is supplied with the BOSS/IX operating system, and is required for the operation of this function.

## Create/Modify Form Definitions

These sub-functions operate identically, except for the first prompt.

Type a form name, <RETURN> to end:

Type the name of the form to create or modify and press <RETURN>. If an existing form name is entered for the "create" function, or a non-existent form name is entered for the "modify" function, a verification prompt is displayed allowing you to change operations.

### Form Definition Parameters Menu

The current definition parameters for the specified form are displayed in a menu. Parameters for a new form definition are supplied from the "stanrd" form, and need to be changed for the new form.

The parameters are:

- o Form name - as specified
- o Length - lines, 1 to 255
- o Width - characters, 1 to 255
- o Lines per inch - 6, 8 or empty
- o Pitch - 10, 12, 16 or empty
- o Translation table - 2-character name
- o Hand sheet feed - yes, no or empty
- o Slew 2-8 - maximum varies for printers
- o Printers - name of printers to use the form, or \* for all printers

Type the number of the item to change. Items 4, 5 and 7 (lines per inch, pitch and hand sheet feed) rotate through the options each time the number is entered. The remaining options prompt for the value. Enter the value and press <RETURN>.

When all values are correct, press <RETURN> alone to save the parameters.

### Delete Form Definition

This option deletes a specified form definition.

Type a form name, <RETURN> to end:

Type the name of the form to delete and press <RETURN>. The form "stanrd" cannot be deleted.

Delete this form definition? (yes/no):

The parameters for the specified form are displayed and you are prompted to verify that it is to be deleted. Answer YES to delete the form, answer NO to keep it.

### Report Form Definitions

This option creates a report of all form definitions.

### Report Device Menu

This menu gives the option of having the report displayed on the terminal screen, written to a file or output to a printer.

If you select "printer", you will be asked to select the printer from a menu of printers. If you select "file", you will be asked to specify the file name.

End of report. Type <RETURN> to continue:

When the report is finished, this prompt is displayed. For a report displayed on the terminal screen, this prompt is displayed at the bottom of the last page of the report. Press <RETURN> to acknowledge the prompt and return to the options menu.

**The Printer Parameters Utility modifies these files:**

- o /etc/defaults
- o /etc/class
- o /etc/forms

These files can be modified using the text editor, "ved". If the "system" defaults definition or the "stanrd" form definition are missing, they must be created using the text editor. The formats of these files are described in Section 6, "System Files."

NOTES

## **SAVE AND RESTORE UTILITIES**

This subsection describes the Save and Restore Utilities. These utilities create backup copies of a fixed disk filesystem or individual files on floppy diskettes or MCS tape cartridge. Floppy diskette backup and restore is not supported on the MAI 3000. Backup to MTS reel tape must be performed using commands.

Backup floppies written by the Save to Diskette Utility are not filesystem devices, so they cannot be mounted and accessed directly. The files must first be restored to the filesystem using the Restore from Diskette Utility.

MCS cartridges written by the Save to MCS Utility are the same format as cartridges written by the "mcssave" command. Accordingly, the cartridge utilities and commands can be used interchangeably.

**The Backup and Restore Utilities are:**

- o Save to cartridge
- o Restore from cartridge
- o Report save cartridge contents
- o Compare cartridge
- o Label cartridge
  
- o Save to diskette
- o Restore from diskette
- o Report save diskette contents

## **Save to Cartridge Utility**

### Program Name

/util/msave

### Related Commands

/bin/mcssave

### Utility Description

The Save to Cartridge Utility copies files from an active filesystem (either the root or a mounted filesystem) to one or more MCS cartridges. If a file being saved is too long to fit in the remaining space of a tape, the program splits the file and puts the overflow on the next tape.

In general, this utility should be run while only one user is logged on the system. Although it is not necessary to be in single user mode, caution should be exercised to prevent another user from modifying a file while it is being saved.

### Prompts

### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Error Handling

The Error Handling option list gives you a choice of three error handling options:

1. You can record errors in a log file and continue the backup without interruption,
2. You can display errors on the screen and pause to allow the operator to specify how each error should be handled, as well as record them in a log file,
3. You can display the errors directly on the screen rather than saving them in a log file. In this case the program must pause to ask how to handle the errors.

## Verify

The Verify option gives you the choice of verifying the information saved on the tape against the original file on disk. Verification makes the utility take longer to run, but the copy is more reliable.

## Report Device

The Report Device option determines whether the report of files saved is to be displayed on the terminal screen, written to a file or output to a printer, or that no report is generated. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

Type a file or filelist name to add, <RETURN> when done:

Type the full or partial name of a file to back up, or the name of a filelist containing the names of files to back up, and press <RETURN>. Pattern matching characters may be used.

Press <CTL-III> at this prompt to modify the file selection environment, as described in "Standard Procedures."

#### Edit List of Names Menu

Refer to "Standard Procedures" for instructions on this menu and its options.

Mount a tape. Type <RETURN> when ready:

Mount the (first) cartridge to be used for the backup and press <RETURN>. The program then reads and displays the current cartridge label.

Do you want to append to this tape? (yes/no):

If the cartridge is already labeled for backup, the program allows the option of keeping the existing backup(s) and appending the new backup to the end of the tape. Answer YES or NO.

Type save set name:

Type a name for the save set and press <RETURN>. You may also press <RETURN> for the default name, which is the current date. The save set name can be up to 20 characters long. The program then saves the specified files.

Do you want to use this tape? (yes/no):

If the cartridge is not labeled for backup, a blank label is displayed. The "cartridge set name" field says either "No label," "Bad label format," or "Label cannot be read." Answer YES or NO.

If you answer YES, you are allowed to change the label fields. The cartridge is then labeled. The program then saves the specified files.

This cartridge is full.

Insert another cartridge. Type <RETURN> when ready:

Insert another cartridge to continue the backup. Before continuing, the program repeats the process of reading the label, verifying that the tape is valid, and saving files.

## **Restore from Cartridge Utility**

### Program Name

/util/mrestore

### Related Commands

/bin/mcsrestore

### Utility Description

The Restore From Mag Tape Utility copies files from a backup MCS cartridge or MTS reel to the file system. The backup tape could have been written by the Save To Mag Tape Utility or by the "mcssave" or "mtssave" commands in file-by-file mode. Filesystem image backups written by "mcssave" or "mtssave" cannot be restored by this utility.

Files are restored to their original directories, with the original owner ID and access modes. Accordingly, it is recommended that the utility be run only by the system administrator to prevent conflicts of access permission. It is further recommended that the utility be run when only one user is logged onto the system.

### Prompts

### Multiple Choice Menu

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" above for instructions on using multiple choice menus.

## File Selection

This option determines whether all files or selected files from the save set are to be restored. If selected files are to be restored, you will be asked to build the list of files after the tape has been inserted and verified.

## Duplicates

This option determines the action to be taken if the file about to be restored already exists on disk. The original disk file can be replaced, the file to be restored can be skipped, or the program can pause to ask the user for the action to take.

## Error Handling

The Error Handling option list gives you a choice of three error handling options:

1. You can record errors in a log file and continue the backup without interruption,
2. You can display errors on the screen and pause to allow the operator to specify how each error should be handled, as well as record them in a log file,
3. You can display the errors directly on the screen rather than saving them in a log file. In this case the program must pause to ask how to handle the errors.

## Verify

The Verify option determines whether or not the data restored to disk is to be verified against the data on the cartridge. Verification makes the utility take longer to run, but the restored data is more reliable.

## Report Device

The Report Device option determines whether the report of files restored is to be displayed on the terminal screen, written to a file or output to a printer, or that no report is generated. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

Insert a cartridge. Type <RETURN> when ready:

Mount the cartridge containing the save set to be restored and press <RETURN>. If the save set is split between two or more cartridges, insert the first, and then each succeeding cartridge as the files are restored and the prompt is repeated.

Is this the correct cartridge? (yes/no):

The label of the cartridge is displayed and you are asked to verify that it is the correct tape. Answer YES or NO. If you answer YES, the specified files are restored.

Type name of save set to restore:

This prompt is displayed if the tape is correct. Type the name of the save set to restore and press <RETURN>. To restore the first save set on the cartridge, you may press <RETURN> alone.

Type a file or filelist name to add, <RETURN> to end:

If selected files are being restored, you are asked to build a list of files. Refer to "Standard Procedures" for further instructions.

Insert volume number xx of save set "name".

Type <RETURN> when ready:

If the save set is split between two or more cartridges, this prompt is displayed when the next tape is required. Change cartridges and press <RETURN>.

#### Error Messages

Cannot restore file "name".

Some directory in the path does not exist.

This message usually results if a directory in the path name of the file being restored does not exist on disk.

## **Report Save Cartridge Contents Utility**

### **Program Name**

**/util/mlist**

### **Related Commands**

**/bin/mcslist**

### **Utility Description**

The Report Save Cartridge Contents Utility produces a report of the files contained in a single save set on backup MCS cartridges. The report is generated from the save set directory that is written on the cartridge before the files were saved. Accordingly, any files that were not saved, due to a disk read error or the file having been deleted prior to saving, are included in the report.

### **Prompts**

#### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Report Device

The Report Device option determines whether the report of files is to be displayed on the terminal screen, written to a file or output to a printer, or that no report is generated. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

Insert a cartridge. Type <RETURN> when ready:

Insert the cartridge containing the save set to be reported and press <RETURN>. If the save set is split between two or more cartridge, insert the first in the series.

Is this the correct cartridge? (yes/no):

When the cartridge is ready, the program reads the label and displays it. If the cartridge is the correct one, answer YES; otherwise answer NO.

Type name of save set to list:

If the cartridge is correct, this prompt is displayed. Type the name of the save set to display and press <RETURN>.

The program then produces the report of files in the save set.

## **Compare Save Cartridge Utility**

### **Program Name**

**/util/mcompare**

### **Related Commands**

**/bin/mcscompare**

### **Utility Description**

The Compare Save Cartridge Utility compares the files from a single save set to the disk files with the save path names. It reports any differences between the files.

If files were saved from mounted filesystems, those filesystems must be mounted to the root filesystem when the save set is compared so the files can be found.

### **Prompts**

### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Report Device

The Report Device option determines whether the report is to be displayed on the terminal screen, written to a file or output to a printer, or that no report is generated. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

Insert a cartridge. Type <RETURN> when ready:

Insert the cartridge containing the save set to be compared and press <RETURN>. If the save set is split between two or more cartridges, insert the first one.

Is this the correct cartridge? (yes/no):

When the cartridge is ready, the program reads the label and displays it. If the cartridge is the correct one, answer YES; otherwise answer NO.

Type name of save set to compare:

If the cartridge is correct, this prompt is displayed. Type the name of the save set to compare and press <RETURN>.

The program then produces the report of files in the save set.

## **Label Cartridge Utility**

### Program Name

/util/mlabel

### Related Commands

/bin/mcslabel

### Utility Description

The Label Cartridge Utility displays the current label on an MCS cartridge and optionally changes it. Labeling a cartridge initializes it, erasing all data on the cartridge.

The utility can create labels for purposes other than backup, so the cartridge labeled might not be usable by the backup and restore programs.

### Prompts

Insert a cartridge. Type <RETURN> when ready:

Insert the cartridge to be labeled and press <RETURN>. The program reads and displays the current label. If the cartridge is not labeled, a blank label is displayed.

Do you want to use this cartridge? (yes/no):

Answer NO if this is the wrong cartridge or if you do not want to change the label. The program then exits.

Answer YES to continue labeling the cartridge.

Type number of item to change, <RETURN> to label cartridge:

Type the number of the cartridge label field to change and press <RETURN>. When all changes have been made, press <RETURN> alone to label the cartridge.

The "Cartridge set name," "Cartridge ID," and "Cartridge serial number" can be any 8 character string. The "Cartridge volume number" field can be any number from 1 through 99. The "Cartridge format" field rotates through the four format options: Backup, Dump, Intersystem, Boot.

The program then erases and relabels the cartridge.

## **Save to Diskette Utility**

### Program Name

/util/dsave

### Related Commands

None

### Utility Description

The Save To Diskette Utility copies files from any active filesystem to one or more floppy diskettes.

If a file being saved is too long to fit in the remaining space of a diskette, the program splits the file putting the overflow on the next diskette.

Files saved using this utility can only be read by using the Restore From Diskette Utility. Save diskettes cannot be mounted and accessed through the file system.

Diskettes must be formatted prior to being used for backup. The Format Diskette Utility or "devfmt" command can be used to format diskettes.

### Prompts

### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Report Device

The Report Device option determines whether the report of files saved is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after exiting the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after exiting the multiple choice prompt.

## Write Verify

The Write Verify option determines whether or not the data written to diskette is to be verified against the data on disk. Verification makes the utility take longer to run, but the copy is more reliable.

Type a file or filelist name to add, <RETURN> when done:

This prompt begins the file selection procedure. Refer to "Standard Procedures" for details of this procedure.

Press <CTL-III> at this prompt to change the file selection environment.

Type save set name:

Type a name for the backup save set. The name may be any string up to 8 characters long. If you press <RETURN> alone, the save set will not be named. The save set information is then displayed.

Insert diskette number 1, <RETURN> to proceed:

Insert a formatted diskette and press <RETURN>.

All data on the diskette will be lost! Proceed?  
(yes/no):

There is no append option for this utility. Any data on the diskette will be written over. Answer YES to proceed with the backup.

Diskette is full.

Insert diskette number x, <RETURN> to proceed:

When a diskette is full and there are more files to be saved, you must insert another diskette.  
Insert the next diskette and press <RETURN>.

Mark the diskettes with their number in the set.

### Error Handling

Diskette error.

Insert another diskette, then type <RETURN> when ready:

This error is displayed if the diskette is faulty. Remove and discard the diskette and insert another. The utility begins again where it began with the faulty diskette.

Disk read error

If a read error occurs while attempting to read data from the source disk, a menu with three options is displayed:

1. Save partial file
2. Skip this file
3. Abort

The "Save partial file" option saves as much of the file as it can read. In this case, the "complete file" column in the final report will indicate "NO". The saved data can be restored, but might not be usable.

The "Skip this file" option skips the damaged file, and overwrites any part of it that has already been saved. Copying resumes with the next file.

The "Abort" option terminates the utility immediately.

## **Restore from Diskette Utility**

### Program Name

/util/drestore

### Related Commands

None

### Utility Description

The Restore From Diskette Utility restores all or selected files from backup diskettes written by the Save To Diskette Utility.

Files are restored with the same directory paths they had when they were saved. The user's working directory has no effect on restoring files.

### Prompts

#### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

#### **File Selection**

The File Selection list determines whether all or selected files in the save set are to be restored. If you specify "selected files", you will be asked to build a list of files, using the procedure described under "Standard Procedures."

## Duplicates

The Duplicates list determines the action the program is to take if the file about to be restored already exists on disk.

The "replace" option causes the copy on diskette to replace the copy on disk. The "skip" option causes the file to be skipped, retaining the copy on disk unchanged. The "ask user" option causes the program to pause to ask the user whether to replace or skip each time a duplicate is encountered.

## Contiguous

The Contiguous option determines whether the program will attempt to restore the file to contiguous blocks on disk to reduce fragmentation of the file on disk. If set to "no", the program will restore the file using the lowest available blocks on disk.

## Report Device

The Report Device option determines whether the report of files restored is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

Insert the LAST diskette of the save set, then type <RETURN> to proceed:

If you are restoring selected files, this prompt is displayed. The directory of files in the save set is contained on the last diskette, and must be read before files can be restored. Insert the diskette and press <RETURN>.

Insert diskette number 1, <RETURN> to proceed:

Insert the first diskette in the set containing files to be restored and press <RETURN>. The save set information is then displayed.

Is this the correct save set? (yes/no):

If the save set is correct, answer YES. The program then begins restoring files.

Insert diskette number n, <RETURN> to proceed:

Remove the diskette in the drive and insert the indicated diskette. Press <RETURN> to continue restoring files.

## Error Handling

### Diskette Error

If a read error occurs while attempting to read data from the save diskette, a menu with three options is displayed:

1. Restore partial file
2. Skip this file
3. Abort

The "Restore partial file" option restores as much of the file as it can read. The restored data might not be usable, depending on what data is missing.

The "Skip this file" option skips the damaged file, and overwrites any part of it that has already been restored. Restoring resumes with the next file.

The "Abort" option terminates the utility immediately.

### Disk Error

If a disk write error occurs, the program displays three processing options:

1. Retry
2. Skip this file
3. Abort

The "Retry" option causes the utility to attempt the write again. If it still fails, the menu is repeated.

The "Skip" option causes the program to skip the file and resume with the next file. The "Abort" option causes the program to terminate immediately.

## **Report Save Diskette Contents Utility**

### Program Name

/util/dsaverpt

### Related Commands

None

### Utility Description

The Report Save Diskette Contents Utility produces a report of the files is a save set of diskettes. The diskettes must have been written by the Save To Diskette Utility for this utility to work.

### Prompts

#### Report Device Menu

This is a single list multiple choice menu. Refer to "Standard Procedures" for instructions on using multiple choice menus.

If you select "printer", you will be asked to select a printer from a menu after exiting this prompt. If you select "file", you will be prompted for a report file name after exiting this prompt.

Insert the LAST diskette of the save set, then type <RETURN> to proceed:

The directory of files in the save set is on the LAST diskette. Insert the diskette and press <RETURN>. The program then displays the save set information.

Is this the correct save diskette? (yes/no):

If the information displayed is for the desired save set, answer YES. The program then produces the report.

#### Notes

The report lists the file name, the number of the diskette containing the file, the number of bytes stored, and whether or not the file is complete. Files that are split between diskettes are listed once for each diskette that contains part of the file. The file size is the number of bytes of the file are on that diskette.

A complete file is a file that was completely saved, whether on one or more diskettes. An incomplete file is a file that was only partially saved due to a disk read error.

## NOTES

## **BASIC PROGRAM UTILITIES**

This subsection describes the BASIC Program Utilities. These utilities are designed as aids for Business BASIC programmers, providing useful tools in the BASIC programming environment.

The BASIC Utilities are:

- o **BASIC Program Cross-reference**, which produces a formatted listing of programs and cross-references statement elements
- o **BASIC Program Renumber**, which renames the statements in a BASIC program
- o **BASIC Program String Search & Replace**, which searches one or more programs for a specified string and optionally replaces it with another
- o **Compare BASIC Programs**, which compares the statements of pairs of BASIC programs and reports any differences
- o **Merge BASIC Programs**, which merges one program into another
- o **Display BASIC Program Names**, which lists all BASIC programs in a directory
- o **Encrypt BASIC Program**, which encrypts a BASIC program making it unlistable

## **BASIC Program Cross-reference Utility**

### Program Name

/util/bxref

### Related Commands

None

### Utility Description

The BASIC Program Cross-Reference Utility produces a formatted listing of one or more BASIC programs and includes cross-referencing information for specified program elements. The cross-referencing includes:

- o variables
- o arrays, functions, system variables
- o statement numbers
- o directives
- o global variables

### Prompts

#### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will list and cross-reference a single program file or several selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Report Device

The Report Device option determines whether the listing and cross-reference report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

## Cross-reference Options Menu

The Cross-reference Options Menu is a toggle prompt. Press the number of an option to select/deselect that option. Refer to "Standard Procedures" for further instructions. The options are:

- o Program listings - lists each specified program
- o Variables - lists all variables used in each specified program

- o Arrays, functions, special variables - lists all arrays, functions and system variables in each specified program
- o Statement numbers - lists statement numbers referenced in branching operations
- o Directives - lists all directives used in each specified program
- o Global variables - lists all variables occurring in more than one of the specified programs
- o Formatted listing - the program listing is formatted, indenting loops and conditions and separating compound statements to one statement per line

The program then produces the reports according to the options selected. When the report is finished, the program terminates.

## **BASIC Program Renumber Utility**

### Program Name

/util/brenumber

### Related Commands

None

### Utility Description

The BASIC Program Renumber Utility renumbers the statements in a BASIC program to a specified increment. The utility can renumber an entire program, or a specified range of statements within a program. All statement references in the program, except those in REM statements, are changed to reflect the renumbering.

### Prompts

#### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will renumber a single program file or several selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Report Device

The Report Device option determines whether the report of programs renumbered is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

## Increment

This option specifies that the program statements are to be incremented in steps of 10 or in an increment to be specified by the user.

## Range

The Range option specifies whether the entire program or a range of statements to be specified by the user are to be renumbered.

## Create Backup File

The Create Backup File option determines whether the original program will be retained in a backup file, or replaced by the renumbered program. In either case, the original program name is given to the renumbered program.

In order to create the backup file, the name of the original file must be 16 characters or less. This requirement is to accommodate the ".bak" extension. If the file name is too long, the backup is not created.

Type line number increment (<RETURN> = 5):

If the increment size was to be specified, this prompt is displayed. Type the increment size and press <RETURN>.

Type first line number of range:

Type last line number of range:

If the range to be renumbered was to be specified, these prompts are displayed in succession. Type the first (last) line number in the desired range and press <RETURN>.

The utility then renames the specified program(s) and produces the report. When processing is finished, the utility terminates.

## **BASIC Program String Search and Replace Utility**

### Program Name

/util/bsearch

### Related Commands

None

### Utility Description

The BASIC Program String Search and Replace Utility searches one or more BASIC programs for a specified string. The string can optionally be replaced by another specified string. The search can be restricted to BASIC code segments or literals, and can be directed to pause on a match or to make the specified change without pause.

### Prompts

### **Multiple Choice Menu**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will search a single program file or several selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Report Device

The Report Device option determines whether the report of matches and replacements is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

## Action

Three search and replace action options are available. The "search only" option searches for the search string and reports any occurrences found. The "search & replace" option searches for the search string and replaces all occurrences with the replace string without pausing. The "pause on match" option searches for the search string and requests the next action, either to replace with another string or to continue searching.

## Scope

The Scope option determines the context in which the search string must occur in order to match. The match can be restricted to literal (e.g., quoted strings), BASIC code, or any occurrence regardless of context.

## Create Backup File

The Create Backup File option determines whether the original program will be retained in a backup file, or replaced by the modified program. In either case, the original program name is given to the modified program.

In order to create the backup file, the name of the original file must be 16 characters or less. This requirement is to accommodate the ".bak" extension. If the file name is too long, the backup is not created.

## Type search string (35 char. max.):

Type the string to be searched and press <RETURN>.

## Type the replacement string (35 char. max.):

This prompt is displayed if the "search & replace" option is specified. Type the string to replace the search string and press <RETURN>.

## Search for variables only? (yes/no):

If the search string entered is interpretable as a variable, this prompt is displayed. If you are searching only for variables, answer YES; otherwise, answer NO.

ENTER REPLACE STRING:

<RETURN> to skip, <CTL-II> to delete, <CTL-III> previous action

If the "pause on match" action was specified, this prompt is displayed for each match. The replace string may be up to 35 characters long.

Press <RETURN> (or <CTL-I>) alone to make no change and continue searching. Press <CTL-II> to delete this occurrence of the search string. Press <CTL-III> to repeat the action taken on the previous match. <CTL-IV> terminates the program.

When the program has been searched, the report of matches found and replacements made is produced. The utility then terminates.

## Compare BASIC Programs Utility

### Program Name

/util/bcompare

### Related Commands

None

### Utility Description

The Compare BASIC Programs Utility performs a line-by-line comparison of one or more pairs of programs and produces a report of their differences. The comparison of multiple pairs of files is performed on selected files and files in a single directory with the same names as the selected files.

### Prompts

#### Multiple Choice Menu

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will compare a single pair of program files or several selected files. If you select "single file" you will be prompted for the file names when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu and the directory containing the compare files. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Report Device

The Report Device option determines whether the comparison report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "file" you will be asked for the file name after you exit the multiple choice prompt. If you select "printer" you will be asked to select a printer from a menu after you exit the multiple choice prompt.

### Type name of file to compare:

If "single file" was selected, this prompt is displayed twice. Enter the names of the two files to be compared and press <RETURN>.

### Type name of directory to compare:

If "selected files" was selected, this prompt is displayed following the usual file selection procedure. Specify the directory containing the files to be compared to the specified files. The file names in this directory must be the same as the names specified. No selection is allowed for this step.

/file/name CANNOT BE FOUND  
<RETURN> TO SKIP, <CTL-IV> TO ABORT:

This prompt is displayed if there is no file in the specified directory with the same name as one of the specified files. Press <RETURN> to skip the file and continue the comparison with the next file. Press <CTL-IV> to terminate the utility.

After all files have been processed and the report has been generated, the program terminates.

## **Merge BASIC Programs Utility**

### Program Name

/util/bmerge

### Related Commands

MERGE Directive in BASIC

### Utility Description

The Merge BASIC Programs Utility merges a single program into one or more other programs.

Unlike the MERGE directive in BASIC, this utility merges programs directly rather than merging an Indexed or Serial file into a program.

### Prompts

#### Multiple Choice Menu

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will merge a program into a single file or into selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you select "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard Procedures" for instructions on using the file selection procedure.

## Create Backup File

The Create Backup File option determines whether the original program will be retained in a backup file, or replaced by the modified program. In either case, the original program name is given to the modified program.

In order to create the backup file, the name of the original file must be 16 characters or less. This requirement is to accommodate the ".bak" extension. If the file name is too long, the backup is not created.

Type name of merge program (destination):

If "single file" was specified, this prompt asks for the name of the destination file, i.e., the program into which the merge program will be merged.

Type name of program to be merged:

This prompt asks for the program to be merged into the specified file(s). Type the name and press <RETURN>.

While the program is processing, the names of the destination files are displayed as they are being merged.

When all files have been merged, the program terminates.

## **Display BASIC Program Names Utility**

### Program Name

`/util/bdisplay`

### Related Commands

`/bin/ls -l` lists all files in a directory,  
with file attributes

### Utility Description

The Display BASIC Program Names Utility produces a report of the BASIC program files in a directory. No other file types are included in the report.

### Prompts

#### **Multiple Choice Prompt**

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## Directory

The Directory option determines whether the contents of the current working directory or of another directory are to be displayed. If you select "other", you will be prompted for the new working directory when you leave the multiple choice prompt.

## File Selection

The File Selection option determines whether all or selected files are to be reported. If you select "selected files", the file selection process (described in "Standard Procedures") is invoked to specify files when you leave the multiple choice prompt.

## Report Detail

The Report Detail option specifies the amount of detail in the report. The report can include file names only or file names and file attributes. The file attributes are:

- o Full directory path and file name
- o Number of blocks used
- o Program ID and initial remark, if any

## **Report Device**

The Report Device option determines whether the report is to be displayed on the terminal screen, written to a file or output to a printer. If you select "printer", you will be asked to select a printer from a menu of printer names when you leave the multiple choice prompt. If you select "file" you will be prompted for a file name when you leave the multiple choice prompt.

When the report has been produced, the utility terminates.

## **Encrypt BASIC Program Utility**

### Program Name

/util/bencrypt

### Related Commands

ENCRYPT Directive in BASIC

### Utility Description

The Encrypt BASIC Program Utility encrypts BASIC programs, saving them in an unlistable format. The utility allows you to create an unencrypted backup of the file.

### Prompts

#### Multiple Choice Menu

The utility begins with a multiple choice prompt. The multiple choice parameter lists are described in the following paragraphs. Refer to "Standard Procedures" for instructions on using multiple choice menus.

## File Selection

The file selection option determines whether you will encrypt a single program file or several selected files. If you select "single file" you will be prompted for the file name when you exit the multiple choice menu. If you choose "selected files" you will be asked to build a list of file names when you exit the multiple choice menu. Refer to "Standard procedures" for instructions on using the file selection procedure.

## Create Backup File

The Create Backup File option determines whether the original program will be retained in a backup file, or replaced by the encrypted program. In either case, the original program name is given to the modified program.

In order to create the backup file, the name of the original file must be 16 characters or less. This requirement is to accommodate the ".bak" extension. If the file name is too long, the backup is not created.

Type name of file to be encrypted:

This prompt is displayed if a single file is being encrypted. Type the name of the program and press <RETURN>.

When all specified programs have been encrypted, the utility terminates.

NOTES

NOTES

## **SYSTEM UTILITIES**

This chapter describes the System Utilities. These utilities perform system administration functions. The utilities are:

- o Date and Time, which sets the system date and time and specifies the format in which they are displayed
- o Port Configuration, which displays and modifies the configuration of terminals and printers
- o Edit Menus, which creates and modifies menus and menu systems
- o Operator Information, which creates and modifies the operator information file

## **Date and Time Utility**

### Program Name

**/util/sdate**

### Related Commands

**/bin/date**

### Utility Description

The Date and Time Utility controls the system date and time. The utility performs the following four functions:

- o display system date and time
- o set date
- o set time
- o set date and time display formats

Only the system administrator can set the date, time and display formats.

### Prompts

Type <RETURN> to continue:

For users other than the system administrator, the current date, time and display formats are displayed on the terminal screen. No changes can be made. Press <RETURN> to end the utility.

## Date and Time Function Menu

For the system administrator, the current system date, time and display formats are shown. The menu allows you to select to set the date, set the time and set the date and time display formats. Each of these functions is described below.

Type the number of the desired function and press <RETURN>. Press <RETURN> alone to end the utility.

### Set Date Function

Type year number, <RETURN> for no change:

Type either the four-digit or two-digit year number and press <RETURN>. Press <RETURN> alone to leave the current year unchanged.

Type month name or number, <RETURN> for no change:

Type the month name or number and press <RETURN>. If entering the name, at least the first three letters must be given; the rest is not necessary. Press <RETURN> alone to leave the month unchanged.

Type day number, <RETURN> for no change:

Type the number of the day and press <RETURN>. Press <RETURN> alone to leave the day number unchanged.

After the day number has been entered, the system date display is updated and the functions menu is repeated.

## Set Time Function

Type time in format hhmmss, <RETURN> for no change:  
Type time in format hhmm, <RETURN> for no change:

Type the numeric representation for the time in hhmmss or hhmm format as shown in the prompt, where:

hh = a number from 1 to 12 (for 12-hour clock format) or from 1 to 24 (for 24-hour clock format) for the hour. The leading zero is allowed but not required for single digit hours.

mm = a two-digit number from 00 to 59 for the minute (leading zero is required)

ss = a two-digit number from 00 to 59 for the second (leading zero is required)

Then press <RETURN>. The digits are checked for validity. If they are incorrect, the prompt is repeated.

When a legal value has been entered, the utility proceeds. If the current time format is 24-hour clock, the time display is updated and the functions menu is repeated.

### **12-hour clock half day choice**

If the current time format is 12-hour clock, a menu is displayed allowing you to specify morning (am) or evening (pm). Select an option and press <RETURN>. The time display is then updated and the functions menu is repeated.

## Set Date & Time Format Function

This function runs as a sequence of the following three menus.

### Date Format Menu

This menu displays the three allowed date display formats: month/day/year, day/month/year and year/month/day.

### Time Format Menu

This menu displays the two allowed time formats, differing only in the inclusion of seconds in the display.

### Clock Format Menu

This menu allows choosing a 12-hour or 24-hour clock format.

When all choices have been made, the display is updated to reflect any changes to the formats and the functions menu is repeated.

## **Port Configuration Utility**

### Program Name

/util/configure

### Related Commands

None, but see notes

### Utility Description

The Port Configuration Utility displays, adds and modifies the terminal and printer devices configured on the serial and parallel ports. Any user can use the utility to display the port configuration, but only the system administrator can modify the configuration.

The utility should only be executed while the system is in single-user mode to assure system integrity. Using the utility to modify configured ports while they are in use can be destructive of data as well as disruptive to the operator using that port.

The utility begins with a display of the ports and the devices configured on them. A menu of Port configuration options allows you to:

- o Add a device
- o Remove a device
- o Move a device
- o Switch two devices
- o Report port attributes
- o Modify port attributes

Each of these options is described below.

To exit the utility without saving changes, press <CTL-IV>. To exit the utility and save changes, selection option 7. Press <RETURN> alone at the options menu prompt to display more ports.

Changes made to the port configuration record files are only saved when the utility is exited normally, by pressing <RETURN> at this menu.

Changes made to the port configuration only take effect after the system has been rebooted. A prompt is displayed noting this. Press <RETURN> again at this prompt, and the utility updates the configuration files.

#### Add a Device Function

The Add a Device function configures a port for a terminal or printer.

Type the port number, <RETURN> to end:

Type the port number to configure and press <RETURN>. Press <RETURN> alone to end the function. If you select the number of an already configured port, a verification prompt is displayed allowing you to switch to the Modify Port Attributes function.

#### Device Type Menu

Three device types can be configured on BOSS/IX 7.2: terminal, printer and graphics terminal. The printer option covers graphics printers and plotters as well as text printers. This menu is skipped for port 2, since only a parallel printer is allowed on this port.

## Terminal Types Menu

If "terminal" or "graphics terminal" was selected, a menu of terminal types is specified. The terminal types displayed correspond to mnemonics tables defined in the "/etc/ttymntbl" directory.

## Printers Menu

If "printers" was selected, the menu of printer types is displayed.

## Terminal/Printer Names Menu

Select a terminal or printer name. It is usual to select either the first name available or the name that corresponds to the port number.

## Menu of Port Parameters

The menu of port parameters varies with the device type selected. Default values are supplied and generally should not be changed, except that any comment (item 5) can be added. If you choose to change any of these parameters, a prompt asks for the new value, usually as a selection from a menu or as a yes/no prompt. These additional prompts are not described here.

If you change the slave printer parameter for a terminal, an abbreviated printer configuration menu is displayed when you leave this menu.

The port is configured only after the port parameters menu is exited normally by pressing <RETURN>.

### Remove a Device Function

This function is used to remove the configuration record from a port.

Type the port number, <RETURN> to end:

Type the port to deconfigure and press <RETURN>. Press <RETURN> alone to exit this function.

Remove the device from this port? (yes/no):

Verify your decision to remove the device. The selected port is noted at the top right side of the screen. Answer YES to remove the device, NO to keep it.

### Move a Device Function

This function moves a port configuration record to an deconfigured port.

Type the port number of the device to move, <RETURN> to end:

Type the number of configured port which is to be moved.

Type the destination port number for the device.

Type the number of an deconfigured port and press <RETURN>.

If you enter the number of an already configured port, a message is displayed saying so. Press <RETURN> to repeat the "destination port" prompt.

The utility moves the device definition and then exits to the utility functions menu.

#### Switch Two Devices Function

This function switches the port location of two defined device configurations.

Type the number of the first port, <RETURN> to end:

Type the number of one of the ports to be switched and press <RETURN>.

Type the number of the second port:

Type the number of the other port to be switched and press <RETURN>.

The utility switches the two port definitions and then exits to the utility functions menu.

#### Report Port Attributes Function

This function produces a report of the port attributes. If the report is displayed on the terminal screen, only the port requested is reported. If the report is written to a file or printer, the configuration for all configured ports is reported.

## Report Device Menu

This menu gives the choice of having the report displayed on the terminal screen, written to a file or output to a printer. Type the number of your selection and press <RETURN>.

If you select "file", you will be prompted for the name of the file after you exit this menu. The file format is suitable for printing at a later time. The attributes of all configured ports are reported. The utility then exits to the utility functions menu.

If you select "printer", you will be asked to select the printer from a menu after you exit this menu. The attributes of all configured ports are reported. The utility then exits to the utility function menu.

Type the port number, <RETURN> to end:

This prompt is displayed only if "terminal" is selected as the report device. Enter the port number and press <RETURN>. The configuration attributes are then displayed. Press <RETURN> to exit the report and repeat the port number prompt. Press <RETURN> alone to exit the function.

## Modify Port Attributes Function

This function changes the attributes of an already configured port.

Type the port number, <RETURN> to end:

Type the number of the configured port to be modified and press <RETURN>. If a port that has not been configured is selected, a verification prompt is displayed allowing a device to be added to the port.

### Port Attributes Menu

The configuration attributes of the specified port are displayed. Type the number of the parameter to be changed and press <RETURN>. Press <RETURN> alone to define the port with the displayed attributes.

The prompts for changing the port attributes are not described here. In general the prompt supplies a menu of options or a verification prompt.

When the desired changes have been made to the port, press <RETURN> alone to exit the menu. The utility functions menu is then repeated.

### Notes

Changes to the port configuration only take effect after the system has been rebooted.

Changes can be made by editing the configuration files. The formats of these files are described in Appendix B of this manual.

/etc/ports  
/etc/terminals  
/etc/ports  
/etc/defaults  
/etc/printers  
/etc/plotters

Files in the "/dev" directory for any terminal or terminal device added or deleted are also created or deleted.

The "/etc" files can be modified with the text editor. The "/dev" files can be created with the "makedev" command.

## **Menu Editor Utility**

### Program Name

/util/menedit

### Related Commands

None

### Utility Description

The **Menu Editor Utility** maintains the BOSS/IX menu system. Menus and menu sets can be created, modified, deleted and displayed.

Security parameters assigned to menus can be used to restrict access to certain menu items, or to restrict a user to a single menu set.

A menu set is a group of menus associated by a common prefix. Two utilities are provided to maintain menu sets.

The **Menu Editor Utility** functions are:

- o Create menu
- o Modify menu
- o Create menu set
- o Modify menu set
- o Display menus
- o Delete menus

The files maintained by these functions are used by the menu driver program, "/util/menu", which displays the menus and controls program and menu access through the menus. The files themselves are stored in the directory "/util/mn".

Only the Display Menus function can be run by users other than the system administrator.

Note that the menus managed by this utility and used by the menu driver do not include menus "hard coded" into various programs.

### Create/Modify Menu Functions

The Create and Modify Menu Functions create and modify menu definitions. Although the functions are distinct, they operate the same. If you specify an existing menu to "create," you are notified of this and allowed to modify it. If you specify a non-existent menu to modify, you are notified of this and allowed to create it.

Type name of menu to create, <RETURN> to end:

Type name of menu to modify, <RETURN> to end:

The menu name may be up to ten characters long, including the two or three character menu set prefix (one or two characters and a period (.)). Press <RETURN> alone to exit this function.

### Part of Menu to Edit Menu

This menu allows editing the menu options, the menu action steps, the menu title and security, and the menu help text.

### Edit Menu Options

The menu options are the lines of text displayed by the menu driver describing the menus and/or programs accessible from this menu.

Only the descriptive text is entered in this option. Option numbers and screen formats are handled by the menu driver at the time the menu is displayed.

The options editor commands are displayed at the bottom of the screen, and are the same as the basic commands for the text editor, "ved", with the exception of the use of the four control keys <CTL-I> through <CTL-IV>. Characters insert at the cursor position.

Up to 15 lines of text may be entered with up to 32 characters on each line. Each line corresponds to one menu option on the screen displayed by the menu driver.

Comment lines may be entered by beginning the line with a back-slash (\). Comment lines are not displayed by the menu driver, but use up one line available for menu options. A comment line may be up to 39 characters long.

When menu options have been completed, press <CTL-IV> to exit the editor and return to the Part of Menu to Edit menu.

#### Menu Action Steps

A menu action step is the program to be executed when a menu option is selected. The menu options defined above are displayed in the numbered format as displayed by the menu driver.

Type menu option number, <RETURN> to end:

Type the number of the option for which the action step is to be defined and press <RETURN>. Press <RETURN> alone to exit this function.

Type new security code, <RETURN> for no change:

If a security code is assigned, menu access is restricted to users with a matching access code or no access codes. Type a single letter or numeral for the security code and press <RETURN>. Press <RETURN> alone for no security code. Type a space and press <RETURN> to remove a security code.

The menu action steps editor screen is then displayed. The editor works the same as it does for the Menu Options editor, except that a line may contain up to 64 characters and 16 lines are available.

The syntax for action step command lines is the same as for the command language except that arguments must be separated by commas instead of spaces. Full path names of programs and files should be given to avoid search errors.

If an action step is a BASIC program, it must begin with the file name "/bin/basic". The program name and arguments then follow this file name, as described in the command language section. For example:

/bin/basic,s=128,pgm=/apps/PGRM

starts the basic program "PGRM" with a start size of 128 pages.

If the action step is another menu, the menu driver must be executed followed by the menu name (the directory path is not required in this case), for example:

```
menu ut.dir
```

The action step "logoff" will log the user off of the system.

When you are finished editing the action steps, press <CTL-IV> to return to the "Type menu option number" prompt.

### Menu Title and Security

This selection allows the assignment and modification of the following:

- o **Menu title** - 0 to 60 characters displayed at the top center of the menu screen.
- o **Security code** - a single character restricting menu access to users with a matching security code or no codes.
- o **Revision code** - 0 to 8 character comment field available to noting the revision code of this menu.
- o **Comment** - 0 to 30 characters for documentation purposes.

The fields are supplied in menu form. Select the item to modify and enter the information at the displayed prompt. Press <RETURN> alone to exit this menu and return to the previous menu.

## Help Text

The menu driver allows you to supply help text to be displayed if the user presses <CTRL>+<Z> or the <HELP> key at the menu.

The help text editor does not limit the length or number of lines, and in this way functions just like the text editor, "ved". The only exception is that "ved" ESCAPE commands cannot be used since pressing <ESC> exits the editor. <CTL-I> displays the next 18 lines of text; <CTL-II> displays the previous 18 lines of text.

Enter the text as it should be displayed. Note that <CTRL>+<L> (which is displayed as ^L) forces a new screen, and so can be used for help text formatting.

When you are finished editing the help text, press <CTRL>+<W> to exit the editor and return to the previous menu. (<CTRL>+<C> and <ESCAPE> also exit the editor without losing text.)

## Exiting Editor Options Menu

When you are finished creating/modifying a menu and press <RETURN> alone at the part of menu to edit menu, you are prompted:

Save this menu definition? (yes/no):

Answer YES to save the modifications made to the menu. Answer NO to abandon the modifications. The "Type name of menu to create (modify)" prompt is then redisplayed.

## Create/Modify Menu Set Function

A menu set has a one- or two-character name matching the prefix of menu names in the set. The Create and Modify Menu Set functions maintain the following information:

- o Default menu in set
- o Title of menu set
- o Revision code
- o Comment

Type name of menu set to create, <RETURN> to end:  
Type name of menu set to modify, <RETURN> to end:

Type the one- or two-character name of the menu set and press <RETURN>. Press <RETURN> alone to exit this function. If you specify an existing menu set to "create," you are notified of this and allowed to modify it. If you specify a non-existent menu to "modify," you are notified of this and allowed to create it.

## **Menu Set Fields Menu**

The information fields are listed as a 4-option menu. Select the field to modify and supply the requested information

Type name of default menu, <RETURN> for no change:

Type the name of the default menu for the menu set. The menu is usually a menu in the set, but does not need to be. The default menu is "menu", the menu driver, which executes its own default menu.

Type menu set title (0-60 characters), <RETURN> for no change:

The menu set title is displayed at the top center of the menu screen and above the menu name for menus in the set. There is no default menu set title, though the menu driver has a default display.

Type revision code (0-8 characters), <RETURN> for no change:

This is a comment field supplied to document revisions in the menu set.

Type comment (0-30 characters), <RETURN> for no change:

This is a comment field supplied for menu set documentation.

### Display Menus Function

The Display Menus Function generates a report of the menus defined for the system. The report can consist of menu names or names and contents. The report can include all or selected menus, and can be displayed on the terminal, stored in a file or output to a printer.

### Multiple Choice Prompt

The function begins with a multiple choice prompt. Refer to "Standard Procedures" for instructions on using multiple choice prompts. The options available are described below.

## Menus

The Menus list specifies whether all or selected menus are included in the report. If "selected" is chosen, you are prompted for the menus or menu sets to be reported. The procedure differs slightly from the usual file selection procedure, so see the prompt description below.

## Detail

The Report Detail list specifies whether the report includes menu or menu set names only, or includes all of the fields supplied in the create/modify menu and menu set functions. The detailed report of a menu set includes definition information on the set itself plus full information on each menu in the set.

## Report Device

The Report Device list specifies whether the report is displayed on the terminal screen, written to a file or output to a printer. If you select "file", you will be prompted for the file name when you exit the multiple choice prompt. If you select "printer", you will be asked to select a printer from a menu when you exit the prompt.

Type name of menu or menu set, <RETURN> to end list:

The list created at this prompt is a temporary list only. The many options available for the usual file selection procedure (described under "Standard Procedures") are not available here.

Type the name of a menu or menu set to report and press <RETURN>. The name is not repeated on the screen, but the prompt is repeated for another entry. When the desired menus and menu sets have been entered, press <RETURN> alone to begin the report.

If the "names only" detail option was selected, the names of menus in specified menu sets are not reported unless they are individually specified.

The report is then generated. Reports displayed on the terminal are displayed one screen at a time. Press <RETURN> to display the next screen. When the report is complete, the utility functions menu is displayed.

#### Delete Menus Function

The Delete Menus function deletes menus and menu sets. If a menu set is deleted, the menus in the set may optionally be deleted.

Type name of menu or menu set to delete, <RETURN> to end:

Type the name of the menu or menu set to delete and press <RETURN>. Press <RETURN> alone to return to the utility functions menu.

Delete menu set "xx"? (yes/no):

Answer YES to delete the definition of the menu set named in the prompt. At this point, only the menu set definition is deleted; no menus are deleted. If you answer YES, you will be prompted for each menu in the set also.

Delete menu "yyyyyy"? (yes/no):

Answer YES to delete the menu definition.

## **Operator Information Utility**

### Program Name

/util/oprinfo

### Related Commands

None

### Utility Description

The Operator Information utility maintains the operator logon security to the system. For each user there is an associated logon name, user number, home directory, initial program, and optional password and menu security codes.

The utility functions are:

- o Add an operator
- o Modify an operator
- o Delete an operator
- o Change password
- o Operator report

Only the system administrator is allowed to use these functions. If a user other than the system administrator executes this utility, that user's logon information is displayed without an option to make changes.

### Add/Modify An Operator Functions

These functions create a new operator logon and modify an existing operator, respectively.

Type operator name, <RETURN> to end:

Type a logon name for the new operator and press <RETURN>. The name can contain up to 20 characters. Upper and lower case letters are treated as distinct.

If the name entered to create is the same as for an existing operator, it is rejected. If a new name is entered to modify, it is rejected.

When an operator name has been entered, default values are given for operator parameters and displayed in a menu. The fields and their defaults are:

1. Operator name: - the name entered
2. ID number: - the least user number available, greater than 0
3. Home directory: - /usr/xxx, where xxx is the operator name
4. Initial program: - menus
5. Initial menu: - no default shown; menu driver defaults to "top"
6. Security codes: - none, giving full menu access

Type number of item to change, <RETURN> to define operator:

Select an item to change, or press <RETURN> to define the operator with the displayed parameters.

Type operator name, <RETURN> for no change:

Type a new operator name, up to 20 characters long, and press <RETURN>. Changing the operator name at this point does not change the home directory.

Type ID number, <RETURN> for no change:

There is seldom a reason to change this parameter. Operator numbers less than or equal to 0 give the operator system administrator privileges. Type the new number and press <RETURN>.

Type home directory, <RETURN> for no change:

The working directory of the user running the utility is described, and the new directory may be specified relative to that directory. Type the new home directory and press <RETURN>. The full path name will be entered on the parameter line.

### Initial Program Menu

If the Initial Program field is selected, a menu is displayed allowing you to select:

1. Menus
2. Command Interpreter
3. BASIC command mode
4. Other

Type the number of the desired initial program and press <RETURN>. If you select "Other", the following prompt is displayed.

Type initial program, <RETURN> for no change:

The working directory of the operator running the utility is displayed. Type the full or partial pathname of the program to be run when the user logs on and press <RETURN>.

Type initial menu, <RETURN> for no change:

This field can be changed only if the Initial Program is "menus". Type the name of the menu to be run and press <RETURN>.

1. Add security codes
2. Delete security codes

Type number of selection, <RETURN> to end:

Select to add or delete security codes.

Type security codes to add, <RETURN> to end:

Type security codes to delete, <RETURN> to end:

Type the security codes to add or to delete from the current codes and press <RETURN>. Any alphabetic or numeric character is allowed. Upper and lower case letters represent different codes. When the codes have been entered, the parameters menu is redisplayed.

Type password:

Type password again:

When the parameters menu is exited in the Add an Operator function, but not in the Modify an Operator function, the program prompts for the optional operator password. The password must be entered twice, identically. If the entries are different, the prompts are repeated. The password may be 0 to 8 characters long. The password is not displayed while it is entered.

To change an existing operator's password, use the Change Password function.

### Delete An Operator Function

This function deletes the definition of an existing operator.

Type operator name, <RETURN> to end:

Type the name of the operator to delete and press <RETURN>. Press <RETURN> alone to return to the utility functions menu.

Do you want to delete this operator? (yes/no):

The operator parameters for the specified operator are displayed and you are asked to verify that this operator is to be deleted. Answer YES to delete the operator, or NO to keep the operator.

If you answer YES, the operator is deleted and the utility functions menu is repeated. If you answer NO, the operator is not deleted, and the Type operator name prompt is repeated.

### Change Password Function

This function changes the password for an existing operator.

Type operator name, <RETURN> to end:

Type the name of the operator whose password is to be changed and press <RETURN>. Press <RETURN> alone to return to the utility functions menu.

Type password:  
Type password again:

Type the new password. It must be entered identically twice. If it is entered differently, these prompts are repeated. The password may be 0 to 8 characters long. The password is not displayed while it is entered.

When the passwords are entered correctly, the password is changed and the utility functions menu is repeated.

#### Operator Report Function

The Operator Report Function produces a report of operator parameters (except passwords) for all users.

#### Report Device Menu

This menu specifies whether the report will be displayed on the terminal screen, written to a file or output to a printer. Select the desired report device.

If you select "printer", you will be asked to select the printer from a menu after you exit this menu. If you select "file", you will be prompted for the file name after you exit this menu.

If the report is displayed on the screen, parameters for two users are displayed at a time. When the report is finished, the utility functions menu is repeated.

NOTES

## SECTION 5

### COMMAND LANGUAGE

This section describes the BOSS/IX command language. The command language provides a user initiated interface to the BOSS/IX operating system, allowing a wide range of system functions to be performed by directly issuing a command to the system.

The BOSS/IX command language interface consists of two main parts:

1. The language itself, with both its vocabulary and syntax
2. The command interpreter, which is a program that interprets the commands you give

### THE COMMAND INTERPRETER

The command interpreter is a program that mediates the communication between you and the operating system (O.S.). It interprets commands you enter on the terminal keyboard, and then directs the O.S. to execute whatever program is required to do what you request.

While it is active, the command interpreter monitors your terminal keyboard. When you enter a string of characters the command interpreter reads the input, checks for syntactical correctness, interprets the input, and then passes the necessary instructions on to the operating system for processing.

When the instructions given to the O.S. are completed (or immediately if the process is run in the background), the command interpreter displays its prompt again, indicating that it is ready for another command.

You communicate with the command interpreter by entering commands using the "command language". When the command interpreter is ready to receive a command, it displays a prompt.

The command interpreter prompt is usually made up of your user name followed by ">". For example:

john>

A special case is the system administrator prompt, in which case the prompt is "ADMIN>".

The prompt may be modified by changing the user "environment" as described later.

The command you enter typically consists of the name of an executable file, followed by any other information (arguments) needed to execute the file and other special symbols recognized by the interpreter. Most BOSS/IX commands are executable files, as are the utility programs and several special applications programs.

To enter a command, type the name of the command and any additional information required, and then press <RETURN>. The command and arguments you type is called the "command line". <RETURN> is the command line "terminator".

Some processes executed by the command interpreter take an indefinite amount of time to complete. This is the case, for instance, when the BASIC interpreter is the process evoked. In this case, the command interpreter interprets the command "basic", plus any parameters given with it, and requests that a Business BASIC process be started on the terminal.

It then waits until you are finished with BASIC. The command interpreter prompt isn't displayed again until you are finished with BASIC.

While you are typing in the information, it is stored in a buffer. Only when you have pressed <RETURN> does the command interpreter read what you have typed and attempt to execute it.

### Control Characters

Although they are not part of the command language, control characters exercise some control over the command interpreter by affecting what is happening at the terminal. They are transmitted when the control key, <CTRL>, is held down while some other key is pressed.

Control characters are used either to correct typing mistakes, to stop and resume output or to stop and suspend a running process.

<CTRL>+<C>	This command "kills", or unconditionally ends, the command being run. The command interpreter prompt is then displayed.
<CTRL>+<D>	This command indicates either the end of the line or the end of the file, depending on the context.
	If anything has been typed on the line prior to the command, it means "End-Of-Line". That line is sent to the system, and the cursor moves down to the next line on the screen. The command interpreter prompt is not displayed at this time.

When the command is typed at the beginning of a line, it means "End-Of-File". If there have been lines of data entered since the command interpreter prompt was last displayed, it signals the end of standard input (data input from the keyboard). If typed alone at the command interpreter prompt, it ends your session with the command interpreter.

〈CTRL〉+〈Q〉	The command cancels the effect of the 〈CTRL〉+〈S〉 command (below), continuing the screen display.
〈CTRL〉+〈R〉	This command redisplays the current input line. It is useful when input and output are mixed on the screen.
〈CTRL〉+〈S〉	This command suspends output to the terminal, freezing the display information. If a program is generating output rapidly, this command can be used to stop it while you examine a screen display.
〈CTRL〉+〈Y〉	This command suspends execution of a command. The execution can be resumed using the "resume" command.

## **THE COMMAND LANGUAGE**

The command language has a specific vocabulary and syntax. The vocabulary consists of the following elements:

- o Names of executable files
  - Commands (described later in this section)
  - Application Programs
- o Redirection symbols
- o Pattern Matching symbols

The specific syntax associated with the various commands is given in the description of each command later in this section. Any specific syntax requirements for execution of an application program are described in the documentation for that application.

## General Syntax

The general format for issuing a command in the command language is:

commandname args redir

where commandname is the name of the executable file (command, program, etc.), args is a list of arguments (options, parameters, file names, etc.) and redir is a process redirection expression.

## Argument Syntax

Commands frequently take three kinds of arguments which we will call "dash" arguments, "equals" arguments and unmarked arguments. Dash arguments are preceded with a "-". Equals arguments consist of an argument name, followed by "=", followed by an argument value, with no spaces between these parts. Unmarked arguments are usually names of files to be processed by the command.

Generally, the order in which arguments are placed following a command is unimportant. There are exceptions, however, such as in these cases:

- o Since the "write" command accepts an entire line, these characters are not interpreted within the line;
- o Because "command" expects a command argument, all "-" and "=" arguments must immediately follow the command in order to be interpreted correctly.

## **Process Redirection**

As noted above, the command interpreter usually takes input (the user's commands) from the keyboard and sends output (the results of the command) to the terminal screen. The command interpreter also recognizes several symbols that redirect input and output allowing control over where input comes from and where output goes. Processes can be chained together to form complex commands. Processes that require no interaction with the terminal can be run as a background process, thus freeing the terminal for other uses.

### **Input/Output Redirection**

There are three standard channels used for input and output operations, called Standard Input, Standard Output, and Standard Error. Standard Input is the terminal keyboard, where the command interpreter expects data to be entered. Standard Output and Standard Error are the terminal screen, where results of a command and error messages are displayed. All three of these can be redirected using special redirection symbols.

- o Input can be taken from a file or device
- o Output can be written to a file or device
- o Error messages can be written to a file or device
- o Output from one process can be fed as input to another

The redirection symbols are listed and described in Table 5-1.

Table 5-1. Input/Output Redirection Symbols

SYMBOL	DESCRIPTION
<	Takes input from the specified device or file
>	Redirects output to the specified device or file, writing over existing data.
>>	Redirects output to the specified device or file, appending new output to the existing data.
%	Redirects error messages to the specified device or file. Writes over existing data.
%%	Redirects error messages to the specified device or file. Appends new output to existing data.
:	Redirects the output from a process into another process, causing the output from the first process to be the input for the second.

The format for using these redirection symbols is:

source process redirection symbol destination

For example, you can write the directory listing of the root directory to a file with this command line:

@>ls / > /file/name

The file is created if it does not already exist, and the contents of the root directory are written to it. If the file already exists and you want the information appended to the already existing data, use the ">>" redirection symbol.

The ":" symbol, frequently called a "pipe", allows you to string several commands together, taking the output from one command and passing it on to the next as input. For instance, this command line lists out the contents of "/bin" one screen at a time:

```
@>ls /bin | p
```

The output from listing "/bin", instead of being directly displayed on the terminal screen, is piped to the command "p". It is then processed by "p", which paginates the listing as it is displayed on the screen. Without piping the listing through "p", the listing runs by too quickly to be read.

The pipe can be continued through as many processes as you want.

### **Background Processing**

The command interpreter usually waits for completion of the command before it prompts you for another command. In the case of commands that take a long time to process, especially when no user interaction is required, this is inconvenient. To eliminate this inconvenience, the command interpreter can process commands in the "background."

When you execute a command in background, the command interpreter starts the process, echoes the process identification number (pid), and immediately repeats the prompt. You may then continue processing other commands. The background process continues to operate without interaction from your terminal. It is assigned a lower priority number, so may take slightly longer to run, but you are free to do other things.

To execute a process in the background, enter the command line as usual, and type an ampersand, "&", at the end of the line before you press <RETURN>.

When execution is complete, the background process terminates as usual. Completion is not indicated unless some message is included as the last step of the process.

### **Pattern Matching**

The command interpreter recognizes a large variety of pattern matching characters. The pattern matching characters can stand for single characters, groups of characters, or the placement of a pattern on a line.

The pattern matching characters available are summarized in Table 5-2. There is a wide variety of pattern matching characters, so you can create nearly any pattern you need.

Table 5-2. Pattern Matching Characters

Symbol	Pattern Matching Function
*	Matches any number of characters in this position. E.g. "*z" matches "abz", "abcdz", etc.
?	Matches any single character in this position. E.g. a?b" matches "axb", "acb", etc., but not "axxb" or "axcb".
#	Matches any number of occurrences of the character following the pound sign. E.g. "a#zb" matches "azb", "azzzb", etc.
!	Allows alternative matching patterns. E.g., "ab!ac!de" matches "ab" and "ac" and "de".
[c-d]	Used to specify a range of characters, e.g., "a[c-h]b" matches "acb", "adb", "aeb", ... , "ahb", but not "abb" or "aib". The range is determined by ASCII order.
[^xxx]	When followed by a pattern (xxx) and enclosed in brackets, the caret means "not". Anything not matching the pattern following the caret is a match. E.g. "a[^b]c" matches "aac", "acc", "a=c", etc., but not "abc".
^	Matches the beginning of a line. It may be followed by a string to find it at the beginning of a line. E.g., "^s" matches "s" only when it is the first character on a line.

Table 5-2. Pattern Matching Characters  
(cont'd)

\$	Matches the end of a line. It may be preceded by a string to find it at the end of a line. This is used primarily in the text editor, "ved".
\	Used to specify that a pattern matching character or an unprintable character is to be matched.

### **Additional Rules**

There are several rules that **must** be followed when you enter a command for processing by the command interpreter.

#### Case

The BOSS/IX command interpreter is "case sensitive", which means that it recognizes upper- and lower-case alphabetic characters as distinct. All BOSS/IX command names are spelled in lower case letters only, and must be entered in lower case letters only.

#### Punctuation

#### Argument Separators

If a command line contains arguments, the arguments must be separated from the command and from each other by a space. The space serves as a delimiter, telling the command interpreter where the command or argument ends. More spaces are allowed, but one is required.

Redirection symbols do not require spaces either before or after the symbol. They are interpreted as redirecting symbols wherever they occur unless they are "protected" (see below).

### Compound Commands

Compound commands can be formed by following a complete command (command with arguments and redirection as necessary) by a semicolon (;) and another complete command, all on a single command line. Unlike commands strung together with the pipe symbol, commands strung together with the semicolon are processed independently.

### Protecting Command Line Fields

Several characters are recognized by the command interpreter as having special meanings, as described above. These include, but are not restricted to, these characters:

<, >, >, %, !, &, -, =, \*, ?, #, !, ^, \$

Sometimes it is necessary to use these characters without their special meanings. For instance, although it is recommended that these characters not be used in file names, they are allowed.

In such cases it is necessary to indicate to the command interpreter that these symbols are to be taken "literally", i.e. as not having their special meanings. The method used to protect command line fields that use these special characters varies depending on the kind of character and its place in the command line.

## Dash and Equals Signs

The command interpreter recognizes the dash (-) and equals sign (=) as indicating arguments for a command. If an unmarked argument, such as a filename, begins with a dash or includes an equal sign, it will be interpreted as a dash or equals argument and (probably) result in an error.

To indicate that these symbols are to be taken literally, precede the argument with double dashes. For instance, if you have files named "-afile" and "time=mony" to delete, you must protect the dash and equal sign from interpretation:

```
@> delete -- -afile  
@> delete -- time=mony
```

Without the double dashes, an error message would be displayed and the files would not be deleted.

## Redirection Symbols

To prevent the interpretation of redirection symbols, the entire expression containing the symbol must be enclosed in either double or single quotation marks. For example:

```
@> echo hello>test
```

results in a file, named "test", containing the word "hello". To have the literal expression "hello>test" echoed to the terminal screen, you must enter:

```
@> echo "hello>test"
```

## Pattern Matching Characters

Interpretation of a pattern matching character is suppressed by preceding the character with a back-slash (\). For example, to use the "match" command to search for every question mark in a text file, this command line will not work:

```
@> match ? file.txt
```

Since "?" taken as a pattern matching character matches any character, this command line will find every character in the file, not just the question marks. Instead, this command line should be used:

```
@> match \? file.txt
```

## COMMAND FILES

In addition to executing commands as they are entered at the command line, the command interpreter has the ability to open a file and read command lines from it. A file that contains command lines is call a "command file."

## Creating Command Files

Command files are created using the text editor, "ved". Each line of the command file must be an executable command line.

## **Executing Command Files**

There are two basic methods available for executing command files, depending on whether or not the execute attribute is set for the file.

When you create a command file using `ved`, the access modes usually include read and write access only. The file cannot be executed directly, but can be executed by invoking the command interpreter explicitly. To execute the command file, enter this command line:

```
@>command filename
```

The command "command" invokes the command interpreter for execution of a command file. The command interpreter opens the file, and then executes each of the commands in sequence.

If the file access modes are modified to include the execute attribute, the command file can be executed directly by entering the file name at the command interpreter prompt. (Use the "filenames" command to add the execute attribute.) In this way, a command file can be executed just like any executable file.

## **Variables in Command Files**

The command interpreter recognizes a certain notation as representing variables. The command interpreter accepts file names and command parameters as values for these variables when a command is being executed. The acceptable variable names are:

```
$1, $2, $3, ... , $n, $r1, $r2, $r3, ... , $rn
```

The variables \$1, \$2, \$3, ... take single files, parameters or options as their values. \$1 always takes the first file listed, \$2 always take the second, \$3 the third, and so on.

The variables \$r1, \$r2, \$r3, ... each take the file names from the number indicated to the end of the list.

### POUND-SIGN COMMANDS

The command interpreter recognizes several special commands beginning with the pound-sign (#). These commands control the environment of the command interpreter by conditioning the way it interprets the command line and displays responses.

The commands are summarized in Table 5-3. A thorough description of each command, including command formats, follows the table.

Table 5-3. Pound-Sign Commands

COMMAND	MEANING
#a	adds a new abbreviation usable anywhere in a command string
#b	adds a new abbreviation usable at the beginning of a command string
#d	deletes an abbreviation
#e	repeats the last executed command
#h	displays pound-sign command help file
#?	displays pound-sign command help file

Table 5-3. Pound-Sign Commands (cont'd)

COMMAND	MEANING
#l	lists abbreviations
#p	pushes an abbreviation in the stack
#q	exits the command interpreter
#s	sets the default table for # commands
#x	displays and sets the environment
#v	displays and sets verbose mode
#!	executes the command taking the rest of the file as input

#### **Formats and Uses of Pound-sign Commands**

The formats of the pound-sign commands and descriptions of their functions are given in the following paragraphs. First, however, a few points about the commands in general must be noted.

#### Limitations

The pound-sign commands are all internal to the command interpreter itself, unlike most of the commands discussed in the remainder of this section. As a result, they do not have the full functionality of regular commands. The restrictions are:

- o Pattern matching characters are not used as pattern matchers, but are taken literally
- o The "#e" command does not repeat a preceding pound-sign command

## Global and Local Abbreviations

Abbreviations are distinguished as "local" or "global." Global abbreviations are kept in the file ".globals" in your home directory. Whenever you create a global abbreviation, it is stored in this file in your home directory, regardless of your working directory when created. With one exception, global abbreviations are available only to you, and can be used whatever your working directory is at the time. The exception is that, if another user has the same home directory as you do, then you share global abbreviations.

Local abbreviations are kept in the file ".locals" in the directory that was your working directory when you created it. They are available to you only when your working directory is set to the same directory as when they were created. They are also available to any user whose working directory is set to that directory.

Unless specified otherwise, abbreviations usually become global by default. The default can be changed to local with the "#s" command.

**#a**

## Format

```
@>#a{g|l} name expression
```

## Description

The "#a" command defines an abbreviation that can occur anywhere in a command string. For instance, if you often change your working directory to a directory with a long path name, you can define an abbreviation for that path name.

The options specify either a global ("g") or a local ("l") abbreviation. The default is to global unless it has been changed.

If another abbreviation with the same name already exists, it is replaced. This is true regardless of whether the earlier abbreviation was created using "#a" or "#b".

## Examples

1. @>#a jy install/doc/work/july

This command line sets the definition of "jy" as a global abbreviation. The abbreviation can be used to change your working directory to include this as part of the path as follows:

```
@>cd /usr/john/jy/manuals
```

Your working directory is now:

```
/usr/john/install/doc/work/july/manuals
```

2. @>#al jy install/doc/work/july

This command line sets the same definition, but as a local abbreviation.

Notes

"#a" differs from "#b" in that the abbreviation defined by "#a" can occur anywhere in the command string, whereas the abbreviation defined by "#b" can only occur at the beginning of a command string.

## #b

### Format

```
@>#b{g|l} name expression
```

### Description

The "#b" command defines an abbreviation that can occur only at the beginning of a command string. For instance, to run MAI Office, the command line required to start it can be reduced to very few letters, e.g., "maio".

The options specify either a global (g) or a local (l) abbreviation. The default is to global unless it has been changed.

If another abbreviation with the same name already exists, it is replaced. This is true regardless of whether the earlier abbreviation was created using "#a" or "#b".

### Examples

1. @>#b maio cd /oms;basic lib=/lib/bftf.lib -q  
pgm=MS -nr; cd

This command line sets the definition of "maio" as a global abbreviation. MAI Office can now be executed simply by entering "maio".

2. @>#bl maio cd /oms;basic lib=/lib/bftf.lib -q  
pgm=MS -nr; cd

This command line sets the same definition as a local abbreviation.

## Notes

"#b" differs from "#a" in that, whereas abbreviation defined by "#b" can only occur at the beginning of a command string, an abbreviation defined by "#a" can occur anywhere in the command string.

## **#d**

### Format

```
@>#d{g:l} name
```

### Description

This command deletes the abbreviation specified.

The options specify either a global ("g") or a local ("l") abbreviation. The default is to global unless it has been changed.

If the specified abbreviation is stacked (see "#p") only the top abbreviation in the stack is deleted.

### Examples

1. @>#d jy

This command line deletes the global abbreviation "jy" created earlier.

2. @>#dl jy

This command line deletes the local abbreviation "jy" created earlier.

**#e**

Format

@>#e

Description

This command repeats the last command you executed. It does not repeat a previous pound-sign command.

Examples

1. @>date  
Wed May 9 1984 14:47:54

@>#e  
Wed May 9 1984 14:48:03

The first command line displays the current time and date. The second command line, "#e", repeats the first.

**#h or #?**

Format

@>#h

or

@>#?

Description

This command displays the complete help file for the pound sign commands.

Example

@>#h

The list of pound-sign commands is displayed with brief descriptions.

**#1**

**Format**

**@>#1{g|l} {name}**

**Description**

This command displays the specified abbreviation. If no abbreviation name is specified, all abbreviations are displayed.

The options specify either a global ("g") or a local ("l") abbreviation. The default is to global unless it has been changed.

**Examples**

1. **@>#1**

This command line displays all abbreviations in the default table, usually the global table.

2. **@>#1l now**  
**#b now date;who;ps;pwd**

This command line displays the local abbreviation "now."

**#p**

### Format

@>#p{g|l}{a|b} name expression

### Description

This command allows you to temporarily redefine your abbreviations. It does this by handling abbreviations in a "stack", as if one abbreviation is stacked on top of another. If you add an abbreviation with the same name as an existing abbreviation using "#p", the stack is pushed down and the new abbreviation temporarily takes its place. The "#a" and "#b" commands would permanently replace the old definition with the new one.

The options specify either a global ("g") or a local ("l") abbreviation. The default is to global unless it has been changed.

The pushed abbreviation can further be specified as a "#a" or a "#b" type abbreviation. The default is to "#a" type.

The "#d" command deletes the top definition in a stack.

Example

```
@>#1
#b mt mount /dev/fd0 /mnt
#a jy doc/work/july

@>#pb mt mount /dev/boot /mnt

@>#1
#b mt mount /dev/boot /mnt
#a jy install/doc/work/july
```

In this example, all abbreviations are listed first, and then the original abbreviation "mt" is temporarily replaced.

#q

### Format

@>#q

### Description

This command exits the command interpreter. It has the same effect as <CTRL>+<D>. When you exit the command interpreter, you return to the process that called it or log off, depending on the context of the command.

### Examples

>!command

@>#q

>

The command interpreter was called from BASIC.  
The "#q" command exits the command interpreter, returning to BASIC.

**#s**

### Format

@>#s{g|l}

### Description

This command sets the default value to global or local for pound-sign commands that use the default. The default is to the current default. The default definition lasts until you logoff, at which time it reverts to global.

### Examples

@>#s{l}

This command line sets the default to "local."

**#x**

### Format

@>#x{d} {NAME{=value}}

### Description

This command displays and sets environmental parameters. The 'standard environment parameters are:

- PATH - the directory paths searched by the command interpreter when looking for the initial command to execute
- HOME - your home directory
- TERM - your default terminal type

Changes made to the environment last only for the duration of the log on session. The environment may be changed at logon by including this command in the user's ".init" file.

### Examples

1. @>#x  
PATH=:/bin:/util:/usr/bruce  
HOME=/usr/bruce  
TERM=evdt

The command alone displays the current values of the environment parameters.

2. @>#x PATH=::/bin:/util:/usr/bruce:/usr/dennis

This command line adds another directory as a search path sequence. All previous paths must be repeated or they are deleted from the search path options. Repeat the command of example (1) to verify the change.

3. @>#xd DBPATH

This command line deletes the environment parameter line "DBPATH" from the environment.

#### Notes

The three parameters shown in the description are only examples. These three are used in the standard setup of operators on BOSS/IX systems. Another, optional, parameter that is recognized by the O.S. is "PROMPT=", which defines your command interpreter prompt. Others may also be recognized by applications software, such as "DBPATH=", which is used by some data base management systems.

## #v

### Format

```
@>#v {on|off}
```

### Description

This command displays the current status of verbose mode (on or off), or sets verbose mode as specified. The default value has verbose mode turned off.

With verbose mode set to "on," all commands are displayed on the screen before they are executed. This includes each command in any command files you execute.

### Examples

```
@>#v  
Verbose off.
```

```
@>#v on
```

```
@>#v  
Verbose on.
```

In this example, the "#v" command is first used alone to display the current status of verbose mode. Verbose mode is then turned on. Finally, "#v" is used alone again to verify the status.

**#!**

### Format

`@>#! command {arguments}`

### Description

The "#!" command is used to invoke the command interpreter from within a command file.

It can be used, for example, to execute a command file that does not have execute access turned on.

## BOSS/IX COMMANDS

The remainder of this section describes the entire set of BOSS/IX commands included in the base O.S. product. The commands are described in alphabetical order. Full information on the format and argument options for each command are included, as well as a description of the command's function.

Most of the commands are executable files in the directory "/bin". A few, however, reside within the command interpreter program itself, and so are not listed in any directory listing. These commands are:

```
ad
cd (chdir, cwd)
do
prompt
remark
resume
setmask
sync
wait
```

A few more are kept in the "/sys" directory, as noted under the command descriptions.

A summary of the commands is contained in Table 5-4, following which are the command-by-command descriptions. They are grouped by function and listed in alphabetical order.

Table 5-4. Commands Summary by Function

FUNCTION	COMMAND	DESCRIPTION
DIRECTORY FUNCTIONS	ad	adds alternate working directories
	cd	changes working directory
	chown	changes owner of a file or directory
	delete	deletes a directory
	diskusage	displays number of blocks used by a directory
	filemodes	sets access modes for files and directories
	ls	lists contents of a directory
	makedir	creates a directory
	mount	mounts a filesystem onto a directory
	move	moves files
	pwd	prints working directory name
	unmount	unmounts a filesystem

Table 5-4. Commands Summary by Function (cont'd)

FILE FUNCTIONS	addname	adds name to a file
	advance	advances eventcount
	cat	concatenates files
	change	substitutes for a pattern in files
	chown	changes owner of a file
	copy	make a copy of a file
	debe	write by byte
	delete	deletes a file
	diff	displays lines differing between files
	filemodes	sets access modes on files and directories
	iconf	install configuration file
	kychk	repairs keyed files
	login	log on program
	makedev	makes a device file
	makeec	makes an eventcount file
	makesta	creates remote station
	match	finds occurrences of a pattern in files
	move	moves a file
	p	paginates text characters
FILESYSTEM FUNCTIONS	pmask	prints the current file creation mode mask
	pr	paginates file
	setmask	sets file modes mask
	sort	sorts lines in a file
	ved	text editor
	makefs	creates a filesystem
	mount	mounts a directory system
FILESYSTEM FUNCTIONS	space	displays amount of free space in a filesystem
	unmount	unmounts a filesystem

Table 5-4, Commands Summary by Function (cont'd)

PRINTER FUNCTIONS	lpaint	:maintenance program for :printer queues
	lpq	:displays status of print :queue
	lpr	:submits a print job
	lpstat	:displays printer status
	pted	:maintains printer tables
TERMINAL FUNCTIONS	makettymntbl	:makes a terminal :mnemonics table
	makettyxlate	:makes a terminal :translation table
	ttymntbl	:sets terminal mnemonic :table
	ttymodes	:modifies terminal display :modes
	ttyxlate	:sets terminal translation :table
TAPE FUNCTIONS	mcscompare	:compares files on mcs
	mcslabel	:labels an mcs tape
	mcslist	:lists contents of an mcs :tape
	mcsrestore	:restores files from mcs :tape
	mcssave	:saves files to mcs tape
	mtscompare	:compares files on 1/2 :inch mts tape
	mtslabel	:labels a 1/2 inch mts :tape
	mtslist	:lists the contents of a :1/2 inch mts tape
	mtsrestore	:restores files from 1/2 :inch mts tape
	mtssave	:saves files to 1/2 inch :mts tape

Table 5-4. Commands Summary by Function (cont'd)

SYSTEM FUNCTIONS	date	shows the system date and time
	devfmt	formats a device
	errlog	initializes or displays an error log file
	fschk	checks and repairs filesystem structure
	install	software install program
	install key	installs a public key
	kill	forcibly terminates a process
	makeec	makes an eventcount file
	makefs	creates a filesystem
	makesta	creates remote station
	message	allows or prevents sending messages
	ps	displays process status
	resume	resumes execution of a suspended process
	shutdown	shuts down the system to single user mode
	suspend	suspend execution
	sync	updates disk with information from memory
	sysinfo	displays OS level and configuration
	usb	update super block
	vconf	maintain configuration file
	who	displays users currently logged in

Table 5-4. Commands Summary by Function (cont'd)

MISC.	admin	become the system
COMMANDS		administrator
	advance	advances eventcounts
	command	BOSS/IX command
		interpreter
	echo	displays arguments
	exec	executes a program
	prompt	prompts for a character
		before proceeding
	remark	introduces a comment line
	wait	waits for process to exit
	write	sends a message to a
		logged in user

## ad

### Format

```
@>ad dir1 ... dir8
```

### Description

The "ad" command changes the alternate search directories to those directories included in the argument list. The current working directory, along with dir1 ... dir8 are then searched for a given file name or command.

At least one directory must be included in the argument list, although it may be the "null" directory, signified by "". In this case the directory list is cleared. Not more than eight directories may be included.

### Examples

1. @>ad /usr/john /etc

This command line changes the alternate search directories list to include only "/usr/john" and "/etc".

2. @>ad /usr/\*

If there are no more than eight directories in "/usr", they are all added to the search path. If there are more than eight directories, or if there is a file that is not a directory, an error message is displayed.

3. @>ad ""

This command line deletes all directories from the alternate directories list.

Notes

Pattern matching characters may be used in specifying the directory list. All files matching the pattern must be directories, and they must meet the restrictions mentioned above, or an error is generated.

The "-a" option to the "pwd" command lists all alternate directories.

## **addname**

### Format

```
@>addname newname oldname
```

### Description

The "addname" command creates a "link" to a file, which is a directory entry referring to a file. The same file can have several links to it. There is no way to distinguish a link to a file from its original entry.

By using addname, a single file can be referred to by several different directory path names without creating multiple copies of the same file.

A link cannot be formed across directory systems. For example, addname cannot form a link between a file in the root directory system and a mounted directory system.

When you delete one of a group of linked names, only the link is deleted. The file itself is not deleted until the last link has been deleted.

When you write to a file with multiple names, all of the linked files reflect the new contents.

### Example

```
@>addname name2 file
```

This command line results in two different directory path names leading to the same file.

## Notes

Links should be created only on a temporary basis, for a specific procedure, and then deleted.

Although addname cannot create links across filesystems, it can create links between files in different directories within the same filesystem.

The copy command can be used to make copies of files in a different filesystem. The copy command creates a real copy of the file rather than creating a link. The two files are handled independently by the system.

Most programs that handle multiple files do not recognize linked files. In addition, the commands "change", "ved" and "copy" with the "-r" option, cause files to become unlinked. Changes made to a file using these commands are not made in the linked file(s).

The maximum number of names that any single file can possess is 127. The ls command, when executed with the "-links" option, shows how many names a file has. This is the number in parentheses before the owner of the file.

**admin**

**Format**

```
@>admin  
Password:
```

**Description**

The "admin" command changes your access privileges to those of the system administrator. When you execute admin, the command interpreter prompts you for the password, if there is one. You must enter the correct password before you will be given system administrator privileges.

**Notes**

Certain commands can only be executed by a system administrator. This command makes it more convenient for a user to gain system administrator privileges without going through the full log off and log on process.

When you enter <CTRL> + <D> after using "admin," your own command interpreter prompt is displayed and your usual access privileges are returned.

**Files Used**

/etc/passwd      To verify the system administrator's password.

## **advance**

### Format

```
@>advance ecname1 ... ecnamen
```

### Description

The "advance" command advances the first part of the specified event count files by one and displays the new event count value.

An event count is a special file type that simply keeps track of the number of times it has been advanced. In some circumstances, event counts can be used to perform the function of incrementing loops in programs.

### Examples

1. @>advance /usr/test.ec  
/usr/test.ec .1:4 .2:0
2. @>advance /usr/test.ec  
/usr/test.ec .1:5 .2:0

Each occurrence of the command increases the specified event count by one.

## **cat**

### Format

`@>cat file1 ... filen`

### Description

The "cat" command opens each of the specified files and reads all the characters found in each. It then writes the characters to standard output. If several files are specified, the output is the concatenation of the files in the specified sequence.

If "cat" is used with no arguments, it reads from standard input and writes to standard output.

Because "cat", unlike "p", simply reads the characters in the files without interpreting or filtering them, it can be used to merge several files.

The fact that "cat" does not interpret the characters causes it to give erratic displays if used on non-ascii files displayed to standard output.

### Examples

1. `@>cat file`

Displays the contents of "file" on the terminal screen.

2. `@>cat file1 file2 > file3`

Merges file1 and file2, creating file3

Notes

Avoid constructs such as:

```
@>cat file1 file2 file3 > file3
```

This command line will cause the disk to be filled to capacity unless it is terminated.

**cd**

**Format**

@>cd { /directory/path/name }

or

@>cd {partial/path/name}

Also:

@>cwd { /directory/path/name }

@>chdir { /directory/path/name }

**Description**

The "cd" command changes your working directory to the specified directory. You can specify either the full path name or the partial path name relative to your working directory. You must have execute permission to a directory in order to make it your working directory.

If no directory path name is specified, your working directory is changed to your home directory.

All directories in the directory path name must exist. The last name is the path name must be a directory.

## Examples

1. @>cd /usr/john/work

Whatever your current working directory is, this changes your working directory to "/usr/john/work".

2. @>cd work

If your current working directory is "/usr/john," this changes your working directory to a "/usr/john/work." When a partial directory pathname is specified, your current working directory is prefixed to the specified directory.

## Notes

".." may be used to designate the parent directory of your current working directory.

## **change**

### **Format**

```
@>change {-v} pattern substitution {file1 ...  
      filen}
```

### **Description**

The "change" command substitutes the substitution string for the pattern string in the specified files.

All pattern matching characters are recognized by "change". If a pattern matching character is to be taken literally, it must be preceded by a backslash, "\". When pattern matching characters are used in the pattern string, enclose the entire string in quotation marks.

The ampersand, "&", is used in the substitution string to include what was matched in the substitution. Multiple occurrences are allowed.

The substitutions are made in place. If a file is specified, the changes are made in the file and its name remains the same. No backup ("bak") copy of the file is made.

If no files are specified, change operates on standard input and writes to standard output. Changes are made line by line. Input is ended with <CTRL>+<D>.

Change patterns are matched line by line, so a two-line change should be divided into two single-line patterns.

### Options

-v Verbose. Lines are displayed before they are changed, once per change.

### Examples

1. @>change recieve receive \*.doc

Corrects the spelling of receive in all ".doc" files.

2. @>change "ax" "&b&"

axy

axbaxy

3. @>change "devspin" -- "dev=spin" chap.1

The dashes cause the equal sign to be interpreted literally.

## **chown**

### Format

```
@>chown newowner file1...filen
```

### Description

The "chown" command changes the current owner of file1 through filen to the specified new owner. Only the system administrator can change the ownership of files.

All pattern matching characters are available for specifying the files.

### Example

```
@>chown admin /bin/ps /bin/cpw
```

Changes "/bin/ps" and "/bin/cpw" to be owned by the system administrator.

### Notes

Some commands require ownership by admin to function properly (e.g., login, ps, cpw).

Only the system administrator is allowed to change a file's owner.

### Files Used

/etc/passwd      To look up the user identification number of the new owner.

## **command**

### **Format**

```
command {-c}{-v} {command/file/name} {arg1 ...  
argn}
```

### **Description**

The command "command" invokes the command interpreter. The command interpreter reads and supervises the execution of most command lines. Inherent in this capacity is its ability to recognize abbreviations, special characters; pipes, I/O redirection and detached processes.

Called as a command, the command interpreter can be used to interpret a command file. The command lines in the command file are executed in sequential order. The arguments following the name are passed as the values required by the file for execution.

### **Options**

<b>-c</b>	Indicates that arg1 is itself a command line. arg2 through argn then serve as arguments for arg1. If arg1, the command line, is compound, enclose it in either single or double quotation marks, but don't mix them.
<b>-v</b>	Verbose mode. Echoes each command before it is executed.

## Examples

1. @>command /usr/fred/testfile -rev /usr

Suppose /usr/fred/testfile is a command file containing the one command line: ls \$1 \$2.

This example directs the command interpreter to execute the command file, passing it an option and a directory name as arguments for the variables \$1 and \$2. The execution is exactly as if this command were entered:

@>ls -rev /usr

2. @>command -c "ls \$1 \$2" -rev /usr

This example executes exactly as does the previous example. Instead of executing a command file, the -c option indicates that the first argument is a command line. The command line is complex, so it is enclosed in quotes. The command line contains variables, and the following arguments supply the values.

## Files Used

/etc/passwd	To determine the user's home directory.
.globals	For the user's defined global abbreviations.
.locals	In the current directory for local abbreviations.

## copy

### Format

```
@>copy {options} {/source/file/name  
/target/directory/name}  
  
@>copy {options} {/source/file/name  
/target/file/name}  
  
@>copy {options} {/source/file/namel ...  
/source/file/namen /target/directory/name}
```

### Description

The "copy" command makes a copy of the source file(s). The copy can be placed either in the source directory or in another directory. If a single file is being copied, the name of the copy can be changed by giving the full destination file name rather than only the destination directory name. If a new file is not given, the current file name is kept.

If a source file does not exist or cannot be opened, or if the destination directory cannot be opened, an error message is displayed. If the second file already exists, it is overwritten.

Copying directories, devices, event counts and the like are permitted, but are treated differently by the o.s. Copying a directory just creates the target directory. Copying a device just creates a new raw device file. Copying an eventcount file just creates a new eventcount.

To copy multiple source files to a directory, list the source files on the command line followed by the name of the destination directory. The command interpreter verifies that the last named file is a directory before attempting the copy.

When you copy a file or directory, the owner is changed to your user identification. The exception to this is if you are logged on as the system administrator and use the "setowner" option. In this case the original owner is kept. In either case, the original user access modes are kept.

### Options

- c Contiguous. This option stores the new copy of the file on contiguous blocks.
- i Interactive. This option causes "copy" to prompt:  
From:  
To:  
until you press <RETURN> alone at the from prompt.
- nc Non-contiguous. This option explicitly allows the new copy of the file to be stored on non-contiguous blocks.
- o Over. This option is used only with the "-r" option. If the directory being copied already exists in the new location, a new copy of the directory is not created. Instead, the contents of the source directory are copied over the contents of the destination directory.

- q      Query. Requests confirmation before each file; "Y", "y", "YES", or "yes" to do the copy, anything else to skip the file.
- r      Recursive. If the source file is a directory, "-r" creates the target directory and all subdirectories, and copies the files contained in them. Every directory and file from the source directory to the end of the tree structure is copied.
- setowner      Setowner. This option copies the files keeping the original owner (only the system administrator can use this option).
- v      Verbose. This option reports the result of the copy.

### Examples

1. @> copy current.rec old.rec

This command line creates a copy of "current.rec" in the same directory.

2. @> copy file directory

This command line creates a file with the same name in specified directory.

3. @> copy -i

From:

The command line starts "copy" as an interactive process which prompts you for source and destination files.

4. @> copy -r /usr/fred /usr/fred/old

This command line creates copies of all files in directory "/usr/fred" in directory "/usr/fred/old".

#### NOTES

When copying a directory using the "-r" option, if the target directory already exists, a subdirectory with the same name as the source directory is created and used as the destination for all the files. If the target directory does not exist, "copy" first creates the target directory and then copies all the files to it.

## **cpw**

### Format

```
@>cpw (user)  
Enter current password:
```

### Description

The "cpw" command creates, changes or eliminates your password. Simply type "cpw" and then press <RETURN>. The program prompts you for your old password and then twice for your new password. Your current and new passwords are not echoed on the screen as they are entered.

The system administrator may also change the password for other users by specifying the user name.

A password may be up to eight characters long.

### Example

```
@> cpw  
Enter current password:  
Enter new password:  
Again:
```

### Files Used

/etc/passwd	To check the current password, and then modifies it to contain the new password.
-------------	--

## **date**

### Format

```
@>date {options} {parameter=value}
```

```
@>date hhmm(ss) {mm{/}dd{/}yy}
```

### Description

The command "date" sets and displays the system date and time. Date and time may be displayed in BOSS/IX format (Day-name Month-name Day-number Year-number hh:mm:ss) or in BFS format (hh:mm:ss MM/DD/YY).

The command "date" can also be used to change the time and date display formats.

### Options

time=hhmm(ss)(xx)

Set time in hour, minute, second, AM/PM format. Seconds and AM/PM indicators are optional.

date=xx(/)xx(/)xx

Set date in current BFS format.

**-i**      Interactive. You are prompted for the date and time.

The remaining options only apply to BFS date and time formats.

-mdy	Changes to month/day/year date format.
-dmy	Changes to day/month/year date format.
-ymd	Changes to year/month/day date format.
-24	Changes to 24-hour clock format.
-12	Changes to 12-hour clock format.
sec	Displays seconds.
-nosec	Does not display seconds.
b=xx	Two characters to indicate evening.
b=xx	Two characters to indicate morning.
-default	Sets date and time formats to the system defaults.
-bfs	Prints date and time in BFS format.
-fmt	Displays the current date/time format.
-off	Turns off the calendar chip for date/time functions (MAI 3000 only).

#### Examples

1.      @>date  
             Fri Mar 23 1984 12:32:41

This command line, including no arguments, displays the current system date and time in BOSS/IX format.

2. @>date -bfs  
03/23/84 04:41:59 PM

This command line displays the current system date and time in BFS format.

3. @>date -mdy time=0200 date=03/23/84 a=AM

This command line sets the date format to "mm/dd/yy" and sets the system date and time.

4. @>date -i  
02:42:34 PM, 03/23/84. Update clock:  
hhmmssxx mmddyy

#### Notes

Only the system administrator can change the time, date, or formats.

## **debe**

### Format

```
@>debe {options} {parameter=value{b\k}}
```

### Description

The command "debe" copies the specified input to the specified output, performing the requested conversions. The standard input (terminal keyboard) and standard output (terminal screen) are used by default.

When numbers are specified, they are assumed to specify the number of bytes in the parameters "iskip=", "oskip=", "ibs=", "obs=", "bs=", "ivsize=", "ovsize=", "ivskip=", "ovskip=", "skip=" and "count=". However, "b", meaning "blocks" (512 bytes), and "k" meaning "1024 bytes", can also be used immediately following the value (e.g. oskip=10b bs=1k).

The "debe" command is one of the most powerful commands available, and consequently also has great destructive power. Extreme caution should be exercised when using "debe" any time it is being used to copy data to a device. In these cases, existing data is not respected.

Volumes are created by specifying their size using the "ovsize=" parameter. This is used when the output must be broken into pieces, as when copying a hard disk partition to floppy diskette. When the input file is in volumes, the "ivsize=" parameter must be used.

## Options

- fill** Pads every output record with zeros to the requested size (ovsize=#).
- pg** Displays a dot each time a record is written to indicate progress.
- noerror** Continues transfer despite errors. error messages are displayed on the screen.
- notrunc** Does not truncate an already existing output file, enables copying one file into another.
- swab** Swaps bytes (except for the last byte in odd block sizes and odd transfers due to EOF).
- iskip=** Skips the number of bytes of the input file before beginning transfer.
- oskip=** Skips the number of bytes of the output file before beginning transfer.
- skip=** Combines the functions of "iskip=" and "oskip=#".
- ivskip=** Skips the number of bytes at the beginning of each input volume. The "iskip=" option still works, but only applies to the first volume, where it adds its value to the value of ivskip.
- ovskip=** Skips the number of bytes at the beginning of each output volume. The "oskip=" option still works, but only applies to the first volume, where it adds its value to the value of "ovskip."

**ivsize=** Specifies the maximum length, in bytes, of each input volume. When the end of the volume is reached, "debe" asks for the next volume to be mounted. When you are ready to resume copying, type "yes" and press <RETURN>.

If you type "no" and press <RETURN>, it is treated as an end of file condition, and "debe" writes any buffered data to output and exits.

**ovsize=** Specifies the maximum length, in bytes, of each output volume. At the end of each volume "debe" prints a message requesting that the next volume be mounted. When you are ready to proceed, answer YES.

If you answer NO, "debe" stops and displays statistics showing that more data was read than written.

**ibs=** Input block/buffer size in bytes.

**obs=** Output block/buffer size in bytes.

**bs=** Both input/output block sizes in bytes.

**if=** Input file name.

**of=** Output file name.

**count=** Transfers the specified number of input records or until EOF.

## Examples

1. @>debe if=/usr/file1 of=/usr/file2

This command line copies "/usr/file1" to "/usr/file2". This is not a recommended use for "debe".

2. @>debe if=/dev/boot of=/dev/fd0  
ovsize=1280b -pg

This command line copies the boot partition to floppy diskettes. Since the boot partition is larger than a floppy, the data is broken up into volumes of 1280 blocks each, the size of the floppy. The "-pg" option is used to indicate progress while "debe" is working.

## Notes

When "debe" copies data to a new file, the file is created with read and write access modes only, regardless of the access modes of the source file.

Extremely large buffer sizes ("bs=") that work in the single volume case can cause the error message:

debe: cannot allocate storage

if an "ivsize=" or an "ovsize=" is specified that is not a multiple of the buffer size.

When using "debe" with multiple volumes, the following message is printed:

debe: Done with input volume 1. Do you want to continue: If so, mount volume 2 on /dev/rfd0 before answering. Answer yes or no:

## **delete**

### Format

```
@>delete {-i} {-q} {-r} {-s} file1...filen
```

### Description

The "delete" command removes the specified files from a directory. The files may be directories. If a file has multiple names, delete removes only the names included in the list of file names.

### Options

- i      Interactive. You are prompted for the files to delete. If "-i" is used with file names, the interactive option is ignored and the file is deleted.
- Query. You are asked to verify that a file should be deleted. A response of either "yes" or "y" deletes the file; any other response is interpreted as "no", and the file is skipped.
- r      Recursive. When a file to delete is a non-empty directory, delete with the "-r" option recursively deletes all files and then the directory itself.
- s      Silent. Entries are removed without reporting the names of files being deleted.

## Examples

1. @>delete test.j

This command line removes the file test.j

2. @>delete -i  
delete>x.j  
delete>y.j  
delete>.D

This command line prompts you for a file to delete, and continues to prompt for file names until you press either <CTRL>+<D> or <RETURN> alone at a prompt.

## Notes

You must have write permission to the directory containing the files in order to delete them.

## **devfmt**

### Format

```
@>devfmt {arguments} raw device  
name
```

### Description

The "devfmt" command formats the specified "raw" (unbuffered) floppy drive for the number of tracks specified by "n=" starting with the track specified by "s=". If no starting track or number of tracks is specified, default values are used. Specifying a starting track without specifying the number of tracks (or vice versa) is allowed (see examples below).

Verification of tracks formatted can be accomplished in three ways:

1. The "-v" option does a read comparison of the data field.
2. The "-rw" option does a read-write-read comparison of the data field with a different data pattern.
3. The "-a" option, which requires the disk to be formatted before analysis can be done, performs a read comparison of the data field.

If a bad block is found during formatting with verification, the track is reformatted. The verification continues with a check of the same block. If that block fails again, it is recorded as a bad block. Note here that you are able to set the format retries to any positive number you like via the "r=" option. This enables you to reformat a bad track as many times as necessary.

## Options

s= Starting track number. Default is track 0.

n= Number of tracks to format. Default is to the end of the medium.

r= Number of format retries. Default is 1 (-v or -rw must be used).

-v Read only verify.

-rw Read-write-read verify.

-a Analyze format (tracks must be already formatted)

## Examples

1. @>devfmt s=10 n=30 /dev/rfd0

This command line formats the 30 consecutive tracks starting at track number 10.

2. @>devfmt n=3 r=0 -v -a /dev/rfd0

This command line performs a read only verification of the first 3 tracks with no format retries. Those tracks must already have been formatted.

## Notes

Care must be used in specifying the options. Any time the "-a" option is used, no formatting will occur unless a bad block is found and the number of retries is greater than 0.

## **diff**

### Format

```
@>diff file1 file2
```

### Description

The "diff" command shows what lines differ between two string files.

Lines are noted as deleted, created or changed in the order the files are specified. No output is generated if the files are the same.

### Examples

The following examples use these three files:

<u>file1</u>	<u>file2</u>	<u>file3</u>
aa	bb	bb
bb	cc	ce
cc	dd	dd

1. @>diff file1 file2

```
----- 1 line deleted at 0:  
aa
```

```
----- 1 line added at 3:  
dd
```

2. @>diff file2 file3

----- 1 line changed at 1 from:

cc

----- to:

ce

## **diskusage**

### Format

```
@>diskusage {-v} {file1 ... filen}
```

### Description

The "diskusage" command shows the number of blocks on disk that are being used by the specified files or directories. If no files are specified, the disk usage of your current working directory is given.

With the "-v" option, the names of each directory (including normally hidden subdirectories) and file, and the number blocks used by each are displayed. On the final line, the total disk usage is given for all the specified directories.

### Options

**-v**      Verbose. The name of each directory and subdirectory is displayed along with its disk usage.

### Examples

1.      @>diskusage /etc  
11 /etc/maillists  
7 /etc/mailboxes  
14 /etc/ttydesc  
Total blocks: 238

The disk usage in blocks for each subdirectory in "/etc" is displayed, followed by the total disk usage for "/etc."

### Notes

Since diskusage only looks at the size of the file and not at the bit map describing each block, it can only give an approximation of the number of blocks used.

Since it does not take into account filesystem indirect blocks, diskusage underestimates the size of large files. In addition, if there are any linked files in the subdirectories, diskusage cannot accurately reflect the true usage.

## do

### Format

```
@>do "command string"
```

### Description

The "do" command executes a command string. If the command string has one or more arguments, enclose the entire string in single or double quotation marks.

By using variables in the command string, "do" can be useful in defining abbreviations.

### Examples

1.     @>do "diskusage /usr/jws /usr/sdm"  
      10 /usr/jws  
      260 /usr/sdm  
      total blocks: 270

This command line executes the command string with the single command diskusage with two arguments.

2.     @>#b du do "diskusage \$rl"  
      @>du /usr/jws /usr/sdm  
      10 /usr/jws  
      260 /usr/sdm  
      total blocks: 270

This pair of command lines first defines an abbreviation with a variable, and then uses the abbreviation to process the specified files.

## **echo**

### Format

```
@>echo {-nnl} {-nl|-fl} message
```

### Description

The "echo" command displays the specified message. It can be used to display a message to the terminal screen, for instance, when inserted in a command file, to display remarks reflecting the progress of the program.

Pattern matching characters are accepted in the message.

Output is to your terminal unless it is redirected elsewhere.

### Options

**-nnl** This option suppresses a line feed at the end of each message.

**-nl** or **-fl** This option inserts a new line at the end of each message. It is useful for creating filelists.

### Examples

1. @>echo good morning  
good morning

This command line simply displays the specified message.

2. @>echo /bin/d\*  
/bin/date /bin/delete /bin/diskusage  
/bin/dump

This is a simple way to list the names of all the files that match a pattern.

3. In a command file:

```
ls /usr/dir > /usr/dir/list
echo "'/usr/dir' has been listed to
'/usr/dir/list'"
move /usr/dir/list /usr/records
echo "file '/usr/dir/list' has been moved
to '/usr/records'"
```

In this example, "echo" is used to notify the user of the progress of a command file.

## **errlog**

### Format

```
@>errlog {file} {options} {parameters}
```

### Description

The "errlog" command initializes and displays the contents of the system error log file. When a system error occurs, it is logged to the file "/etc/error.log" by "errlog", which is run in the background.

If the "errlog" command is used without specifying a file, options or parameters, the contents of the active error log file are displayed.

### Options

- quiet** Suppresses display of error log.
- initial** Initializes the error log file.  
The file must be specified.
- size=** Sets the maximum number of errors an error log may have. The default is 100. The size can only be specified when initializing an error log.
- errors=** Sets the current starting error number for an error log. The default is 0. The error number can only be specified when initializing an error log.

### Example

```
@>/sys/errlog /etc/error.log -initial  
errors=10
```

This command initializes the system error log, numbering the "error number" field beginning with 10. This allows for keeping track of the number of errors.

### Notes

The "errlog" command is in the "/sys" directory.

## **exec**

### Format

```
@>exec {options} program
```

### Description

The "exec" command starts a program running. The program can be run in foreground or background mode, or on another terminal that is not in use. Input, output and error messages can be redirected.

### Options

All options except "ceil=" and "floor=" may be abbreviated by their first letter.

**alias=** This parameter specifies a name for the process to be used in the "PROGRAM" column of the "ps" display.

**ceil=** **floor=** These parameters specify the upper and lower bounds for the priority of the process. They may be any number between 0 and 9.

**i=** Input. This parameter redirects standard input to take input from the specified device or file. Default is the keyboard.

**o=** Output. This parameter redirects standard output to the specified device or file. When the "-combine" option is used, the output device must be specified. Default is the terminal screen.

e= Error. This parameter redirects standard error message output to the specified device or file. Default is the terminal screen.

t= This parameter specifies the terminal device on which the program is to be run.

-append This option appends output and error messages to the devices specified by "o=outdev" and "e=errdev" without writing over the data already in those files.

-combine This option combines the output and error message devices. The device or file must be specified by the "o=outdev" option.

-detach This option executes the program in background (detached) mode.

-notify This option reports completion of the program with a message. The "-notify" option runs the process in background mode.

-silent This option suppresses printing of the child process number when the "-notify" option is used, and the message displayed when the "wait" option is used.

### Examples

1. @>exec t=/dev/tty2 -detach basic

@>

This command line starts BASIC on terminal tty2. BASIC runs in foreground on tty2 but in background from the host terminal.

2. @>exec i=/dev/null -notify -combine -detach  
o=prm.log program  
223

•  
•  
•  
Process 223 completed due to normal comple-  
tion returning 0.

This command line starts the program  
"program" in background mode. Output and  
error messages are redirected to the file  
"prm.log." Input is taken from the null  
device. The user is notified upon com-  
pletion of the program.

## **filemodes**

### Format

```
@>filemodes {+|-}{rwx} {+|-}{rwx} {-m}
file1...filen
```

### Description

The "filemodes" command establishes the read (r), write (w) and execute (x) access modes of the specified files, permitting access to the owner and to all other users. The first set applies to the owner, the second set to all other users. Only the owner or the administrator can change the modes of a file.

### Options

+ or -      Used with a "r", "w" or "x" to add or delete that mode. If neither is given, the filemodes listed are absolute.

r,w,x      Without + or - gives only that mode.

-m      Changes the modes for other users to be the same as those for the owner.

### Examples

1. @>filemodes rw -w test

This command line changes the modes to allow read and write access only to the owner of test, and to disallow write access to all other users.

## Notes

Options must be specified in the order given. A mode string must be given for both the owner and for all other users. If no access is desired, it can be specified as a null string and represented as a pair of quote characters, either "" or ''.

## **fschk**

### Format

```
@>fschk device name {options}
```

### Description

The "fschk" command checks and optionally fixes the filesystem on the specified device.

An exit code is also returned by "fschk." The exit codes are described in table 5-5.

### Options

The following options can be abbreviated to the first letter.

- i        Inode. If this option is followed by an inode number, the name of the file using that inode is returned.
- b        Block. If this option is followed by a block number, the name of the file using that block is returned.

- fix** Fixes repairable filesystem errors on an unmounted filesystem.
- quiet** Displays no output. This option is useful only when "fschk" is started by a process that uses the exit code returned by "fschk." This option is ignored when the "-u or -utility" option is used.
- utility** Displays output using the utility program message files.

### Examples

1. **@> fschk /dev/fd0**  
Checks for filesystem errors on "/dev/fd0" but does not attempt to fix them.
2. **@> fschk /dev/fd0 -u -f**  
Checks for filesystem errors on "/dev/fd0" and attempts to fix them automatically. Uses the utility program.
3. **@> fschk /dev/boot -i 23**  
Inode 23 is file /bin/makedev. Returns the name of the file on "/dev/boot" using inode 23.

### Notes

If you want more control over the examination or repair of a filesystem, you can use the standard utility program "/util/fsdbg."

A program that spawns "fschk" and uses the "-quiet" option should be able to interpret the following exit return codes and produce appropriate error messages when necessary.

Table 5-5. "fschk" Exit Codes

Exit Code	Description
< 0	System error number that caused "fschk" to terminate prematurely
0	Normal termination with no error
100	User aborted during execution.
201	No device name was specified in command line.
202	Device name incorrectly specified.
203	Error by "fschk" in parsing command line for options.
204	Unknown option found on command line.
205	Error by "fschk" in parsing command line for options.
206	Could not open specified device for reading.
207	Could not open specified device for writing and so could not fix the filesystem on that device (e.g. device may be mounted).
208	Attempt to use "-fix" option by some account other than "admin".
209	User chose to abort after a file i/o error. This should occur only when the "-utility" option is used.
300	Normal termination. The exit code is 300 plus the number of errors detected.

### Files Used

The following files are required when the "-utility" option is specified:

- /util/utmsg.txt
- /util/utmsg.ind
- /util/uthelp.txt
- /util/uthelp.ind
- /util/stderror

## **iconf**

### Format

```
@> iconf {-new} {-save}
```

### Description

The "iconf" command is a special command for installing the system configuration record from the filesystem to a special location on the first fixed disk. If executed without options, "iconf" displays the current configuration record.

The configuration record must be written to the filesystem from the distribution medium. The usual installation procedure (described in the BOSS/IX Technical Reference Manual) writes the configuration record directly to its position on disk, bypassing the need to write it to the filesystem first.

### Options

<b>-new</b>	This option displays the configuration record in the filesystem. The file is "/sys/config.X", where X is the system serial number.
<b>-save</b>	This option writes the configuration record to its position on disk.

### Example

```
@>iconf -save
```

## **install**

### Format

```
@>install device PID1 (PID2 ... PID3) (options)
```

### Description

The "install" command installs software products from diskette, 1/4 inch MCS or 1/2 inch MTS tape release sets. Installation from diskette is specified by entering "fd0" or "/dev/fd0" for the device. For installation from 1/4 inch MCS cartridge type either "cs" or "/dev/cs". For installation from 1/2 inch MTS type either "ts" or "/dev/ts". The Product Identification Code (PID) is a three letter code associated with each software product.

Installation of several software products at one time is allowed only for installation from cartridge. In this case, all the products must be on the same cartridge. For installation from diskette, only one product can be specified in each command line.

### Options

<b>to=</b>	This option allows you to specify a directory path to be prefixed to the files in the software product release set. This allows a product to be installed onto a mounted filesystem.
<b>-query</b>	This option only applies to installation from tape. The release level file for each product being installed is displayed, and you are asked to verify that this is the correct product.

## Examples

1. @>install /dev/fd0 EUT

This command line installs the EUT product, the BOSS/IX utility set, from diskette.

2. @>install fd0 EUT to=/out

This command installs the EUT product from diskette to the directory '/out'.

3. @>install cs EUT EBS EIT EDB EDS

This command line installs the five software products specified from magnetic cartridge steamer. Release level files are displayed before installation, but verification is not requested prior to installation.

## **install\_key**

### **Format**

```
@>install key /absolute/file/path/name
```

### **Description**

The "install key" command opens the specified file and reads the current "public key" from the file. You are then asked to type the new public key or press <RETURN> for no change. If you enter a new public key, you must enter it twice. After being correctly entered twice, the new public key is written to the file.

### **Example**

```
@>install key /etc/level/EUT
Current Public Key: 12345678
```

Input 8-character public key (cr=no  
change):

The command line displays the current public key of the EUT software product and prompts you for the new key.

## **kill**

### Format

```
@>kill (signal=#!-#) pid1 ... pidn
```

### Description

The "kill" command forcefully terminates each of the currently existing processes specified. If the process does not exist, this message is displayed:

kill: process pid does not exist

Only the process owner or the system administrator can kill a process.

Processes that have terminated (EXIT state) are unaffected by "kill". Only a process that waits for the child can clear the entry. A user program that does a fork/exec but does not ever wait for the child might cause this state.

If a process id number of 0 is given, all of the user's processes are terminated and the terminal is logged off.

The "signal=" parameter can only be used if the "/include/signal.h" file is present. It is included in the optional Program Development Package (EDS).

### Parameters

signal=# or -#	This parameter is not used at this time, release 7.2.
-------------------	--

### Example

```
@> kill 104 296
```

This command terminates the two processes, 104 and 296.

## **kychk**

### Format

**@> kychk filename {-data} {-listkeys} {-fix}**

### Description

The "kychk" command checks and optionally repairs the keyed file "filename." Keyed files include the "direct" and "sort" file types.

BOSS/IX keyed files are managed by C-ISAM file organization processes, which are based on b-tree structures.

### Options

All of the options may be abbreviated by their first letter.

**-data** This option specifies that each data record is to be examined.

**-listkeys** This option displays information about each b-tree node along with detailed information about each entry in the node. (See notes.)

**-fix** This option fixes repairable keyed file errors.

## Examples

1. @>kychk test -l

The file "test" is checked for errors, but no attempt is made to repair them. Any errors found are reported.

2. @>kychk test -d -f

The file "test" is checked for errors, including errors in the data record. An attempt is made to repair any error found.

## Notes

The following information is included on each entry in a node if the "-listkeys" option is used:

- o flag value
- o total length
- o key length
- o duplicated number
- o lead count
- o trail count
- o key value

## **login**

### **Format**

```
@>login {name} {-quick} {t=time}
```

### **Description**

The "login" command executes the logon procedure program. When the system goes from single to multi-user mode, it automatically runs this program for every terminal that is specified in the "/etc/ports" file.

To log on the system, type your user name (which is defined in the "/etc/passwd" file) on any terminal that displays the prompt:

Account name:

If the account has a password assigned, the program prompts:

Password:

After a successful log on, the login file is updated and the user is informed of the message of the day.

### **Options**

<b>-quick</b>	Prevents displaying of the message of the day.
<b>t=time</b>	Specifies a time limit in seconds for the user to log on the system. The default is 3 minutes.

### Example

```
@>login fred -quick  
password:
```

Logs Fred onto the system, bypassing the message of the day. Fred has a password assigned, which must be entered at the additional prompt.

### Notes

Entering <CTRL>+<D> logs off the latest login.

## **lpmaint**

### **Format**

```
@>lpmaint {printer} {options}
```

### **Description**

The "lpmaint" command performs three functions:

1. it changes the status of the entries in the printer queue,
2. it fixes any inconsistent printer queue,
3. it signals a printer that a form has been changed.

Any user can execute the third function. The user who submitted a job and the system administrator can execute the first function. Only the system administrator can execute the second function.

### **Options**

Options may be abbreviated by their first letter.

<b>-kill</b>	Terminates the print jobs specified by job request numbers. If the job is in the queue but not printing, it is removed. If the job is printing at the time of the request, it is stopped and then removed.
<b>-stop</b>	Stops the print jobs specified by job request number. The jobs remain in the queue with the status "stopped".

- resume Starts the print jobs specified by job request number. The job status is changed from "stopped" to "waiting". The jobs will be printed in turn. Unless specified otherwise, printing is resumed at the top of the page on which the job was stopped.
- fix Checks the printer queue and returns it to a consistent state. This should not be used while despooling is running. It verifies that the linking between print jobs is correct so that system errors do not occur.
- fc Signals the printer that a new form has been mounted and that printing can be resumed.
- fp Signals the printer to print the new form for alignment purposes. This may only be done when a job is in a form change state.
- priority Changes the priority of the print job to the new priority specified.
- unit Changes the name of printer on which the specified job is to be printed.
- form Changes the form used for the specified print job.
- begin job# page Changes the first page to be printed of the specified print job.
- end job# page# Changes the last page to be printed in the specified print job.

**-copies job# copies**

Changes the number of copies to be printed by the specified print job.

- dn job#** Turns the delete option for the specified print job to "on".
- df job#** Turns the delete option for the specified print job to "off".
- nn job#** Turns the notify option for the specified print job to "on".
- nf job#** Turns the notify option for the specified print job to "off".
- qn job#** Turns the requeue option for the specified print job to "on".
- qf job#** Turns the requeue option for the specified print job to "off".
- rn job#** Turns on the "raw" option for a job. When the job is printed, all mnemonics processing and character translation is bypassed. If this option is set, starting and stopping page numbers are ignored. The file is printed from beginning to end.
- rf job#** Turns off the "raw" option for a job.
- wn job#** Turns on the "wait" option for a job. This causes the despooler to delay printing the file until all processes have closed their access to this file.
- wf job#** Turns off the "wait" option for a job.

**-time time/date**

Delays despooling until the specified time/date. The format options for the date are hhmm, MMDhhmm or MMDDhhmmYY, where hh is hours, mm is minutes, MM is months, DD is days and YY is years.

**-version** Displays the version level of the spooler software.

### Examples

1. @>lpmaint -stop 2 4 -kill 1 5

This command line stops print jobs 2 and 4, and kills jobs 1 and 5

2. @>lpmaint -begin 4 3 -copies 4 5

This command line changes print job 4 so that printing begins on page 3 and 5 copies will be printed.

### Files Used

**/etc/printers**

For the location of the printer queue.

**/etc/ queues/lpq.printer.killev**

For the printer's kill event count file name.

**/etc/ queues/lpq.printer.info**

For the printer's name and job number.

**/etc/-queues/lpq.printer.form**

The form event count file.

# lpq

## Format

```
@>lpq {printer} {-version}
```

## Description

The "lpq" command displays the entries in the queue of the specified printer. If no printer is specified, then all printers in the default queue are displayed. The following information is displayed.

Job	Print job number
Pri	Priority (0 to 9)
State	Current state. The five states are: <ul style="list-style-type: none"><li>- working (currently being despooled)</li><li>- waiting (waiting to be printed)</li><li>- fwait (waiting for a form change)</li><li>- error (an error occurred during printing)</li><li>- stopped (suspended, not printing)</li></ul>
User	User who submitted the print job
File/	The name used to notify the user when the job is completed.
Alias	
Pr	The name of the printer
Form	The name of the form on which the job is to be printed.
#	The number of copies to be printed.
D	The delete option (Y or N) to delete file upon completion of printing.
N	The notify option (Y or N).

R	The requeue option (Y or N).
RP	Restart page. The first page of the file to be printed when restarting.
SP	Stop page. The last page to be printed (0=entire file).
H	Hold priority (0 to 9).

### Options

**-version** Displays the version level of the spooler software.

### Notes

A "working" state indicates that the despooler is attempting to print the job. Printing may be interrupted, however, and the state be "working" if the printer is offline, out of paper or ribbon, in use by redirected output or BASIC, or if admin has manually "killed" the despooler while it was running. To check the reason for a job not printing while the state is "working," use "lpstat."

### Files Used

**/etc/printers**

For the location of the printer queues.

**/etc/ queues/printer.killv**

The print queue kill event count file.

**/etc/ queues/printer.info**

Name and packet number for the kill request.

## **lpr**

### **Format**

**@>lpr {options} file1 ... filen**

### **Description**

The "lpr" command places string files in printer queues for printing. If there are no files given, it reads from standard input into a temporary file, and then places this file into a queue for printing.

### **Options**

All options may be abbreviated by their first letter.

**alias=** Specifies a name (the alias) to be used instead of the file name when notifying the user of completion. This name is displayed as the file name in the "lpq" report.

**class=** Specifies the name of a print job class contained in the file "/etc/class." Print job parameters that are not specified are supplied from the class definition.

**copies=** Specifies the number of copies to be printed.

**form=** Specifies the name of the form on which the job is to be printed.

**list=** Prints the file on the specified printer.

<b>priority=</b>	The print job priority. The priority may be any number from 0 to 9. 9 is the highest priority. Jobs with priority 0 are not printed until the priority is changed to above the hold priority.
<b>-delete</b>	Deletes the file when the job has been completed.
<b>-notify</b>	Notifies the user when the job has been completed.
<b>-off</b>	Turns spooling off. The job is printed directly without going through a print queue.
<b>-raw</b>	When the job is printed, all mnemonics processing and character translation is bypassed. This flag would be used to print a file that was generated by the graphics system.
<b>-requeue</b>	Places the file at the end of the queue with priority 0 after the current print request has been completed.
<b>-spool</b>	This option forces the job to be spooled, overriding the "/etc/defaults" file when it specifies that spooling is off.
<b>time=</b>	This causes the despooler to delay printing of the file until the date specified. The format options are hhmm, MMDDhhmm and MMDDhhmmYY.
<b>-version</b>	Displays the current level of lpr.

- vfu** This option unconditionally sets the electronic vfu (top of form) when spooling is turned off ("off"). Make sure the paper is set to top of form before executing this option.
- wait** Spools the entire file before starting to print the job.

### Examples

1. **@>lpr temp .globals Makefile**

This command line prints the files "temp," ".globals," and "Makefile." The files are submitted to the spooler which then places them in a print queue.

2. **@>lpr -notify alias=expense-report copies=2 expenses**

This command line prints two copies of the file, expenses, and notifies the user upon completion. The notification message refers to the print job as "expense-report."

### Notes

The names of printers are contained in the file, "/etc/printers." The printer queue is the file "/tmp/\_queues/xxx.que," where xxx is the printer name.

## Files Used

**/etc/ qtemp**

A directory containing queued files

**/etc/printers**

To find the location of the printer queue.

**/etc/forms**

To verify form definitions.

**/etc/class**

For the printer class information.

**/etc/defaults**

For system printer default values.

## **lpstat**

### Format

```
@>lpstat {printer} {-version}
```

### Description

The "lpstat" command reports the status of all printers on the system. With the "-version" option, the level number of lpstat is displayed. The information displayed by lpstat is:

Printer	The printer's name.
PID	The process identification number of the printer's despooler.
Job#	Print job number.
Form	The current form installed on the printer.
Copy	Number of the current copy being printed.
Page#	Page number currently being printed.
Status	Current status of the printer. The possible statuses are:  idle - no job being processed printing - working chng form - form change required offline - printer is off line no paper - paper is out mnem err - mnemonics error not init - printer not initialized i/o error - input or output error no ribbon - ribbon is out

### Files Used

/etc/printtab      The printer status file.

## ls

### Format

```
@>ls {options} {file1 ... filen}
```

### Description

The "ls" command lists the file names and additional information about the files specified. If no file names are given, the current directory is assumed. Otherwise for each argument, "ls" determines if it is the name of a directory. If it is a directory, "ls" lists its contents unless the "dir" option was used. If a file was specified, only its name is printed. Additional information can be requested by using the options described below.

### Options

- a All. This option shows all files, including files whose names begin with '.', which are normally considered "hidden" files.
- b BFS. This option displays additional BFS information.
- blks Blocks. This option displays the number of blocks used by the file.
- date This option displays the latest file creation/update date.
- dir Directory. This option lists the specified directory entry instead of the files in it.

- fileno** File number. This option displays the file number, which can be used for debugging purposes.
- l** Long. This option shows the length of the file in bytes, the access permission, the owner, the modification date, and the name of the file. If the file is a directory, a DIR precedes the listing. If the file is a device file, a "U" or "B" (Unbuffered or Buffered) is displayed followed by the major/minor device numbers. If the file is an event count, the word "EVENT" appears. If it is a system event count, the word "EVENT" is followed by the event count number. If the year the file was created or modified is different than the current year, it is included in the listing.
- links** Links. This option shows how many directory path names, or links, a file has. Links are made with the "addname" command.
- modes** This option displays file access modes.
- owner** This option displays the name of the file's owner.
- p** Path name. This option shows full directory path names rather than just the entry name. This only works if the file arguments are full path names.
- r** Recursive. This option lists all directories and files from the specified directory to the end of the directory structure.

<b>-rev</b>	Reverse. This option lists the files in reverse order. When used in conjunction with "-time," displays modification from newest to oldest, otherwise the sort is reverse alphabetical order.
<b>-size</b>	This option lists the size of the file in bytes.
<b>-t</b>	Type sort. This option sorts the listing by file type.
<b>-time</b>	Modification or Creation Date. This option lists the files in order of modification, oldest to newest.

### Examples

1.     @>ls stuff junk trash

This command line lists "stuff," "junk," and "trash" if they are files, or their contents if they are directories. If any don't exist, they are skipped.

2.     @> ls -l work  
          365 rw. rw. john Aug 20 13:40:06 work

This command line displays, in long form, the file "work".

3.     @>ls -time \*.c

This command line lists all files matching the pattern "\*.c" in order of their creation or most recent modification time.

### Notes

You must have read permission to a directory in order to list its contents.

## **makedev**

### Format

```
@>makedev name {u|b} major minor
```

### Description

The "makedev" command creates a device file entry. The first argument is the name of the entry. The second is either "u" for unbuffered (direct access to the device driver) or "b" for buffered (access through the system buffers). The last two numbers specify the major and the minor device types (e.g. unit, drive, or line number). These numbers are assumed to be decimal.

For buffered devices there is usually an associated unbuffered device, called the "raw" device. The major and minor modes for both devices is the same, although they have different names and one is buffered and the other unbuffered.

For example, a listing of your "/dev" directory will probably show both of these device entries:

```
Dev U 14,0    rw. rw. admin Sep 10 11:15:33 rwd0
Dev B 14,0    rw. rw. admin Sep 10 11:15:34 wd0
```

The first entry is for the raw device, and the second is for the buffered device.

Some commands that work on devices can process a raw device more quickly than they can process buffered devices.

## Options

u        Unbuffered

b        Buffered

## Examples

1.        @> makedev /dev/null u 0 0

This command line creates the null device entry, an unbuffered device.

2.        @>makedev /dev/fd0 b 7 0

This command line creates the floppy drive 0 entry.

## Notes

The assignment of major and minor device numbers is specific to your system.

Although device files can exist in any directory, by convention they are in "/dev". Device files can have multiple file names, but it is not recommended.

## **makedir**

### Format

```
makedir directory1 ... directoryn
```

### Description

The "makedir" command creates new directories with the given path names. All directories in each path name except the last one must already exist. The last name in the path name must not exist.

### Examples

1. @> makedir /usr/john/jdoc

This command line creates the subdirectory "jdoc" in the directory "/usr/john." Both directories, "/usr" and "/usr/john," must already exist.

## **makeec**

### Format

```
@>makeec {sys=} ecl...ecn
```

### Description

The command "makeec" makes an eventcount file.

### Parameters

sys=      Creates the specified eventcount files as  
or           system eventcounts. It is maintained for  
s=           the use of system functions rather than  
              for the use of special programs.

### Example

```
@>makeec s=1 /dev/clock
```

This command line makes the system clock entry.

### Notes

Making an eventcount with the same name as an existing file removes the file entry and its value and replaces it with the new eventcount entry.

Currently defined system eventcount numbers are:

- sys=1      System Clock , which is advanced at fixed intervals (1/10 second).
- sys=2      Child exits, which is advanced whenever a child process exits or suspends. The eventcount is advanced once per process.

Only the "/dev/clock" system eventcount is available to users.

Eventcounts can be deleted with the delete command.

## **makefs**

### Format

```
@>makefs device {size} {file=value}
```

### Description

The "makefs" command creates a filesystem on the specified device. This is done by writing the following information at the beginning of the device:

- o the filesystem header
- o file descriptors
- o the bit maps.

The first six blocks of the device are skipped by "makefs". The number of file descriptors is rounded up to fill out a block.

Any block found to contain an earlier filesystem header is erased.

### Options

size	The number of blocks in the device. If size is not specified, the total number of blocks in the partition is used.
file=	This option specifies the maximum number of files to be allowed in the filesystem. If no number is specified, 9/64 of the number of blocks is the number used.

## Examples

```
@>makefs /dev/rfd0 1280
making freemap
making fdmap with 184 fds
making new fds
1248 blocks available, writing header
```

This command line creates a filesystem on the diskette in the diskette drive fd0. There are 184 file descriptors and 1248 blocks available for use.

## Notes

Specifying the "raw" device makes the process slightly faster.

## **makesta**

### Format

```
@>makesta name network# station#
```

### Description

The "makesta" command creates a remote station file in two parts, given a network address consisting of the network address number and the station number.

This is used with the local area network system. Further description of the network addresses can be found in the documentation for that software.

### Example

```
@>makesta /sta/payroll 0 17
```

## **makettymntbl**

### Format

@>makettymntbl terminal name

### Description

The "makettymntbl" command produces a table which is used by the operating system to implement mnemonics on the specified type of terminal. The parameters required by "makettymntbl" are supplied by the user in a file.

The parameter file must be in the "/etc/ttymntbl" directory and must be called "terminal name.mntbl" (the terminal name with ".mntbl" appended). The name of the terminal must also appear in the file "/etc/terminals".

The file must have one line for each function. The function name must be followed by hexadecimal encoded bytes separated by spaces. Any functions not included in the list are assumed to be non-functional on the specified terminal.

The following functions are available. If a function is not present, it is set to null.

FUNCTION	DESCRIPTION
home	cursor home
clear	clear screen
eos	clear to end of screen
eol	clear to end of line
left	cursor left
right	cursor right
up	cursor up
down	cursor down
ic	insert character
dc	delete character
il	insert line
dl	delete line
b_normal	bright normal
b_underline	bright underline
b_blink	bright blink
b_blink_underline	bright blink underline
b_reverse	bright reverse
b_reverse_underline	bright reverse underline
b_reverse_blink	bright reverse blink
b_all	bright reverse, blink and underline
d_normal	dim normal
d_underline	dim underline
d_blink	dim blink
d_blink_underline	dim blink underline
d_reverse	dim reverse
d_reverse_underline	dim reverse underline
d_reverse_blink	dim reverse and blink
d_all	dim reverse, blink and underline
xy	move cursor to x,y
read_cursor	read cursor position
clear_foreground	clear foreground
expanded_print	expanded print
start_protect	start protect
end_protect	end protect
transmit_screen	transmit screen
page_mode	print page

## Example

```
@>makettymntbl /etc/ttymntbl/evdt
```

This command line creates the file "/etc/ttymntbl/evdt". Parameters are read from the file "/etc/ttymntbl/evdt.mntbl.". The file "evdt.mntbl" contains the name of the function and, to the right, the hexadecimal code evoking that function. The first few lines of this file are shown below:

home	1B 5B 48
clear	1B 2A
eos	1B 59
eol	1B 54
left	1B 5B 44
right	1B 5B 43
up	1B 5B 41
down	1B 5B 42
ic	1B 51
dc	1B 57
il	1B 45
dl	1B 52

## Notes

"Delay" inserts about a 50 ms delay after mnemonics which scroll the screen, if set to a non-zero value. This is used for terminals which do not support flow control.

"Start-narrow" begins normal screen width mode.

"Start-wide" begins 132 column mode, which is supported only on the EDT terminal.

## **makettyxlate**

### Format

@>makettyxlate table name

### Description

The "makettyxlate" command is used to create and modify translation tables used by the command "ttyxlate". The translation table is a byte long record, arranged as a 16-by-16 (00 through FF) position matrix, which specifies how each numeric representation is to be translated.

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
01	01	11	21	31	41	51	61	71	81	91	A1	B1	C1	D1	E1	F1
02	02	12	22	32	42	52	62	72	82	92	A2	B2	C2	D2	E2	F2
03	03	13	23	33	43	53	63	73	83	93	A3	B3	C3	D3	E3	F3
04	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
05	05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
06	06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
07	07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
08	08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
09	09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
0A	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
0B	0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
0C	0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
0D	0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
0E	0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE	BE	CE	DE	EE	FE
0F	0F	1F	2F	3F	4F	5F	6F	7F	8F	9F	AF	BF	CF	DF	EF	FF

The hexadecimal code for a character in position XY in the table specifies which character is to be displayed when the code XY is transmitted to the terminal. So, if position 7A (hexadecimal) in the table contains the code 41, the ASCII code for "A", the character "A" will be displayed when the hexadecimal code "7A" is transmitted to the terminal.

When the "makettyxlate" command is executed, the 16 by 16 matrix of hexadecimal numbers is displayed. To change the code at any position, move the cursor to that position and type over the current code. Cursor movement is done with these commands:

<CTRL>+<P>	moves the cursor up one line
<CTRL>+<N>	moves the cursor down one line
<CTRL>+<H>	moves the cursor left one character
<CTRL>+<F>	moves the cursor right one character
<CTRL>+<C>	ends the editing session

When a new table is first created, by executing the "makettyxlate" with a new table name, the positions are filled with codes 00 through FF, so no translation occurs.

Translation does not occur until the "ttxylate" command is executed.

#### Notes

The translation is affected by the selection of terminal modes with the ttymodes command. The operation of mnemonics is affected when translation is in effect. Before using the translation tables, the user should consider mode setting changes resulting from use of the commands ttymodes, ttxylate, makettytbl and makettydesc.

## **match**

### Format

```
@>match {options} pattern {file1 ... filen}
```

### Description

The "match" command searches each line of the named files for all occurrences of the given pattern. If no files are specified, "match" searches lines read from standard input. The pattern can consist of literal characters and pattern matching characters.

All lines that match the pattern are output to standard output.

Only one pattern string can be specified for each match, but there is no limit to the complexity of the string.

### Options

- s** Suppresses the printing of the filenames.
- n** Includes the file line number of each input file.
- not** Prints all lines that do not match.
- v** Verbose.

## Examples

1. @>match printf \*.c

Displays all lines containing "printf" in the ".c" files.

2. ls -l | match admin

Prints only the lines of the directory listing that contain the word "admin".

## Notes

See "change" for changing a pattern in a file.

To take pattern matching characters literally, enclose them in quotation marks to prevent interpretation by the command interpreter.

## **mcscompare**

### Format

```
@>mcscompare {options} {saveset1...savesetn}
```

### Description

The "mcscompare" command compares files on an MCS cartridge to files on disk, and reports files that are common to both media, and files that are different.

### Options

<b>filesystem=</b>	This parameter causes "mcscompare" to open the specified filesystem device rather than the filesystem device in the saveset. This allows the user to compare a filesystem after a restore, when the filesystem restored was one other than the one specified in the saveset file.
<b>ip=</b>	If the files were saved with the ip option then the part of the path name stripped off will have to be given so that the file can be opened on disk.
<b>-all</b>	This option compares all savesets on the cartridge.
<b>-errors</b>	This option specifies that only names of files that miscompare, i.e. that are not the same, are displayed.

- fn      This option specifies that only the file's existence on tape, and not the data, is to be verified.
- query    This option specifies that the user is to be asked before the compare is begun for each saveset.
- saveset1...    This is a list of savesets to compare
- savesetn

Example

@> mcscompare name=backup

This command line compares files in save set backup.

## **mcslabel**

### Format

```
@>mcslabel {options} {parameters}
```

### Description

The "mcslabel" command allows a user to read and write a tape label using the specified magnetic cartridge streamer device.

### Options

- brief**      Specifies a brief presentation of the tape label. On one line, the set name, date of creation, and date last written is displayed.
- ser=**      When any of these options are specified, it causes the new label to be written to tape. The entire tape is erased, the label is written, and the new label id displayed. The maximum length for each of the fields is eight (8) characters.  
**set=**  
**id=**
- query**      Query before writing a tape label. The current label will be displayed, and the user will be asked if this is the desired tape. A response of "y" will allow a new label to be written. "n" will stop the label process.

## Examples

1. @>mcslabel

This command line displays a complete tape label.

2. @>mcslabel set=backup id=source -q

This command line displays the old label and prompts you for verification that the right tape is in the drive. If you verify it, the new label is written.

~~t = mcs erased~~

## **mcslist**

### Format

```
@>mcslist {options} {setname1...setnamen}
```

### Description

The "mcslist" command displays the contents of the specified savesets on the MCS cartridge.

### Options

**-all** This option displays all the savesets on the tape.

**-long** In addition to the file name, file information will be included in the output. As appropriate for each file, information includes:

- o file type
- o number of records
- o record length
- o key length
- o major/minor numbers
- o modes
- o date of last modification

**-query** Before a saveset is displayed, the tape label is displayed. When the query option is selected, the user will be queried to determine if this is the desired tape. If "y" is the response the search and presentation of a saveset will begin. When the response is "n", the process is stopped, unless the -all option is used.

-tension Precede the tape access with a retension of the tape. This should be used when the tape has been shipped, dropped, stored without use for some time, or when media read errors are encountered.

Example

@>mcslist -l

This command line displays the tape label and, in long format, the contents of the first save set on tape.

Notes

When an entire filesystem is backed up, the save set will have only one entry. That entry will be the name of the filesystem. An individual accounting of the contents within that filesystem is not available.

## **mcsrestore**

### Format

```
mcsrestore {options} {file1 ... filen}
```

### Description

The "mcsrestore" command allows a user to transfer files and filesystem images from the MCS cartridge to disk.

Only the system administrator can restore a filesystem.

### Options

before=	Only files found in the saveset with a date earlier than the specified date will be restored.
-contiguous	Files will be restored to disk in contiguous blocks.
dups=	This parameter specifies the action in case the file to be restored is already on disk. The values for this parameter are: <ul style="list-style-type: none"><li>o replace, which replaces the original file with the restored file</li><li>o skip, which skips the restore file and goes on to the next file</li><li>o query, which prompts you for the specific action to take at each occurrence</li></ul>

<b>-ignore</b>	When a continuable tape read error occurs, a message will appear on standard error, but no prompt will be given to continue. The rest of the file will be skipped and the next file will be restored.
<b>-filesystem</b>	This option is used to restore a filesystem image. Both the saveset name and the filesystem name must be specified.
<b>list=</b>	The files in the file list that match those in the saveset will be recovered.
<b>name=</b>	The files from this specific saveset will be restored. When a save set name is not given, the first is used.
<b>-ns</b>	This option restores files with the current user as owner, and default
<b>-query</b>	For each file, the user will be asked whether or not to restore it.
<b>-recursive</b>	When a directory is specified as a file to restore, all subdirectories and files found in the saveset will be restored.
<b>since=</b>	Only files found in the saveset dated later then the specified date will be restored. Date formats are hhmm, MMDDhhmm or MMDDhhmmYY.
<b>sf=</b>	This specifies the "start file." Files in the list are skipped until this file is encountered. This allows for starting to restore files from where the process was interrupted earlier. The full path name must be specified.

ssnum=	This parameter specifies the saveset by position on tape (starting with 1).
-stat	This option causes the tape statistics (total blocks read from tape and total blocks retried) to be displayed at the completion of the command. The "-verbose" option implies this option.
-tension	The tape will be retensioned before files are restored.
to=	Files will be restored into the specified directory instead of the name of the file on tape. This is analogous to the "copy" command.
-verbose	After a file is restored a message is displayed that the file has been restored. This option implies "-stat".
-verify	This option causes "mcscompare" to be run after the files have been restored, to verify the files have been properly restored.

### Examples

1. @> mcsrestore / -i -r -t -v

This command line first retensions the tape. All directories and files are then recursively restored from the directory in the first saveset. Tape read errors are ignored. Finally, the names of files that have been successfully restored are displayed.

2. @> mcsrestore -f /dev/root n=rootbackup

This command line restores the filesystem image /dev/root

3. @> mcsrestore n=saturday "/src/bin" -v

This command line looks for the 'saturday' saveset, restores files found in /src/bin, and lists the names of the files that are recovered.

4. @> mcsrestore dev=/dev/cs "/usr/rich/\*.c" to=/usr/temp -v

This command line restores files from the first save set, from the directory "/usr/rich" to the directory "/usr/temp", and lists the files that are restored.

#### Notes

Either selected files, a save set, or both must be specified. When no files are given, all files in the specified save set are restored.

If pattern matching characters are used as part of the file names in the command line, the entire file name must be in quotes (e.g. "/bi").

## **mcssave**

### Format

```
@>mcssave {options} {file1 ... filen}
```

### Description

The "mcssave" command copies individual files, files found in filelists, and filesystems to the MCS cartridge. Files are written onto tape in the order they are found on the command line, i.e. they are not sorted.

Whenever a backup to MCS is done, a file called a saveset file is created. This file contains the list of all files that will be placed on tape and file information for each. The saveset is written to tape before the files are saved.

Only the system administrator can save filesystems to magnetic cartridge streamer tape.

### Options

<b>-append</b>	Appends a new saveset at the end of data on the tape.
<b>before=</b>	Only those files dated earlier than the specified date will be written to tape. Date takes the format of hhmm or MMDDhhmm or MMDDhhmmYY.
<b>-filesystem</b>	Using the first file as the name of a filesystem, an entire filesystem image will be written to tape.

-ignore	Rather than stopping on continuable file errors and prompting for abort or continue, this option will cause the backup to report file errors and continue with the backup of the next file.
-ip=	This option causes the specified path to be stripped from the absolute path name of each file.
list=	The user may specify a filelist of files to backup.
name=	Each saveset on a tape has a name. It can be up to 64 characters in length (spaces may be used if the name is enclosed in quotation marks). When a name is not given a default name of "operatorname MM/DD hh:mm" will be assigned.
-query	The user is queried before each file is placed into the saveset. Responding with "Y" will cause the file to be placed in the save set. "N" will cause the next file to be given.
sf=	This specifies the "start from" file. Files in the list are skipped until this file is encountered. This allows for starting to save files from where the process was interrupted earlier. The full path name must be specified.

-recur-	When a directory is encountered in the list of files to backup, all subdirectories and files in the directory are also placed in the save set.
since=	Only those files dated after the specified date will be written to tape. Date takes the format of hhmm or MMDDhhmm or MMDDhhmmYY.
-ss	This option specifies that the files are to be written to tape in start/stop mode (tape motion stops after each file).
-stat	This option specifies that the tape statistics (total blocks written, total blocks rewritten) are to be given at the completion of the save. The "-verbose" option implies "-stat".
-tension	This option, used with "-append", causes the tape to be retensioned before writing files.
-verbose	After each file is written to tape, a message will be displayed that the backup of the file completed.
-verify	After all specified files have been saved, this option verifies that the data on tape matches the data on disk.

## Examples

1. @>mcssave /usr/dennis /usr/dave -r -v

This command line recursively backs up all files in the specified directories and outputs the names of the files backed up.

2. @> mcssave -f /dev/root

This command line makes an image backup of the filesystem "/dev/root".

3. @>mcssave /usr/rich/src/\*.c since=12230000

This command line performs a backup of all ".c" files in the specified directory that have changed since midnight, December 23rd.

## **message**

### Format

```
@>message {on|off}
```

### Description

The "message" command inspects, sets or clears the permission bits which determine whether other users can write to the terminal.

If "message" is entered alone, the current state of message permission is displayed. If "message" is entered with either "on" or "off", message permission for that terminal is set accordingly.

### Examples

1. @> message on

The terminal can now receive messages.

2. @> message

Messages are currently allowed

Displays the current message permission for the terminal.

### Notes

The access permission on your terminal entry controls message reception. When you log on the system, you are given ownership of your terminal. Thus, you can change the access modes through the message command. When you first log on, message reception is allowed.

## **mount**

### Format

```
@>mount {-readonly} {-oride}  
{device\directory}
```

### Description

The "mount" command incorporates independent filesystems into the root filesystem. Mounting simply makes an association between the root directory on the device and the directory node in the existing filesystem.

The contents of the mount point directory is inaccessible while another filesystem is mounted to it. Thus the directory should be one that is empty and reserved for this special purpose. The directory "/mnt", for instance, is available for this purpose.

Since the software detects no difference between a mounted filesystem and the "static" one, all normal file handling programs operate as usual on mounted filesystems. You can transfer files between mounted devices with the copy commands; you can change your working directory to any directory on the device. You can do backups (using "copy") to other filesystems, including floppy diskettes.

If you attempt to mount a filesystem that was not properly unmounted, the "mount" command rejects the request unless the "-oride" option is specified. The "-oride" option allows you to mount the filesystem, but you should repair it using the "fschk" command.

Typing mount, without any arguments, displays the currently mounted devices and directories on which they are mounted.

## Options

The options may be abbreviated by their first letter.

- oride      This option permits mounting if the filesystem had been left mounted.
- readonly    Mounts the filesystem with read access only. You can not modify a read-only filesystem. The system will reject any attempt to create, delete, change the owner or change in any way, any file or directory in the filesystem. If a floppy diskette is write protected, this option must be used to mount its filesystem.

## Example

```
@> mount /dev/fd0 /mnt
```

The filesystem on device "/dev/fd0" is mounted onto the directory "/mnt."

## move

### Format

1. @>move {-i} {-r} {-setowner} {-v}

prompts for source and destination file names.

2. @>move {-q} {-r} {-setowner} {-v} source file  
destination file

moves a single file giving it the name  
"destination file".

3. @>move {-q} {-r} {-setowner} {-v} file1 ...  
file2 directory

moves multiple files to the destination directory.

### Description

The "move" command renames a file by changing the source file name to the destination file name. The source file name is deleted and the destination file name is either created or, if it already exists, is overwritten.

If the move crosses filesystem boundaries, then it is implemented as if the following sequence of commands had been given:

@>copy file1 file2; delete file1

If, however, the destination filesystem has no space, the file is not deleted, although the destination entry might be partially created.

If the second argument is a directory, the new file will have the same name in the new directory as in the old.

To move multiple source files to a target directory, list the source files on the command line followed by the name of the target directory. Move verifies that the last named file is a directory before attempting the move, and reports an error if it is not.

### Options

- i           Interactive. Causes move to prompt:
  - From:
  - To:
  - until you <CTRL>+<D> or <RETURN> alone at a prompt.
- q           Query. Request confirmation (YES or NO) before moving each file.
- r           Recursive. If the source file is a directory, the option creates the target directory, and recursively moves all files and subdirectories in the source directory.
- setowner   Sets owner. Attempts to retain the original user id. Only the system administrator can do this.
- v           Verbose. Displays the name of each file as it is moved.

## Examples

1. @> move alpha beta.

This command line changes the name of the file from alpha to beta. The file remains in the same directory.

2. @> move /usr/jeff/games/newgame /games

This command line moves the file newgame from the directory "/usr/jeff/games" to the directory "/games." The name of the file is unchanged.

## Notes

If the new name already exists in the destination directory, move first tries to delete it.

## **mtscompare**

### Format

```
@>mtscompare {options} {saveset1 ... savesetn}
```

### Description

The "mtscompare" command compares files/filesystems on 1/2 inch MTS to the files/filesystems on disk.

### Options

<b>filesystem=</b>	This parameter is used to specify the filesystem to use in the comparison instead of the filesystem specified in the saveset. It is generally only used when a filesystem backup is being restored to a different filesystem.
<b>ip=</b>	This parameter specifies the directory path for the files to be used in the comparison. It is used if the "ip=" option was used during the save.
<b>-all</b>	This option specifies that all savesets on the tape are to be compared.
<b>-errors</b>	This option specifies that only files that fail the comparison are to be displayed.
<b>-fn</b>	This option verifies the existence of the files on tape without verifying the data in the file.

- query      This option specifies that the user is to be prompted before each saveset is compared.
- unload      This option specifies that the tape is to be unloaded upon completion of the comparison.

Example

@>mtscmpare dailybak

Compares files, including data, in the saveset "dailybak" to the files on disk.

## **mtlabel**

### Format

```
mtlabel {options}
```

### Description

The "mtlabel" command reads and writes labels on 1/2 inch MTS tapes.

### Options

set=	When a set name ("set="), identification
id=	name ("id=") or serial number ("ser=")
	is
ser=	specified, a new label is written to the tape. All data on the tape is lost. The maximum length for each field is eight characters. Default entries are given for fields that are not specified.
-brief	This option displays the brief, single line, form of the tape label. The set name, id name, volume number, and date last written are displayed.
-query	This option prompts the user before writing the label. The current label is displayed and the user is asked if this is the correct tape. Answer YES to continue with labeling or NO to stop labeling is requested.

## Examples

1. @>mtslabel

Displays the entire tape label.

2. @>mtslabel set=backup id=source -q

Prompts the user before writing a new label with the specified set and id names.

## **mtslist**

### Format

```
@>mtslist {options} {saveset1 ... savesetn}
```

### Description

The "mtslist" command lists the files in the specified savesets. If no saveset is specified and the "-all" option is not used, only the files in the first saveset on the tape are displayed.

When the saveset is a filesystem, only the name and size of the filesystem is displayed. Individual file names are not available.

### Options

Options may be abbreviated by their first letter.

- all      Displays the files in all savesets on the tape.
- long     Displays information on each file in addition to the file name. As appropriate for each file type, the following information is displayed: file type, number of records, record length, key length, major/minor numbers, modes, and date of last modification.
- query    This option specifies that the user is to be prompted before a saveset is displayed. The tape label is displayed, and the user is prompted before any savesets are displayed. Then each saveset label is displayed and the user is prompted before its contents are displayed.

### Examples

1. @>mtplist -l

Displays the tape label and, in long format,  
the contents of the first saveset on the tape.

2. @>mtplist backup

Displays the contents of the saveset "backup".

### Notes

Since the names of the files are in a saveset file at the beginning of the saveset, some files listed may not actually be on the tape. This would happen if, for instance, a file was deleted by another user after the saveset file was written but before the file was saved, or if the save procedure was aborted.

## **mtsrestore**

### Format

```
@>mtsrestore {options} {file names}
```

### Description

The "mtsrestore" command allows you to restore files and filesystem images from 1/2 inch MTS tapes written by "mtssave". If the saveset name alone is specified, then all files in the saveset are restored. If filenames or a filelist are specified, those files are restored. If no saveset is specified, only the first saveset on the tape is searched for the specified files.

### Options

before=	Only files found in the saveset with a date prior to the specified date are restored. The date can be in the following formats: hhmm, MMDDhhmm, MMDDhhmmYY.
dups=	This parameter can specify either "skip", "replace" or "query". This determines the action taken if the file about to be restored already exists on disk.
list=	Specifies a filelist of file names to be restored.
name=	Specifies the saveset from which files are to be restored.

**sf=** Specifies the "start file", the first file in the saveset file to restore. This is used when a restore was interrupted, to specify where to begin restoring.

**since=** Only files found in the saveset with a date later than the specified date are restored. These date formats are allowed:  
hhmm, MMDDhhmm, MMDDhhmmYY.

**to=** Specifies the directory to which files are to be restored, instead of the directory of the file on tape.

**ssnum=** This option specifies the saveset number, by position on the tape, from which files are to be restored. The first saveset is number 1.

**-contiguous**

Restores files as contiguous type files.

**-filesystem** This option is used to restore a filesystem image, created by using the "-filesystem" option with the "mtssave" command. Both the saveset ("name=") and the filesystem name (immediately following the option) must be specified.

**-ignore** This option specifies that when a continuable error occurs, the restore is to continue without prompting the user. A message is displayed, the remainder of the file is skipped, and restoration continues with the next file.

- ns** This option causes the user to become the owner of the restored files (non-secure).
- query** Prompts the user for each file before restoring it. Answer YES to restore the file or NO to skip it.
- recursive** When a directory is specified as a file to restore, this option causes all files in the directory to be restored.
- stat** After the restore is complete, the total blocks processed from tape and the number of read retries are displayed.
- verbose** This option displays a message after each file is restored stating that it has been restored. The restore statistics ("stat") are displayed when the command is completed.
- verify** This option causes the "mtscmpare" command to be executed when all files have been restored.
- unload** This option causes the tape to be unloaded when the command is completed.

## Examples

1. @>mtsrestore / -i -r -v

Recursively restores all files, starting with the root directory, from the first saveset. All continuable tape read errors are ignored, and each file name is displayed when it is successfully restored.

2. @>mtsrestore n=rootbackup -filesystem /dev/root

Restores the file system image "/dev/root". Note that the filesystem name must immediately follow the "-filesystem" option, and the saveset name must be specified.

3. @>mtsrestore "/usr/rich/\*.doc" to=/usr/temp -v

Restores the ".doc" files in the directory "/usr/rich" from the first saveset to the directory "/usr/temp", and displays the names of the files as each is successfully restored.

## Notes

If pattern matching characters are used as part of the file name in the command line, the entire file name must be in quotes, as shown above.

Only the System Administrator can restore a filesystem image backup.

## **mtssave**

### Format

```
@>mtssave {options} {file names}
```

### Description

The "mtssave" command saves individual files and filesystems to 1/2 inch MTS tape. File names may be specified individually or with a filelist.

When a save is performed, a file called the "saveset file" is created and written to tape. This file contains the names (and file information) of all files that will be saved. For filesystem backups, the saveset file contains the name of the filesystem device that was saved.

The saveset file is written to tape before the actual backup begins. Therefore, it is possible for a file name to be included in the saveset file, but the file not be saved.

### Options

**before=** Only those files dated before the specified date are written to tape. The date can be in the following formats: hhmm, MMDDhhmm, MMDDhhmmYY.

**ip=** "Ignore path" causes the specified path-name to be stripped from the fully qualified pathname of each file.

**list=** This parameter specifies a filelist containing the names of files to be saved.

**name=** This parameter specifies the saveset name, which may be up to 64 characters long (with no spaces). If no name is specified a default name is assigned as "operatorname MM/DD hh:mm".

**sf=** This "start file" option causes the command to skip all files (usually in a filelist) until it encounters the specified file name. The remaining files are then saved. This is used to restart a save if the previous execution aborted for some reason. When using "sf=" for this reason, be sure to use a new tape or the "-append" option to avoid losing the previously saved files.

**since=** Only those files with dates after the specified date are written to tape. The date can be in the following formats: hhmm, MMDDhhmm, MMDDhhmmYY.

**-append** Appends the new saveset to the tape without writing over any existing saveset.

**-filesystem**

Writes an entire filesystem image to tape. The filesystem device must be specified immediately following the option. The filesystem should not be mounted during the save.

**-ignore** This option causes continuable errors to be reported but ignored. Execution is not interrupted, and the save continues with the next file.

- query This option causes the program to prompt the user for each file before it is placed in the saveset. If the user responds with "y", the file is added to the saveset; otherwise it is skipped.
- recursive When a directory is encountered in the list of files to save, all files in the directory are also placed in the saveset.
- stat This option specifies that after the command is complete, the total number of blocks written to tape and the number of write retries is displayed.
- unload This option specifies that the tape is to be unloaded upon completion of the command.
- verbose This option specifies that the name of each file is to be displayed after it has been successfully saved. The save statistics ("-stat") are displayed when the command is completed.

## Examples

1. @>mtssave /usr -r -v

Recursively backs up all files in the directory "/usr" and outputs the names of the files as they are saved.

2. @>mtssave -filesystem /dev/root

Performs a filesystem image backup of the filesystem "/dev/ root".

3. @>mtssave /usr/rich/\*.doc since=12230000

Saves all ".doc" files in the directory "/usr/rich" that have been changed since midnight, December 23rd.

## Notes

Since the saveset file is the first file in a saveset, it is only an estimate of what files are in the saveset. Some files in the file saveset may not be saved if, for instance, "mtssave" aborts before all files have been saved, or a file is deleted by another user after being added to the saveset and before being saved.

Only the System Administrator can save a filesystem image.

# p

## Format

p {options} {file1...filen}

## Description

The "p" command reads its argument files or, if no file names are given, it reads from standard input (the keyboard), and displays them on standard out (the terminal screen). It paginates, stopping at the end of each screen with the name of the file, the percentage of the file so far displayed, and the question "MORE?". The recognized responses to the prompt are as follows:

RESPONSE	MEANING
<Y>, <RETURN>,	Yes. The next screen of text is displayed.
<SPACE>	
<H>	Half. The next half screen of text is displayed.
<Q>	Quarter. The next quarter screen of text is displayed.
<L>	Line. The next line of text is displayed.
1-9	The next 1-9 lines are displayed.
<N>	No. No more of this file is displayed. Go on to the next file.
<S>	Stop. Exits the program.

At the end of file, p moves to the next file, and prompts the user for more.

The "p" command also interprets unprintable characters, such as control characters, by printing "^^" followed by a transformed version of the character, for example "^C". Note that spaces and tabs (which are defined at every eighth column) are treated normally. When lines exceed the specified width, they are wrapped around and continued on the next line of the screen.

### Options

All options can be abbreviated by their first letter.

**length=** Sets the number of lines on a page. The default is 22.

**width=** Sets the number of characters in a line. The default is 79.

**-raw** Raw mode. Control characters are not expanded.

**-silent** No prompts are given to the user. Everything is printed. This is used when the output is being redirected into a file.

## Examples

1. @>p /usr/fred/sample

This command line paginates and displays the file "/usr/fred/sample" on the screen.

2. @>p -silent filename | lpr

This command line prints a "hard" copy of the file "filename" in the current working directory ("!lpr" pipes the file to the "lpr" command for printing). There is no pause or message displayed between pages and control characters are filtered out.

3. @>ls / | p

This command line paginates the listing of the contents of the root directory. Execution of the "ls" command is piped through the "p" command.

## Notes

If you type "p" with no arguments and do not redirect its standard input, "p" will read from standard input. It will appear that the system is not responding. To exit this situation, enter <CTRL>+<D> for EOF.

If the eighth bit of a character is "on" on your terminal, the tilde character, "~", is displayed before the character, e.g., "~C". If the character has the eighth bit "on" and is also considered a control character, then the tilde will precede the escape control character sequence, e.g., ^^C.

## **pmask**

### Format

```
@>pmask
```

### Description

The "pmask" command prints the current file creation mode mask. This mask determines the actual modes of a created file.

### Examples

```
@> pmask
rwx rwx
```

All modes are currently allowed.

### Notes

To change the mask, use the "setmask" command.

## pr

### Format

```
@>pr {options} file1 ... filen
```

### Description

The "pr" command produces a formatted version of one or more files on standard output. The output is separated into pages headed by the date, the name of the file and the page number.

The options apply only to files following the option on the command line, and can be reset between files.

### Options

- b      Brief. Prints one line each of the header and footer. Counting blank lines, only six lines are subtracted from the total page size.
- c=      Places "Copyright (c) year author" in the footer.
- F      Places a formfeed after the file.
- f1=      First line of footer.
- f2=      Second line of footer.
- f3=      Third line of footer.
- h=      Uses the specified argument as the header, instead of the file name. (If you use "pr" as a filter, (e.g. "ls | pr"), it will not have a name unless you use the "h=" option.)

- l= Takes page length to be the number of lines instead of the default 66.
- m Prints all specified files simultaneously in separate columns.
- # Produces numbered column output of each file sequentially. Any number between 1 and 69 can be specified. Columns are produced for the page length ("l=").
- +# Begins printing with page number.
- nb No brief. Prints three lines of the header and footer. Counting blank lines, a total of ten lines are printed. This is the default.
- nh No header or footer.
- s= Separates columns by zero, one or more characters instead of white space. To specify no characters, type s= "". The default character is TAB. Do not use control characters as delimiters.
- w= Takes the page width to be the number instead of the default 132.

### Example

```
@> ls : pr w=80 -5 l=1 -nh
```

Produces a five-column listing of the current working directory's files.

### Notes

When using "pr" to display text on the screen, specify the width as 80, and the length as 24.

If you use arguments that consist of several words, enclose them in double quotes.

## **prompt**

### Format

```
@>prompt {character} {message}
```

### Description

The "prompt" command causes the process to wait until a key is pressed.

The "prompt" command is useful in building command files for which it is basically an "if" statement you can use with the command interpreter. In a command file, "prompt" can be used to wait for a key to be pressed.

If a character is specified, prompt acts like a condition. If the specified character is pressed, execution of the command file ends. If any other character is pressed, the remaining commands are processed.

If "prompt" is used without a character, no message is printed and execution of the command file continues.

If both a character and a message are specified, the character must precede string. The message is displayed if the specified character is pressed. A string is specified only if a character is specified also.

## Examples

### 1. In a command file:

•  
•  
•

```
echo -nnl "Hit the ESC key to exit, any other  
key to continue"  
prompt ^l "Exiting now..."
```

•  
•

If the ESC key is pressed, the message "Exiting now..." is displayed and the command file is exited. If any other key is pressed, the command file continues execution. (See "ved" for instructions of entering unprintable characters in a file.)

## ps

### Format

```
@>ps {options}
```

### Description

The "ps" command displays information about the currently active processes. The following information is given:

PID	The process identification
USER	The name of the user who initiated the process
FLG	A three character code indicating the state of the process. The codes are in hexadecimal, and may be combined. The codes for each position are as follows:  8-- uninterruptable wait 4-- three segment storage format 2-- defer signals and eventcalls 1-- locked in memory by user -8- being traced -4- being debugged -2- interruptable wait -1- locked in memory by kernel --8 raw i/o --4 kernel process --1 swapped out

STAT	The state of the process:
WAIT	process waiting for an event
RUN	process executing
RDY	process waiting in the run queue
STOP	process stopped
SSPD	process suspended
EXIT	process has exited
PR	Process Priority (0-8):
0	background processes
1 - 5	interactive processes
6 - 8	system processes and interactive processes holding system resources
WAITFOR	What the process is waiting for. Recognizes file locks as wait events. (Prints "lock".) Locking changes the size of file descriptors.
Kb	The size in Kbytes of the core image of the process
SYS	The amount of system time used in seconds and tenths
USER	The amount of user time used in seconds and tenths
PARENT	The address and PID of the parent process, if any (-long)
PROGRAM	Program name with its arguments, "-" means command interpreter, "=" means the user has executed "admin".

text	Beginning and ending physical address of the text segment (-loc)
data	Beginning and ending physical address of the data segment (-loc)
stack	Beginning and ending physical address of the stack segment (-loc)
MAX	Ceiling (maximum) priority (-long)
MIN	Floor (minimum) priority (-long)
CHILD	Child system time in seconds and tenths (-long)
TIMES	Child user time in seconds and tenths (-long)
AGE	Time in seconds and tenths that a process has been on its present queue (-long)
LU	Number of open logical units (-long)
INT	Outstanding process interrupts (-long)
	80 The process has completed its time quantum
	40 The process will exit
	20 The process will suspend
	10 The process will be preempted
	8 Flag used by kernel tracing routines
SIG	Outstanding signals. BOSS/IX signal numbers correspond to bits in this word (-long)

If no user IDs are specified, "ps" prints the process status of the user. If IDs are specified, "ps" prints the status of those users.

### Options

-all	Prints out the status of all the programs on the system.
-loc	Prints the beginning and ending locations in memory of the text, stack and data sections. Prints the addresses in hexadecimal.
-long	Prints out the long form, all information including the complete command.
pid=	Prints the process status of a specified process. It will only accept one process id.
-pri	Prints the floor and ceiling priorities as well as the current run priority.
-sys	Prints only the system processes (kernel processes). This does not list the caller's processes by default.
user ID	Prints the process status of the specified user(s)

### Example

```
@>ps -a
```

This lists the general process information for all users currently logged on.

## **pted**

### Format

@>pted

### Description

The "pted" command is the printer translation file editor. It allows a way of viewing and changing the printer translation tables. These tables are stored in the file "/etc/ptrans." Table names consist of two alphanumeric characters, a-z, A-Z, 0-9.

Upon execution, the command displays a menu of printer table maintenance functions.

1. Display a Table
2. Edit a Table
3. Dump a Table
4. Create a Table
5. Delete a Table
6. List Table Names
7. Update Modifier Characters
8. Print to a File

The use of these menu options for printer table maintenance is described in the BOSS/IX Technical Reference Manual.

## **pwd**

### Format

```
@>pwd {-all}
```

### Description

The "pwd" command prints the full directory path name of your current working directory.

### Options

all        Print both current and alternate search directories. The option may be abbreviated as "-a".

### Example

```
@> pwd  
/usr/john/doc
```

"/usr/john/doc" is the name of the current directory.

### Notes

Alternate directories are added with the "ad" command.

## **remark**

### Format

@>remark

### Description

The "remark" command tells the command interpreter not to process the arguments. This command is useful for documentation remarks in command files.

### Examples

@> remark "Hello world"

## **resume**

### **Format**

**@>resume pid#**

### **Description**

The "resume" command continues execution of a process which has been suspended, either intentionally, with the "suspend" command, or through an error in the program.

### **Notes**

Resuming a process from the command level causes the command interpreter to wait for it to complete.

Resuming a process does not resume its suspended child processes.

## **setmask**

### Format

```
@>setmask {+|-}{rwx} {+|-}{rwx}
```

### Description

The "setmask" command sets the file creation mode mask. This mask is used to determine access modes whenever a process creates a file.

### Options

+ or -	When used with one or more of "r", "w" and "x", adds that mode to (+) or subtracts the mode from (-) the current file access modes mask.
r,w,x	Indicate read access (r), write access (w) and execute access (x). Without + or -, only the specified modes are assigned.

### Examples

```
@> setmask rwx rx
```

Sets the mask to allow read (r), write (w) and execute (x) access to the file owner, and read and execute access only to all other users.

## Notes

The "setmask" command does not affect the modes of an already existing file.

The "setmask" command is internal to the command interpreter, so that it affects the mask of the command interpreter and any commands invoked by it.

The default mask is: **rw**x **rw**x.

The actual modes of the created file are the logical AND of the mask and the modes requested by the process creating the file.

## **shutdown**

### Format

```
@>shutdown minutes {message} {-r} {-s} {-p}
```

### Description

The "shutdown" command shuts down the system from multi-user mode. The system will either enter single user mode or shut down completely, depending on the system configuration (see "vconf"). On the MAI 3000, the command can optionally power off the system.

The number following the command determines when the shutdown occurs. Shutdown occurs 15 seconds after the number of minutes specified. A warning is displayed on all terminals every minute and then 15 seconds before the system shuts down.

If a message is specified, it is displayed on all terminals with each warning message.

### Options

These options are mutually exclusive.

- r        Reboot. Shuts down the system and displays the reboot prompt.
- s        Shutdown. Shuts down the system to single user mode.
- p        Powerdown. On the MAI 3000 only, this option powers down the system following shutdown.

## Examples

@> shutdown 5 for the evening

This command shuts down the system with five minutes of warning, and displays the reason.

## Notes

Pressing <CTRL>+<C> up to the last possible moment at the terminal originating the shutdown aborts the shutdown.

## Files Used

/etc/ports

To find out what terminals are on the system.

/dev/tty\*

To write a message to each terminal.

## **sort**

### Format

```
@>sort {options} file1 ... filen
```

### Description

The "sort" command orders all the lines in the specified files according to the options specified. If no files are specified, input is taken from the keyboard, standard input. The result is then written to the standard output device or to a file specified with the "o=" option.

### Options

- b** Ignores leading blanks when comparing records.
- bs=** This parameter specifies the size of the internal buffer. The default size is 5000 bytes. A larger buffer makes for a faster sort. If the buffer size is too small for the sort, sort enlarges it.
- c** All character comparisons during the sort are case insensitive, treating upper and lower case characters as identical.
- f=** This parameter specifies the fields to sort. The field separator used is specified by the "d=" option. The modifier "a" specifies an ascending sort, the modifier "d" specifies a descending sort. The modifier "n" specifies a numeric sort. Several fields can be sorted individually by separating the field specifications with commas.

- d= This parameter specifies a single character as the field separator, which may be any single character. The default field separator is a TAB.
- ns Non-stable sort. The original order of identical items is not necessarily maintained.
- n Sorts based on the first numeric characters in the records. Decimal points are recognized as part of a numeric expression.
- o= This parameter specifies the output file.
- r Sorts the file in reverse ASCII order.

Example

```
@>sort d=: f=1a /etc/passwd
```

This command line will display the user information file in alphabetic order by user identification.

## **space**

### Format

```
@>space {-nh} dev1...devn
```

### Description

The "space" command reports the amount of space (in blocks) that is free out of the total amount on each device specified.

### Options

**-nh** This option suppresses the printing of headers in the display.

### Example

```
@> space /dev/boot
```

This command line displays the amount of space available on the boot partition.

### Notes

The program executes "sync" to update the filesystem before it reads the allocation bit maps.

Using the "raw device" form is slightly faster.

## **suspend**

### Format

```
@>suspend pid1 ... pidn
```

### Description

The "suspend" command stops the execution of the specified processes. A suspended process can then be continued, with the "resume" command, or terminated, with the "kill" command.

### Example

Start a background process:

```
@>/a/long/program &  
213  
.  
.
```

```
@>suspend 213  
213 suspended
```

This command line suspends execution of the program which is running in the background.

## **sync**

### Format

@>sync

### Description

The "sync" command causes all information in the system's disk buffers (cache) to be written out to the disk. You should use it before executing processes such as halting the system and copying or dumping the contents of the disks, which depend on all system buffers being flushed.

### Notes

The shutdown and umount commands automatically execute a "sync".

In multiuser mode, the system runs the program "/sys/update" to execute a sync every 30 seconds.

## **sysinfo**

### Format

```
@>sysinfo {options}
```

### Description

The "sysinfo" command prints to standard output (the terminal screen) the system serial number, the booted O.S. version level, and the initial system configuration. All or selected parts of the report may be printed. The default is that all of the information is displayed.

### Options

All options may be abbreviated by their first letter.

**-ssn**      Displays the system serial number.

**-version**      Displays the booted OS version level.

**-conf**      Displays the initial configuration record.

**-all**      Displays all information (-ssn, -version, -conf).

**-raw**      Displays the initial configuration record in its hexadecimal format.

## **ttymntbl**

### Format

```
@>ttymntbl {table} {-default}
```

### Description

The "ttymntbl" command loads or sets the specified terminal mnemonic table according to the specified options.

### Options

**-default** Sets the system default terminal mnemonic table. All further mnemonic processing uses this table.

### Example

```
@>ttymntbl edt
```

This command line loads the mnemonic file for edt terminals for local use.

## **ttymodes**

### **Format**

```
@>ttymodes {options} {modes}
```

### **Description**

The "ttymodes" command displays or sets the modes on the device associated with standard input, which in most cases is your terminal.

The modes deal with input, echo and output characteristics of the terminal. Characters are divided into two major classes: alphanumeric or printable ASCII characters, and control characters. The alpha characters are those in the range of "space" (hex 20) through "~" (hex 7E) inclusive, and hex A0 through hex EF, inclusive. The control characters are those in the ranges hex 00 through 1F and hex 80 through 9F. Special meanings are assigned to characters below 040 and DEL, 0177, when it is the erase character. The remaining control characters are defined as NORMAL and have no individual interpretation applied to them.

To display the modes currently set on the terminal, type "ttymodes" without any arguments and press <RETURN>. To alter the modes, type the mode either with or without the preceding caret ("^"). The caret preceding a mode means that the mode is turned off.

## Options

-default	This switch sets all modes to the standard initial state. This state is shown in the example below. Note that "erase" and "kill" keys are not affected.
-nowait	Normally, the system waits until all previous output has been sent to the terminal before changing modes. The "-nowait" mode forces the system to change the modes immediately.
erase=	The erase character is changed to the character specified. The character can be entered with three different notations to accommodate other meanings that some characters may have: as a number (decimal, unless it begins with 0x for hexadecimal or 0 for octal), as a caret followed by a letter (just like the character would be echoed e.g. ^H for backspace), or as the character itself.
linekill=	The line kill character is changed by this option. The three forms described above are allowed for specifying the new line kill character.

## Modes

The tty modes are described on the following pages.

### Input Modes

irawedit	When on, permits the input-editing characters to be read as normal characters; otherwise these characters have the effect either of deleting previously typed characters (the erase and line-kill functions) or of redisplaying a partial input line ( $\langle \text{CTRL} \rangle + \langle R \rangle$ ).
wakealpha	Characters are normally kept in an intermediate buffer, which is unavailable to the user's programs, until the new line character $\langle \text{NL} \rangle$ is entered. Thus a line can be altered with the erase character or linekill before the data is committed with a new line, which is the normal "wakeup" character.
	When on, wakealpha causes every printable ASCII character ('space' through "~", hex 20 through 7E, inclusive) to be a wakeup character. When it is typed, the line so far is made immediately available to the user. Currently, wakealpha mode implies irawedit mode.

wakectl	<p>This mode is similar to wakealpha but deals with all characters outside the printing ASCII set including ALL characters with the eighth bit on (see i8bit mode, below).</p> <p>When on, wakectl causes any normal control character to be a wakeup character. Alpha characters are saved in the buffer until any control character is typed. Thus printing characters can still be deleted before they have been committed.</p>
i8bit	<p>(Ignored on 7.2A). This mode causes all bits including the eighth bit (parity) to be accepted on input. Characters with the eighth bit on are input. Characters with the eighth bit on are treated as normal control characters for the purposes of echoing and wakeups.</p> <p>Although this mode does not disable the special meaning of any control characters, the system only recognizes those characters when the eighth bit is "off".</p>
prctl	<p>This mode enables recognition of process control characters (&lt;CTRL&gt;+&lt;C&gt; or &lt;CTRL&gt;+&lt;Y&gt;). When off, these characters are treated as normal control characters and thus have no system defined function.</p>

escape	When on, this mode enables escape processing as follows: after typing the ASCII <ESC> character, a dollar sign followed by a backspace is echoed placing the cursor on top of the dollar sign. The next character typed bypasses normal special character recognition and is echoed according to the echoing rules established with the ect1, erawctl and ealpha modes described below.
	This mode allows certain control characters to be entered literally. When off, <ESC> is treated as a normal control character.
eof	When on, <CTRL>+<D> is mapped to mean end-of-file. It acts like a wakeup character but does not appear in the data. If there are any preceding characters (not wakeup characters) they become immediately available to the user's program. If there are no preceding characters, the program will receive a zero-length input which is considered an end-of-file. When the mode is off, it is treated as a normal control character.
icrlf	When on, the ASCII carriage return character <CR> is mapped to the ASCII newline character (hex 0A) on input. When off <CR> is entered as <CR> (hex 0D).

## Echoing Modes

ealpha	Echoing of alpha characters is controlled. (See wakealpha mode.) When on, this mode enables the alpha characters to be echoed. When off, no echoing of the alpha characters takes place.
ectl	When on, this mode enables the control characters to be echoed as ^(LETTER)". When off, the erawctl mode is examined for echoing of NORMAL control characters. When on, this mode supersedes the erawctl mode, below.
erawctl	When on, this mode enables the control characters to be echoed literally as typed. If both ectl and erawctl are off, no echoing of control characters takes place except for those that have special meaning.
enobellerr	When off, this mode causes a <BEL> character (audible beep on most terminals) to be echoed in two cases: when the erase character is typed but there is no previous character in the buffer to erase (at the left margin) and when so many characters have been typed that the input buffer overflows.
etab	When on, this mode enables the echoing of the <TAB> character. How it is echoed is determined by the oxtab mode.
ecrlf	When on, the mode causes echoing of the carriage return character as a carriage return/line feed combination.
prctlecho	When on, echos ESCAPE and ^C to the terminal.

## Output Modes

o8bit	(Ignored on 7.2A.) When on, this mode prevents masking out the eighth bit (parity) on output.
octl	When on, this mode forces control characters to be printed as " <sup>^(LETTER)</sup> " on output, except for <TAB> and <NL>. When off, control characters are output as sent, except for <TAB> and <NL>.
ocrlf	When on, this mode maps <NL> to carriage return/line feed on output. When off, <NL> is output without a carriage return.
oxtab	When on, this mode causes <TAB>, on output, to be expanded to an appropriate number of spaces, depending on column position, assuming stops every 8 columns. When off, <TAB> is output literally.

## Additional Modes

mnemonics	When on, this mode enables mnemonic processing. For example, with mnemonics on, an <u>ESCAPE CS</u> will clear the screen. With mnemonics off, <u>ESCAPE CS</u> will not clear the screen, but will be printed on the terminal as an ESCAPE CS.
-----------	---

mapesc	When on, maps the <ESC> key to <CTRL>+<C>.
mapctlx	When on, after a ^X is read, the next read generates an error. Used for 13XX compatibility in BASIC.

### Examples

1.       @> ttymodes

This command line displays the current tty modes set for the user's terminal.

2.       @> ttymodes ^ealpha ^ectl ^erawctl ^etab  
  ^ecrlf -default

This command line sets the modes to the defaults and turns off echoing.

### Notes

The "ttymodes" command performs its changes on the device associated with standard input. It can, however, be redirected to affect other terminals (see examples).

Not all parameters recognized by ttymodes are necessarily supported by all serial devices on all machines. We have indicated which options are not yet in effect. If you attempt to use an unsupported option you will not be told of its failure.

## **ttyxlate**

### Format

```
@>ttyxlate {options} file
```

### Description

The "ttyxlate" command is used to load, set, or reset I/O translation tables. An I/O translation table is one-to-one mapping of characters from one value to another. For example, one could define a translation table which takes lower case letters and translates them to upper case. Translation can take place on input, output, or both.

### Options

- i        Causes input translation to be set or reset.
- o        Causes output translation to be set or reset.
- reset    Resets specified table or tables.

### Examples

1.        @> ttyxlate -o spanish

This command line uses the spanish translation table for output.

2.        @> ttyxlate -reset -i -o

This command line resets input and output translation.

## **umount**

### Format

```
@>umount dir1\dev1 ... dirn\devn {-oride}
```

### Description

The "umount" command prevents any further access to the filesystem which was mounted on the specified directories or devices.

Within the command, "sync" is executed to assure that the device is in a consistent state, after which no further access is made. Unmount fails if the device is in use (which usually means that some user has a current directory on the mounted device).

If a filesystem is located on a removable medium, such as a diskette, the medium should only be removed after an "umount" has been done.

### Options

The options may be abbreviated by their first letter.

**-oride** This option overrides mount errors, allowing for unconditional unmounting of a filesystem or directory.

### Examples

```
@>umount /mnt
```

This command line unmounts the filesystem mounted to directory "/mnt".

## Notes

Only mounted filesystems that are not busy can be unmounted. That is, if any user's current or alternate working directory is located in a mounted filesystem, or if any user or process is accessing a file in a mounted filesystem, that filesystem cannot be unmounted.

## Files Used

`/etc/mounttab`

To update list of currently mounted devices.

## **usb**

### Format

```
@>usb device {options}
```

### Description

The "usb" command updates or creates the superblock for the specified device. A superblock contains certain information about that device that is needed by the device driver. Only the system administrator can update a superblock.

### Options

<b>type=</b>	This option specifies the name of the device characteristics file in the directory "/etc/diskdesc" which contains device characteristics. This parameter is used when writing the information to the device or memory.
<b>desc=</b>	This option specifies the device characteristics file in the directory "/etc/diskdesc" created by reading the superblock from either the memory or the device.
<b>-set</b>	This option writes the superblock to system memory.
<b>-get</b>	This option reads the superblock from system memory.
<b>-save</b>	This option writes the superblock to the device.
<b>-strip</b>	This option disregards any existing partitions read from memory, device, or device characteristic file.

## Examples

1.      @>usb /dev/rwd0 type=wd0 -save -set

This command line reads device file "wd0" and writes the superblock to both system memory and rwd0.

2.      @>usb /dev/rwd0 -get -save

This command line reads the superblock from system memory and writes it to rwd0.

## **vconf**

### Format

```
@>vconf file {options}
```

### Description

The "vconf" command reads and writes configuration files. It allows changes to configuration files to be done quickly and easily. If a configuration file is saved, the old file name is moved to file name.bak.

### Options

<b>root=</b>	Root device; <b>MM</b> is device major number, <b>mm</b> is device minor number. These numbers should be in decimal.
<b>swap=</b>	Swap device name.
<b>size=</b>	Size of swapper in blocks.
<b>ram=</b>	RAM disk size in blocks.
<b>system=</b>	A string specifying the system name.
<b>-save</b>	Save configuration information to "file."
<b>-silent</b>	Display no messages on boot.
<b>-verbose</b>	Display normal messages on boot.
<b>-os</b>	Display the configuration file of the operating system that is currently being used.

-oride	Permits the filesystem to be mounted during load when left mounted.
-noride	Does not override "directory system left mounted" error.
buffers=	Number of system buffers.
linekill=	Set linekill character (key).
erase=	Set erase character (key).
locks=	Sets maximum number of file locks.
mfsys=	Sets the maximum number of mounted directory systems.
-single	After completing the boot procedure, the system will be in single-user mode.
-multi	After completing the boot procedure, the system will be in multi-user mode.
-secure	The system administrator password is asked for before the boot process can be completed.
-nonsecure	The system administrator password is not asked for in order to complete the boot process.
printers=	Sets the maximum number of printers allowed.
sockets=	Sets the number of "well known" LAN sockets.
dsockets=	Sets the number of "dynamic" LAN sockets.

lanbuffers=

Sets the number of LAN buffers.

lus= Sets the maximum number of logical units on the system.

opens= Sets the maximum number of open files allowed on the system.

eventcalls=

Sets the maximum number of eventcalls allowed on the system.

tsegs= Sets the maximum number of shared text segments.

procs= Sets the maximum number of processes that can simultaneously exist on the system.

-dump Enables the crash dump.

-nodump Disables the crash dump.

-debug System will go to the O.S. debugger on a system crash or when the reset button is pressed.

-nodebug System will not go to the O.S. debugger on a system crash.

ibsize= TTY input buffer size.

tbsize= TTY type ahead buffer size.

-inca Enable on-chip instruction cache (MAI 3000 only)

-noinca Disable on-chip instruction cache (MAI 3000 only)

-daca	Enable the 8K data cache (MAI 3000 only)
-nodaca	Disable the 8K data cache (MAI 3000 only)
dirc=	Maximum number of directory cache entries

### Examples

1. @>vconf /etc/conf root=1403  
swap=/dev/swap size=2000 -save

This command line alters the root, swap  
and swapsize in "/etc/conf.". Note here  
that "/etc/conf" must already exist.

2. @>vconf -os /etc/current -save

This command line reads the current oper-  
ating system's configuration file and  
saves it under the name "/etc/current."

### Notes

An existing configuration file must be specified if  
the "-os" option is not used.

If the "-save" option is not used, the changes in  
the configuration file will not be saved.

## **ved**

### Format

```
@>ved {-readonly} {-vedhelp} {buffers=} {file/name}
```

### Description

The "ved" command calls the BOSS/IX screen-oriented text editor. If you do not name a file, "ved" will assume you want to edit the last file you edited in your current directory. When you edit a file for the first time in an editing session, you must specify the file name.

### Options

The options may be abbreviated by their first letter.

**-readonly** During the editing session, you can only read the files. This is the "look-only" mode and since the file is not altered, the file's modification date remains unchanged.

**-vedhelp** The editor is called in read only mode with the on-line help file.

**buffers=** The number of buffers in memory.

## Editor Commands

All basic "ved" commands are control characters (the control key plus a character) or escape characters (the ESCAPE key and then another key). In the following description, <CTRL> represents the control key, and <ESC> represents the escape key. The combination <CTRL>+<X> is a single keystroke, while <ESC> <X> are two keystrokes. If the command takes an argument, the argument is terminated with a carriage return.

A command may be modified in three ways:

1. Pressing the <ESC> key before the command will change it to an alternate form of the command, usually a related function.

2. Some commands take a number argument. Numbers are entered by typing:

<ESC> ## <RETURN>

3. The ALT mode command,

<CTRL>+< ] >

inserts a mark in the text usually at the current cursor position. You can precede any command that moves the cursor with the ALT command; "ved" drops the mark before moving the cursor.

### Line One

When you execute "ved", the named file is displayed on the screen and the top line on the screen contains information about the editing session. This first line is divided into the following five fields:

- Field 1** Reserved for system messages, and questions. All confirmation is done here.
- Field 2** Contains the current argument to the command. It is reset to 1 after every command and can be set by the user.
- Field 3** The current take buffer. Default is "default", however, it too can be set by the user.
- Field 4** The name of the file being edited.
- Field 5** Error field. All warning messages appear here.

### Command Summary

Table 5-7 summarizes the "ved" commands. The character key in the left column must be pressed while holding down the key or keys shown in the top row. A dot indicates that the combination has no command value.

Table 5-7. "ved" Command Summary

	〈CTRL〉	〈ESC〉 〈CTRL〉	ALT	ALT 〈ESC〉
A	start line	startparagr	start line	start paragr
B	beginning	bottom	beginning	bottom
C	exit	backup file	.	.
D	delete chr	delete word	.	.
E	end line	end paragr	end line	end of paragr
F	forwrd chr	forwrd word	forwrd chr	forwrd word
G	get file	change file	edit macro file	.
H	back chr	back word	back chr	back word
I	tab	.	tab	.
J	save chr	save word	jump to mark	switch mark and cursor
K	kill line	kill paragr	kill sel	.
L	adjust	re-display	.	.
	window			
M	Return	change modes	set mark	.
N	down line	down paragr	down line	down paragr
O	open line	.	open line	.
P	up line	up paragr	up line	up paragr
Q	XON	.	.	.
R	forwrd srch	revrs srch	forwrd srch	revrs srch
S	XOFF	.	.	.
T	save line	save paragr	save sel	save sel
U	#=mult	mult=1	.	.
V	paste	paste and clear buffr	replic sel with buffr	replic sel clear buffr
W	write file	write take buffer	write sel	.
X	exec cmd	exec buffer	exec sel	.
Y	suspend	help	.	.
Z	research	revrs srch	research	revrs srch direction
¶	position from beg	direction position from end	position of mark from beginning	position of mark from end

Table 5-7. "ved" Command Summary (cont'd)

	⟨CTRL⟩	⟨ESC⟩ ⟨CTRL⟩	ALT	ALT ⟨ESC⟩
{	ESC	.	.	.
	ALT	ALT ESC	.	.
\	change buffer	clear pres buffer	.	.
^	quote chr undo delete	.	.	.
DEL	erase char	erase word	.	.

Files Used

/tmp/vedB.pid#	Backwards delete take buffer
/tmp/vedD.pid#	Delete buffer.
/tmp/vedF.pid#	Take buffer.
/tmp/vedX.pid#	Command buffer.
/tmp/ved.default.pid#	Default buffer.
/doc/comds/ved.help	On line help file.
.EDITMP.#	Stores the name of the last file edited in this directory by user number

## **wait**

### Format

```
@>wait {pid1 ... pidn}
```

### Description

The "wait" command causes the command interpreter to wait for child processes pid1 ... pidn to exit before accepting another command. If no pid is specified, the command interpreter waits for the first child that exits.

### Example

```
@>wait 115
```

This command causes the command interpreter to wait for process 115 to exit.

### Notes

An attempt to wait for a process that is not a descendant of the command interpreter executing the "wait" causes the command interpreter to wait for all child processes (if any) and to print the message "no children".

## **who**

### Format

```
@>who
```

### Description

The "who" command lists logged in users by terminal id and provides time of last login.

### Example

```
@> who
john      tty1 Jun 24 08:58:29
sue       tty8 Jun 23 08:43:17
```

### Files Used

/tmp/login

To find out who is on the system.

## **write**

### Format

```
@>write user(s) {options} {message}
```

### Description

The "write" command displays a message on the specified user's terminal. If you specify several users, separate their names with commas.

If you do not specify a message, write prompts you for lines of text. It sends each line until you enter <CTRL>+<D>.

Unless otherwise requested, write prefixes the sender's name and the time to all messages, and informs you if there are no recipients.

### Options

All options can be abbreviated to their first letter.

**-noheader** No header on the message. A message is sent without the sender and time header.

**-oneheader**

One header on the message. This option is used when engaged in a conversation. The first line of the message has a header, all succeeding lines do not.

**-silent**      Silent. Do not inform the sender if recipients were not logged on.

**Example**

dianne>write jeff,toni What about lunch?

On jeff's and toni's screens, the following is displayed:

From dianne @ 12:45:09 What about lunch?

## NOTES

## NOTES

## SECTION 6

### PROBLEM SOLVING

This section describes procedures that allow you to recover from many of the correctable error conditions you may encounter with your BOSS/IX system. There is also a brief description of how to run the diagnostics that were supplied with your system.

The presentations in this section are intended to be procedural only, and do not explain the error condition itself with any depth. The goal here is to give you procedures to aid you in recovering from the most common problems. More "in depth" descriptions for some problems may be found in the BOSS/IX Technical Reference Manual and the MAI 3000 Diagnostics and Error Log Reference Manual.

The error conditions and recovery procedures that are covered in the following chapters are:

- o Premature system shutdown, also known as a "system crash". The recovery procedure is automatically executed when the system is booted. Some of the other procedures described in this section may be necessary.
- o Filesystem damage, which may be a result of a premature system shutdown or removal of a floppy diskette without first unmounting it.

- o "Lost" files, another possible result of premature system shutdown. Suggestions for recovering such "lost" files are offered.
- o Keyed file damage, which is repaired using a command.
- o Auto-run diagnostics, which you may be asked to perform prior to a visit by your service representative.
- o **Printer Problems**

## **PREMATURE SYSTEM SHUTDOWN**

This section explains the procedures to follow if your BOSS/IX system suffers a "premature system shutdown," also known as a "system crash." The recovery procedure is semi-automatic, requiring very little user intervention.

The procedure presented in this chapter is in two steps. The first step saves the contents of system memory at the time of the "crash" for later analysis by an MAI Basic Four system analyst. The root filesystem is then automatically repaired. The second step is to reboot the system, at which time any additional filesystems on the fixed disk are also repaired. The system is then ready for use.

Additional repairs may be necessary, such as repairing a filesystem on floppy diskette, recovering lost files, and repairing keyed files. These procedures are covered in the following chapters.

The following paragraphs give a brief explanation of what a premature system shutdown is, and what is accomplished by the procedures of the remaining two chapters.

### **What is a Premature Shutdown?**

A premature shutdown typically results from a power loss, pressing the reset button, or a "fatal" error in the operating system software. The power loss may be due to someone turning off the power to the Central Processing Unit (CPU) without first going through the shutdown procedures.

In any case, the result is some degree of damage to the root filesystem, and any filesystems mounted to it at the time of the "crash." This damage must be corrected before the system can be restored to normal operation.

In the case of a "fatal" operating system error, knowing the conditions that led up to the failure is essential to preventing its recurrence. The error is usually a result of the interaction of the operating system and the jobs being run at the time of the crash. The hardware may be involved, as well as any applications programs being run at that moment. The procedures presented in this section make it possible for you to get this information from the system and send it to an analyst who will then analyze it in order to find the cause of the system crash.

When a fatal system error occurs, the system turns control over to a debugging program, recognizable by its prompt:

DEBUG>

When this occurs, the system usually automatically dumps the contents of its memory to the swap partition of the fixed disk or to tape. This is changeable as a configuration option (refer to the BOSS/IX Technical Reference Manual).

In many cases, you may not know that a premature shutdown has occurred until you attempt to reboot the system. The system then automatically executes the repair procedures.

## **What is a Memory Dump?**

When a fatal system error occurs, the system stops cold. The contents of system memory remain exactly as it was at the moment of the crash until the system is rebooted or power is turned off. System memory contains a lot of the information required to analyze the cause of a system crash.

A "memory dump" is a procedure for getting the contents of memory to the swap area of disk (on the MAI 2000 or an MAI 3000 without tape) or to tape (on an MAI 3000 with tape). Dumps written to the fixed disk are written to either floppy diskettes or tape, and the floppy or tape is then sent to an analyst.

When you reboot the system, you are given the option of saving this memory dump information before continuing with the repair procedure. As remarked above, the contents of the memory dump are useful in analyzing the cause of the crash.

### **System Dump Configuration**

Before you need any of the procedures in this section, you should make sure you know how your system is presently configured. As mentioned above, your system comes from the factory configured for automatic memory dumps, but you should still check to make sure.

This procedure assumes your system is running under normal conditions, in particular that the system is "rooted" on the root partition, "/dev/root". This is where the system roots during a normal system load. This check cannot be done while the system is in a crashed condition.

The main configuration file for your system is located on the boot partition of the system fixed disk, in the directory "/etc", and is simply called "conf". If you need to change the configuration record, this is the file you would change. Information on changing your system configuration files is contained in the BOSS/IX Technical Reference Manual.

At your command interpreter prompt, type:

```
vconf -os
```

and then press <RETURN>.

Most of the terminal screen is filled with the display of configuration parameters, but at the bottom of the list will be a line such as:

Startup in secure, single-user mode with  
system dump enabled.

or

Startup in secure, single-user mode with  
system dump disabled.

The line may differ, for instance your system may startup in non-secure mode or multi-user mode. The part that you are interested in is whether system dump is "enabled" or "disabled." If system dump is disabled, then your system is configured for manual memory dumping. If system dump is enabled, then your system is configured for automatic memory dumping.

If system dump is disabled, it should be enabled by using the "vconf" command with the following command (the boot filesystem is mounted to "/mnt"):

```
@>vconf /mnt/etc/conf -save -dump -nodebug
```

Your service representative will give you special instructions if there is a reason for automatic dumping to be disabled.

### **Recovery Procedure**

The remainder of this procedure assumes the system is configured for automatic dumping. The procedure varies depending on whether your system is an MAI 2000 or MAI 3000, and the availability of tape drives on the MAI 3000.

If your system crashes due to a fatal system error, it immediately executes the memory dump routine. The routines for the various systems and configurations are described below.

#### **MAI 2000**

The MAI 2000 automatically dumps memory to the swap partition of the fixed disk, and then displays the following:

**System Dump Completed**

Press 'RETURN' key to reboot ('^C'=altload, '^S'=self-test):

Press <RETURN> to reboot.

When you boot the system following a premature shutdown, the boot procedure proceeds as normal except that the root partition cannot be mounted. The boot partition is used as the root filesystem. Messages are displayed indicating this and you are prompted:

Press <RETURN> key to begin automatic system check/repair.

Press <RETURN> to begin the repair procedure.

Since your system has a memory dump on the swap partition of the fixed disk, you are now told that it exists and that it must be handled before any repairs are made. Your choices are to:

- (1) save crash dump to streamer tape.
- (2) save crash dump to floppy disks.

<RETURN> Continue with system check.

Your selection will depend on the removable storage media you have available. Make your selection and follow the prompts. <RETURN> clears the swap partition without saving the dump. Save the dump unless advised otherwise by your service representative.

You will be prompted to insert the floppy or tape and press <RETURN> when ready. Follow the prompts until the dump has been completely written.

#### MAI 3000

If you have an MAI 3000 that does not have an MCS or MTS tape drive, the procedure continues as for the MAI 2000. Refer to the description immediately above. When the save crash dump menu is displayed, you probably have no choice but to press <RETURN> and clear the dump.

If you have one tape drive, either an MCS or MTS, the system attempts to write the memory dump directly to the tape, without writing it to disk first. You are prompted:

Insert an MCS cartridge

or

Mount and make ready an MTS reel

Make the tape drive ready and press <RETURN>. Memory is then dumped to the tape.

If you have two tape drives, a 1/4 inch MCS and a 1/2 inch MTS drive, a menu of tape devices is displayed. Select the drive to use. and proceed as above.

After the dump has been written, the reboot menu is displayed. Press <RETURN> (or selection option 1, "Boot") to reboot the system and begin the automatic filesystem repair procedure.

#### Tape Error Recovery

On the MAI 3000, if tape errors occur while attempting to write the memory dump to tape, the error is noted on the screen, and you are prompted to press <RETURN> to retry.

If after several retries, the errors persist, press <ESCAPE>. On systems with a single tape drive, the memory dump is then written to disk (on the swap partition).

On systems with two tape drives, the system attempts to write the memory dump to the alternate drive. If it fails too, press <ESCAPE> again to write the dump to disk.

Before the filesystem repair procedure begins, you are prompted to write the dump to tape, as described above for the MAI 2000. Since memory is usually larger than the swap area, dump data will be lost, so this is not the preferred procedure.

## **Repairing The Filesystems**

After the memory dump has been handled, or immediately, if there was no memory dump, the process begins repairing damaged filesystems on the fixed disk. The filesystem on the boot partition, if it was damaged, is not repaired at this time, but will be at next boot. A damaged filesystem on a floppy is not repaired automatically, but must be repaired using a procedure given in Chapter 3.

While each filesystem is being repaired, the following message is displayed:

Repairing filesystem for device (14,X) ...

where "X" is the partition number containing the filesystem. The 14 signifies the fixed disk controller.

The length of time required for repair of a filesystem depends on its size. The root filesystem, usually 14,3, is very large, and takes a few minutes to check and repair. When the repairs are completed, the number of errors detected and the number repaired are displayed, and repair of the next filesystem is begun.

When all filesystems have been repaired, the system is shutdown and the reboot menu/prompt is displayed.

Press <RETURN> to perform a normal boot.

When the system is booted normally, the system repairs the filesystem on the boot partition, if necessary, and then completes the normal boot process.

At this point, the system is back to normal functioning. However, other damage may still need repair.

If a floppy was mounted at the time of the premature shutdown, it needs to be repaired also. The procedure to follow is given in Chapter 3, "Filesystem Repair."

Some directories or files may be renamed "/Lost\_#", where "#" is a number (the inode number). Some suggestions for recovering these are given in Chapter 4, "File Repair."

Some keyed files may have become damaged. A procedure for repairing these is also given in Chapter 4, "File Repair."

Lost or damaged files that cannot be recovered by these procedures are either lost forever, or can be restored from backup copies. These procedures are covered in Section 5 of the User Guide, "Backup and Restore."

## **FILESYSTEM REPAIR**

The BOSS/IX file system is organized at its highest level into individual parts called "filesystems." These filesystems, and how they are used, is described in Section 2 of this manual, "The BOSS/IX File System."

In the course of events, filesystems can become corrupted. Probably the most common cause is premature system shutdown. In most cases, corruption to a filesystem caused by premature shutdown is automatically repaired the next time the system is booted. This is true of any filesystem existing on a partition of a fixed disk. It is not, however, true of a filesystem contained on a floppy diskette. These filesystems, if corrupted, must be repaired by the procedure described in this chapter.

Filesystems, even those on fixed disk, may suffer corruption by causes other than premature shutdown. In these cases, the filesystem may not have been left mounted, and so will not be repaired at the next boot. This kind of filesystem damage must also be repaired by these procedures.

There are two facilities available for repairing filesystems: a utility and a command. Both procedures are covered here. The utility program allows you a little more control over the repair process.

#### NOTE

1. Only the system administrator is allowed to repair a filesystem.
2. A filesystem may not be mounted while it is repaired. It may, however, be mounted if it is only being checked for damage.
3. If possible, a backup copy of the filesystem should be made before an attempt is made to repair it. Data may be lost as a result of the repair procedure.

It is assumed in the remainder of this chapter that you are repairing a filesystem. The variations of these procedures for checking a filesystem are described in the Utility and Command Language sections of the BOSS/IX User Reference Manual.

## **Filesystem Repair by Utility**

The Filesystem error analysis utility is a fully interactive program designed to make repairing a filesystem simple. The utility itself has both a semi-automatic repair procedure and a "manual" repair procedure. Only the automatic repair is explained here. The manual procedure is discussed in the BOSS/IX Technical Reference Manual and is intended only for technically knowledgeable users.

To access the Filesystem error analysis utility from the Main Utility Menu, select option 3, "Filesystem & disk," and then selection option 5, "Filesystem error analysis." The Filesystem error analysis utility header and the first prompt are now displayed on the terminal screen.

### Specifying A Filesystem

The first prompt displays a list of the filesystem devices available on the system. The list varies depending on the configuration of your system. A single filesystem device may be represented more than once on the list. Type the number of the filesystem device you want to repair and press <RETURN>.

If the filesystem you want to repair is on diskette, insert the diskette into the diskette drive and select the number of that drive.

When you have selected a filesystem device, the program displays the name of that device and asks if you want to perform the automatic analysis:

This program can analyze the filesystem automatically or with user direction.

Perform automatic filesystem analysis? (yes/no):

You want to proceed with the automatic analysis and repair, so type "yes" or "y" and then press RETURN. Answering "no" leads to the manual procedure.

The program then asks if you want to repair the filesystem:

Repair the filesystem? (yes/no):

If you want the program to repair the filesystem, as we are assuming here, type "yes" or "y" and then press RETURN. Answering "no" will perform an analysis only.

Before repairing the filesystem, the program displays this warning and asks for verification before it proceeds:

Warning: Files or data may be lost. Proceed?  
(yes/no)

If possible, a backup copy of the filesystem should be made before a repair is attempted. If you want to proceed with the repair, type "yes" or "y" and then press RETURN. The program will then proceed with the analysis and repair.

### Analyzing and Repairing the Filesystem

Once you have responded to the above prompts, the program begins with the analysis and repair of the filesystem. As it proceeds, progress report messages are displayed, along with messages indicating any filesystem errors the program finds.

If the program is attempting a repair of the filesystem, and it discovers a reference in an inode to a duplicate or illegal block number, the program asks for the corrective action to take, for example:

Bad block 1234551 in inode 235.  
Delete block from inode? (yes/no)

Type "yes" or "y" to delete the block, "no" or "n" not to delete the block, and then press RETURN. It is generally recommended that you delete the block. If you answer "no", the program asks:

Delete the inode? (yes/no)

Type "yes" or "y" to delete the inode, "no" or "n" not to, and then press RETURN.

If you answer "no," the program displays the message, "you must do one or the other!", and repeats the "delete block?" prompt.

### Exiting the Utility

When the program has finished analyzing and repairing the filesystem, it temporarily displays the number of errors found. It then repeats the list of available filesystems.

When you are finished repairing filesystems, press RETURN or CTL-I alone. The program then ends and exits to the point from which it was accessed.

### **Filesystem Repair by Command**

The filesystem repair command, "fschk," is a fully automatic procedure. Unlike the utility, however, it is not interactive. You must know the name of the filesystem you need to repair in order to repair it.

Devices, including filesystem devices are files in the directory "/dev". List the contents of this directory to see the devices defined on the system.

Only a few of them are filesystem devices. Use the "ls" command with the "-l" option. Devices with a major device number 14 are either a fixed disk or a partition on a disk. Devices with a major device number 7 are floppy diskette drives, and are accepted as filesystem devices.

When you know the name of the filesystem device you need to repair, type at the system administrator prompt:

```
ADMIN>fschk -fix /dev/filesystemname
```

If you are repairing a floppy diskette, the diskette must be in the drive. Press RETURN to begin repairs.

The program first displays:

(Check and fix mode)

A series of messages are then displayed indicating the progress of the program. If the filesystem had been left mounted, a message indicates that. When the program is complete, a message indicating the number of errors found and repaired is displayed. The program then ends.

The length of time it takes for a filesystem to be repaired depends on the size of the filesystem. A large filesystem, such as the root partition, "/dev/root," takes several minutes.

If more errors were found than corrected, run the program again on that filesystem, repeating until all errors are corrected.

## Filesystem Analysis Messages

While the Filesystem Error Analysis utility or "fschk" is running, several messages are displayed. As a minimum, messages indicating the three stages of the checking procedure are displayed. These three messages are:

- o SEARCHING INODES
- o CHECKING BLOCK MAP
- o CHECKING REFERENCE COUNT

The work performed during each of these stages is explained in the Technical Reference Guide. Errors may be detected during any of these stages.

## Error Messages

The following are the error messages that are displayed most often. In these messages, "#" indicates a number that is included in the display. The corrective actions described only apply if the filesystem is being repaired.

### File number does not match on inode #

This message is displayed while searching inodes if the file number in an inode is wrong. The number is changed to the correct value without user intervention.

### Illegal inode type #

This message is displayed while searching inodes if the file type is illegal. The inode is deleted without user intervention.

**Block # in more than 1 inode**

This message is displayed while searching inodes if an inode references a block that has been referenced by another inode already. In the utility procedure, the user is asked to delete either the inode or the block. In the command version, the block is deleted without user intervention.

**Inode # is in the bitmap but not used**

This message is displayed while searching the inodes if an inode is not in use but the filesystem bitmap indicates that it is. The bitmap is changed without user intervention.

**Inode # is not in the bitmap but used**

This message is displayed while searching the inodes if an inode is in use but the filesystem bitmap indicates that it is not. The bitmap is changed without user intervention.

**Inode # is an illegal directory type file-type**

This message is displayed if the inode is for a directory but is not type "small" or "large." In the message, "file-type" is the type. The directory flag in the inode is turned off without user intervention.

**Block # is not in the bitmap but used**

This message is displayed while checking the block map if a block is not in the filesystem bitmap but found to be in use. The filesystem bitmap is changed without user intervention.

**Block # is in the bitmap but not used**

This message is displayed while checking the block map if a block is in the filesystem bitmap but found not to be in use. The filesystem bitmap is changed without user intervention.

### **Remaining Errors**

If the above procedures leave filesystem errors unrepaired, you have suffered serious corruption of the filesystem. If you can mount the filesystem, copy as many files as you can to a removable medium. Other files will need to be restored from earlier backup copies. The filesystem will then need to be remade, and the files restored.

On floppy diskettes, the above procedure is simple. The Format diskette utility makes a new filesystem. Remaking a filesystem on a fixed disk partition requires use of the command "makefs". This command is described in Section 5 of this manual, "Command Language."

## FILE REPAIR PROCEDURES

Occasionally, files become damaged and need to be repaired. In this chapter, two kinds of file damage, and procedures for recovering damaged files, are described. These procedures are:

- o Keyed File Repair
- o Lost File Recovery

Both of these procedures may be necessary after a filesystem repair has been performed. The keyed file repair may be necessary at other times as well.

### **Keyed File Repair**

Keyed files are files of type "direct" and "sort," as well as other files used by certain applications, such as the Informix Data Base Management System. BOSS/IX keyed files are managed by the C-ISAM file organization, which is based on a b-tree structure.

Keyed files occasionally suffer damage to the keys and/or data, and so need to be repaired. The damage may be a result of premature shutdown, "noise," or fluctuations, in the system power, or any number of other conditions.

A utility and a command have been supplied to repair keyed files. The utility is described in Section 4 of this manual. For additional information on the command, refer to Section 5 of this manual.

You must be in the command interpreter to execute the command. You must also have read and write access to the file you want to check and repair.

To execute the keyed file repair command, type at the command interpreter prompt:

```
@>kychk filename -fix
```

and then press <RETURN>.

The "-fix" is an option, specifying that the program is to attempt to fix any errors it finds. If this option is not specified, the file is checked for errors only, and no repair is attempted.

The following additional options are available:

- data      This option specifies that each data record is to be examined.
- listkeys    This option displays information about each b-tree node along with detailed information about each entry in the node.

The program displays messages indicating its current activity, and then summarizes the results of the check and, if any errors are found, the number repaired.

### **Lost File Recovery**

When a filesystem is repaired, files may be renamed as a result of the filesystem corruption. During the repair process, some files may be found, but their names lost. These files are placed, by the repair process, into the root directory of the repaired filesystem. The name of such a file is "/lost #", where "#" is the inode number of the file.

The procedures described here are intended to help you to identify the "lost" files so they can be returned to their proper place in the filesystem. Once a file is identified, it can be moved to the proper directory.

The first step is to locate the lost files. To do this, list the contents of the root directory of the repaired filesystem. This can be done with the command "ls -l" or with the Display Directory utility with the detail list set to "file attributes". Either of these procedures includes the file type, which you need.

The method for identifying the contents of files differs for each file type.

Event count file names cannot be identified, so delete any lost files identified as event counts.

Directories can usually be identified by their contents. List the contents of lost files that are directories to examine the files they contain. A listing of the filesystem prior to the damage is helpful here.

For BASIC program files, start BASIC, load the file, and then list the first few lines. Line 10 often is a REM statement containing the program name. Otherwise, list the program to figure out what it does. From its function try to find its name from previous listings.

The contents of string files may be text or an executable program. Display the contents using the command "p":

```
@>p /lost_#
```

If the file contains an executable program, the display will be meaningless. If it contains text, try to determine the file name from its contents.

The contents of other files can be displayed using the "dump" command:

```
@> dump /lost #
```

Executable files are often identifiable by text in them.

Identify as many files as you can. Use a listing of the filesystem to help determine which files are missing from the repaired filesystem, and match those to lost files.

When you have identified one of the lost files, rename it using the "move" command or the Move File utility. Make sure the destination directory exists. To use the Move File utility, set the "rename files" list to "yes". The "move" command can be used like this:

```
@>move /lost_# /file/name
```

where "/file/name" is the full path name of the file being recovered.

When you have identified all of the lost files you can, any files still missing need to be restored from backup copies. Comparing the contents of these restored files to the remaining lost files may help in identification.

## PRINTER PROBLEMS

There are many printer difficulties that are easily traced and corrected. This chapter covers a few of the most often encountered printer problems.

### Symptoms and Fixes

Symptom	"lpq" indicates that the current print job is "working", but nothing is printing on the printer.
Fix	Type "lpstat". If the indication is "offline", press the "online" button. If it is online, check the ribbon, paper, and cable connections. If it indicates that the paper or ribbon is out, replace what is needed.
	Some printers will finish the lines already received. Nevertheless, the despooler reprints the page that the error occurred on when the error is corrected.
Symptom	lpq shows an "error" for a print job in the print queue.
Fix	There is an error log file used to hold messages when an error occurs. The file name is "printername.err". Substitute a printer name before the ".err". These files are in the "/etc" directory. So, for example, if you have a printer called "lp", type:

`p /etc/lp.err`

and press RETURN.

If the error is a data transmission error, use "lpaint" to resume the job. If it recurs, call an equipment maintenance representative.

**Symptom** If you use lpr to print a job, with or without spooling, and nothing is being printed. Do an lpstat. The status report says working or idle and nothing is printing.

**Fix** Check to see if both ends of the cable are plugged in and check to see if the printer is turned on. This problem only occurs when using MAI Basic Four printers with the "Industry Standard Interface" which does not provide data transmission verification.

**Symptom** The status report shows that the printer is idle, yet the print queue report shows a print job waiting.

**Fix** This is a fairly technical problem requiring the system administrator. For more information on the tasks involved, refer to the Technical Reference Guide. Here are the basic steps:

1. log on to the system console as the system administrator
2. execute the command "ps -a" to report the processes
3. find the process called "/sys/lpd"
4. execute the "kill" command for the number of this process:

ADMIN>kill ###

For printers with spooling configured "ON", since the despooler is automatically restarted. If spooling is configured "OFF", the spooler will need to be started, as described in Section 3 of this manual.

**Symptom**      The print queue still reports a waiting print job after you have restarted the despooler, but nothing is printing.

**Fix**      Try recreating the print job.

1. Shut down to single-user mode

2. Execute this command line:

**@>delete /etc/\_queues/lpq.que**

3. Re-enter multi-user mode

Note that this procedure deletes all print jobs from the queue. All jobs will need to be re-submitted.

## **Problem Areas to Check**

Below is a list of the general checks to make in order to resolve most printer problems:

1. Use "lpstat" to check the printer status
2. If spooling is on, check "lpq" for the status of the print jobs.
3. Check the printer error log. Type

**p /etc/printernname.err**

and press <RETURN>.

4. Check the cables at both ends for good connections.
5. Make sure there is a defaults file. Type

**p /etc/defaults**

and press <RETURN>. There must be at least one entry called "system". Use the text editor to add entries.

6. Make sure there is a class file. Type

**p /etc/class**

and press <RETURN>.

7. Make sure there is a forms file. Type

**p /etc/forms**

and press <RETURN>. There must be at least an entry called "stanrd".

8. Check the printers file. Type

```
p /etc/printers
```

and press <RETURN>. There must be an entry for the printer you are trying to use, and there must be an "alias" entry called "default".

9. Check the ports file. Type

```
p /etc/ports
```

and press <RETURN>. There must be an entry for the printer you are having trouble with.

10. Make sure there is a printer device in the device file for the troublesome printer. Type

```
ls -l /dev/printer name
```

and press <RETURN>.

11. Check the process status to make sure spooling is active. Type

```
ps -a
```

and press <RETURN>. Look for "/sys/lpd" to be running for the correct printer.

12. Check for the existence of the print queue file. Type

```
ls /etc/_queues/lpq.que
```

and press <RETURN>.

13. Use the Port Configuration utility to see if the port characteristics for the port are correct.
14. Delete all spooled print jobs as explained earlier and then shutdown and restart the system.
15. Call your service representative.

## STARTUP ERRORS

The following paragraphs describe a number of errors that might occur during system startup. The probable cause of the problem is then given together with corrective actions to take to eliminate the problem.

The severity of the problems listed here varies. Some of the problems only cause an entry in the system startup log, "/etc/sys.log", others cause the system to shutdown, and still others cause a system crash. The errors are grouped in this order to aid in isolating the cause of the startup error.

The error that occurs also varies depending on whether you are entering single user mode or multi-user mode. These differences are included in the descriptions.

All of the files mentioned in this chapter are on the install tape or floppies marked "EOS". If you need to restore a file from the install copy, use the "mcsrestore" or "mtsrestore" command, and specify "name=EOS" for the saveset.

### Errors Causing Error Log Entries

Most system startup errors result only in the display of a message indicating that an error occurred. The message says that a description of the error is contained in the file "/etc/sys.log". This file can be displayed using the "p" command after the system boot has been completed:

```
ADMIN>p /etc/sys.log
```

Occasionally, some part of the system, a terminal or printer for instance, will not work until the problem is corrected and the system is rebooted.

There are problems which prevent the system from being started into multi-user mode. In this case, the attempt to start multi-user mode fails, a message is displayed, and operation is returned to single-user mode.

These messages and suggested corrections are given below. The messages are described, and are not necessarily as they are shown in "/etc/sys.log".

#### Messages and Fixes

Message exec could not open "/dev/kmem".

Fix Remake the file using this command:

@> makedev u 0 1 /dev/kmem

Then shut down and reboot the system. If the file exists, delete it before executing the command.

Notes This problem causes the system to drop into single user mode.

Message "/bin/lpmaint" is missing, access permission denied.

Fix Verify that the file exists and has execute permission set. Add execute permission if necessary.

If the file is missing, restore it from a backup copy or from the installation copy.

Message	"/etc/lpstart" is missing, access permission denied, or that exec failed.
Fix	Verify that the file exists and has execute permission set. Add execute permission if necessary.
	If the file is missing, restore it from a backup copy or from the Installation media.
Message	There was a problem setting transmission characteristics, opening a device, or that "/etc/ports" is missing.
Fix	Use the Port Configuration Utility to create or correct the "/etc/ports" file. Terminals may need to be redefined.
Notes	This problem returns the system to single user mode.
Message	"/etc/printers" is missing or has a bad entry.
Fix	Use the Port Configuration Utility to create or correct the file. Printers may need to be redefined.
Message	There was a problem reading a mnemonics table, loading a translation table, or file is missing or has a bad format
Fix	Use the Port Configuration Utility to create or correct the file. Terminals may need to be redefined.

Notes	This problem also causes one or more terminals to fail to come ready.
Message	"/etc/ttymntbl/XX" could not be read, where XX is the name of the mnemonics table file.
Fix	Replace the mnemonics table file from a backup copy, or recreate the file using the "makettymntbl" command. Standard mnemonics tables may also be restored from the installation copy.
Message	"/etc/ttyxlt/XX" could not be loaded, where XX is the name of the translation table file.
Fix	Replace the translation table file from a backup copy, or recreate using "makettyxlate" command. Standard translation tables may be restored from installation copies.
Message	"/sys/lpd XX" was not started, where XX is a printer name.
Fix	Verify that the file exists and has execute permission set. Correct the access permission if necessary. If the file is missing, restore it from the backup or from the installation copy.

## **Errors Causing Shutdown**

The following few errors result in a message being displayed indicating the problem, and the system will shut down. No additional damage is done to the system when this occurs. You are, however, prevented from returning to normal operation until the problem is fixed.

The errors that cause a system shutdown, together with their probable causes and solutions, are listed below. The corrective actions require performing an alternate load to the boot partition, "/dev/boot", and mounting the root partition, "/dev/root".

### **Messages and Fixes**

<b>Message</b>	Cannot execute "/bin/admin"
<b>Fix</b>	Verify that the file exists and has execute permission. Correct the filenodes if necessary. If it is missing, copy the file from the boot partition to the root partition.
<b>Message</b>	Execution of "/bin/command" failed. The shutdown prompt is then displayed.
<b>Fix</b>	Verify that the file exists and has execute permission. Correct the filenodes if necessary. If it is missing, copy the file from the boot partition to the root partition.

Message      "/etc/conf" is missing. The system then roots to the boot partition.

Fix           Either:

1. Use the "vconf" command to build a new "/etc/conf" file on the boot partition, or
2. re-install the operating system and run "/sys/install/etc" to restore the configuration file.

Message      "/bin/admin" can't find admin entry.

Fix           Restore the "/etc/passwd" file from a backup copy, or copy it from the boot partition to the root partition and then use the Operator Information Utility to recreate user entries.

## **Errors Causing a Crash**

A few possible problems cause the system to crash or hang either after or during booting. A probable cause and corrective action for each of these problems is given below.

### **Symptoms and Fixes**

**Symptom**      System hangs after booting. No message is displayed.

**Fix**            1. Perform an alternate load to boot partition, mount "/dev/root", and check that "/etc/start" exists and has execute permission. Correct the filemodes if necessary. If it is missing, copy the file from the boot partition to the root partition.

2. A terminal configured with DTR flow control may be powered off or disconnected. Fix the problem, and restart the system.

**Symptom**      System does not boot. An error trap message may be displayed. A system crash message is displayed.

**Fix**            Reinstall the operating system. Refer to the BOSS/IX Technical Reference Manual for instructions.

## **If the Problem Isn't Fixed**

If the problem you are experiencing is still not fixed by the procedures given here, your operating system may need to be reinstalled. Procedures for installing the operating system are given in the BOSS/IX Technical Reference Manual. If the system still doesn't work, you may have hardware problems. Consult your service representative.

## DIAGNOSTIC PROGRAMS

This subsection describes the procedures for running the Logic Tests and System Interaction Test in their automatic modes. These instructions are provided to assist you in running these diagnostics prior to a visit by your service representative. You should run these diagnostics only if you have been instructed to by your service representative.

The full set of diagnostics is contained on a separate MCS cartridge or floppy diskette. In addition, floppy and MCS diagnostics are contained on the fixed disk.

The procedures covered here are

- o Booting the diagnostic executive
  - from floppy diskette
  - from MCS
  - from the system disk
- o Running the Logic Tests in auto-run mode
- o Loading and running the System Interaction Test

The reports produced by the tests are briefly explained, to help you in reporting the results to your service representative.

## **Fixed Disk Resident Diagnostics**

Both the MAI 2000 and MAI 3000 have disk resident diagnostics installed in the factory. Although for both systems you must perform an alternate load to access the diagnostics, the specific load procedure is different.

The MAI 2000 (under BOSS/IX 7.2) supports disk resident diagnostics for checking the floppy and MCS subsystems only. These diagnostics are used to check the floppy and MCS drives in case of boot failures from these devices. Instructions for these diagnostics are given at the end of this subsection.

The MAI 3000 allows all of the diagnostics programs to be resident on the fixed disk. For the MAI 3000, but not the MAI 2000, you can perform the diagnostic procedures described in this subsection by booting the fixed disk resident diagnostics.

## Alternate Load

To run the Logic Tests and System Interaction Test, first you need to perform an alternate system load. On the MAI 2000 you must boot to the diagnostic floppy diskette or MCS cartridge. On the MAI 3000 you may boot to the diagnostic MCS cartridge, MTS reel, or, if the full diagnostics are installed on your fixed disk, from the fixed disk.

1. Shutdown the system to the reboot prompt (MAI 2000) or reboot menu (MAI 3000).
2. For the MAI 2000, press <CTRL>+<C> at the reboot prompt.

For the MAI 3000, select item 2, "Alt-load", at the reboot menu.

### 3. Boot Device:

Specify the device containing the diagnostics program and press <RETURN>. The device designations are:

fd0	- floppy drive (MAI 2000 only)
cs	- MCS drive (MAI 2000 or MAI 3000)
ts	- MTS drive (MAI 3000 only)
wd0	- fixed disk (MAI 3000 only)

#### 4. System file:

For diagnostics on floppy, MCS or MTS, press <RETURN> alone.

For diagnostics on the fixed disk (MAI 3000) type:

diag

and press <RETURN>.

The diagnostic executive is loaded and its prompt is then displayed:

<exec>

You are now ready to run the Logic Tests or to load and run the System Interaction Test. You may do either or both, as instructed by your service representative.

#### Start Printer Output

If you have a parallel printer or a serial printer with an industry standard interface configured as the system printer, you can have the diagnostics print their messages on the printer as well as the terminal screen.

To print messages to the serial printer, type at the <exec> prompt:

option sp

and press <RETURN>. For a parallel printer, type:

option lp

and press <RETURN>.

Make sure the printer is on-line. Now proceed with the tests.

## **Running the Logic Tests**

Once you have performed the alternate load to load the diagnostic executive you are ready to run the Logic Tests. Only the automatic test mode is described here.

At the diagnostics executive prompt, type:

**auto**

and then press <RETURN>. The program then displays this warning and asks you to verify that you want to continue:

**WARNING: ALL NON-WRITE-PROTECTED MEDIA IS  
SUBJECT TO CORRUPTION (WINCHESTER DISK  
EXEMPT)... Continue (y/n)?**

The tests perform read/write testing on the MCS and floppy diskette drives. Make sure your diagnostics cartridge or floppy is write protected. In addition, make sure that any other cartridge or floppy is either write protected or contains only data that can be destroyed. The write tests do destroy data.

This begins running the Logic Tests. No further intervention is required until the tests are complete. Note that the memory test lasts a long time (several minutes) with no apparent activity.

At each step of its execution, the program displays a message indicating what it is doing. First it loads a particular logic test, then runs it. Each logic test is made up of several sub-tests. As each sub-test is run, its number and a brief description is displayed. When a logic test has been completed, a message indicating the number of errors that occurred is displayed. The next test is then loaded and run, and so on until all tests are finished.

The last test to run is the fixed disk logic test. When the logic tests are finished, the prompt for this test is displayed:

<disk>

Unless you are printing the test results, the tests do need to be monitored, so if any errors occur the logic test and sub-test that gave the error can be noted. The total errors for each logic test should also be noted. These are the results you should report to your service representative.

Some apparent errors only indicate that the MCS cartridge or floppy diskette is write protected, and these errors can be ignored. Similarly for the fixed disk test, some sub-tests are not optioned to run. If in doubt of the relevance of an error, write it down.

#### Shutdown

If you are only running the logic tests, you may shut down the system at this time, and reboot. At the disk test prompt, type "shutdown" and then press <RETURN>. The usual reboot prompt or menu is then displayed.

#### **Load and Run the System Interaction Test**

When you have loaded the diagnostic executive you may load and run the System Interaction Test. You may also run these tests after you have run the logic tests.

At the diagnostic executive prompt (or at the last logic test prompt), type:

LOAD SIT

in either upper or lower case, and then press <RETURN>.

This loads the System Interaction Test into memory. While the program is loading, several rows of dots are displayed on the screen. When it is finished loading, it displays:

SYSTEM INTERACTION TEST (SIT) Rev. XX, wait <= 30 seconds for auto sizing

The SIT program is now operating. It begins by searching for the devices that are ready to be used during the test. It then builds a configuration file for those devices for use during the test. This is what is meant by "auto sizing"

Note that SIT will configure a floppy drive for testing only if there is a diskette in the drive and the lock button is pressed in. Similarly, it will configure the MCS drive for testing only if there is a cartridge in the drive and the lock lever in the up position.

When SIT is finished auto sizing, it displays a report of the controllers and devices present and the status of each.

The devices configured list will be needed by your service representative for evaluating the test results. Copy the list, including at least the contents of the device and status columns.

Below the configuration table display, the memory addresses where the SIT program is contained and the memory addresses that are free are displayed. Below that is the System Interaction Test prompt:

<sit>

To run the test, type:

run

in upper or lower case letters, and then press <RETURN>. This starts the test running.

The System Interaction Test will run for as long as you leave it. Your service representative may give you a minimum duration, for example, over night, during which it should not be interrupted.

While it is running, the program will periodically display a "News Update," summarizing the results of the test to that point. At any time you can write down the results, or read them off to your service representative.

#### Shutdown

To stop the SIT test, press ESCAPE. The sit prompt is then displayed:

<sit>

Type "shutdown" and press <RETURN>. The reboot prompt or menu is then displayed.

## **MAI 2000 Disk Resident Diagnostics**

As noted above, the fixed disk resident diagnostics on the MAI 2000 consist only of logic tests for the floppy and MCS drives. An alternate load is required to access these diagnostics but the diagnostics floppy and MCS cartridge are not needed. Once the diagnostic executive is loaded, the test and error report can be sent to the printer as described above.

### **MCS Diagnostic**

To load and run the MCS diagnostic, do the following:

1. Shut down the system and press <CTRL>+<C> when the reboot prompt is displayed.

If the system is powered off, turn it on and press <CTRL>+<C> while the self-tests are being run.

2. Place an MCS cartridge in the drive. The cartridge should not be write protected so all tests can be run.
3. When you are asked for the boot device:

Boot device:

press <RETURN>.

4. When you are asked for the system file:

System file:

type "/sys/mcs" and press RETURN.

The system loads the diagnostic executive and the MCS logic tests. The program also checks the MCS drive to see if it is ready to be tested. The cartridge must be in place during this test. When the test is ready, it displays the test prompt:

<mcs>

5. To run the test, type "run" and press RETURN.

The system then runs a series of tests on the MCS drive, displaying a test number and brief description of each test as it is run. If any errors occur during a test, a description of the error is displayed. Make a note of any errors reported.

When all of the tests have been run, and report of the total number of errors is displayed. The MCS test prompt is then displayed again.

To shut down, type "shutdown" and press RETURN.

#### Floppy Disk Diagnostic

To load and run the floppy diagnostic, do the following:

1. Shut down the system and press <CTRL>+<C> when the reboot prompt is displayed.

If the system is powered off, turn it on and press <CTRL>+<C> while the self-tests are being run.

2. Place a floppy diskette in the drive. The diskette should not be write protected so all tests can be run.

3. When you are asked for the boot device:

Boot device:

press <RETURN>.

4. When you are asked for the system file:

System file:

type "/sys/floppy" and press <RETURN>.

The system loads the diagnostic executive and the floppy logic tests. The program also checks the floppy drive to see if it is ready to be tested. The floppy must be in place during this test. When the test is ready, it displays the test prompt:

<floppy>

5. To run the test, type "run" and press <RETURN>.

The system then runs a series of tests on the floppy drive, displaying a test number and brief description of each test as it is run. If and errors occur during a test, a description of the error is displayed. Make a note of any errors reported.

When all of the tests have been run, and report of the total number of errors is displayed. The floppy test prompt is then displayed again.

To shut down, type "shutdown" and press <RETURN>.

## ERROR NUMBERS

This section concludes with a list of error numbers that may be displayed in the course of operation. These errors are returned by the operating system kernel, the subroutine libraries, LAN and communications drivers. This is indicated by the "-" prefixed to each number.

Most of the time a message is displayed with the number, or a message may be displayed with no number. The following numbers and very brief explanations are offered in case a number alone is displayed.

These numbers are distinct from the Business BASIC error numbers displayed while in BASIC or while running a BASIC program. They do, however, often describe an error also described by a BASIC error number. BASIC error numbers are described in the Business BASIC Level 7 Reference Manual.

For on-line reference to the following numbers, a list very similar to the following list is contained in the file "/etc/comerr.list", and may be displayed using the "p" command:

```
@>p /etc/comerr.list
```

## Kernel Errors

- 1 Attempt to access beyond end of file
- 2 An I/O error occurred
- 3 A bad sector has been specified
- 4 Program is too large to fit in memory
- 5 File does not exist
- 7 Bad modes string in system-level open call
- 8 No more file descriptors free in the system
- 9 No more logical units are available to this process
- 10 An operation has been requested on a closed logical unit
- 13 Attempt to access nonexistent BFS device
- 14 A bad file name has been given to the file system
- 15 The file system on the specified device has the wrong format
- 17 Out of space on the file system specified
- 18 An object file being executed has a bad format
- 19 The requested operation requires a directory
- 20 Some directory in the pathname does not exist
- 21 Attempted to delete a non-empty directory
- 22 The specified file is a directory
- 23 An attempt was made to open a directory for writing
- 24 No more mount table space is available
- 25 The file specified is not a device file
- 26 The directory or device is busy
- 27 The specified device is not mounted
- 28 No more swap space is available
- 29 There are no more processes or the process does not exist
- 30 The specified logical unit is not that of a file
- 31 The file, directory or device is in use
- 32 A bad device was specified
- 33 A bad system call was made
- 34 An odd address was given to 'raw' I/O

- 35 An invalid byte count was given to 'raw' I/O
- 36 Cannot addname across filesystems
- 37 The specified file or directory already exists
- 38 The current process has no children
- 39 Trap instruction
- 40 The specified logical unit is not that of an eventcount
- 41 The request requires a 'double' eventcount
- 42 A bad logical unit number was specified
- 43 No more eventcall slots are available
- 44 The requestor does not have read permission
- 45 The requestor does not have write permission
- 46 There are temporarily no more pipe structures
- 47 An attempt was made to write to a pipe with no one reading
- 48 Access permission denied
- 49 An attempt was made to execute a file which has a bad header
- 50 A negative file position was specified
- 51 An attempt was made to map a file which has not been added
- 52 No shared data table entry available
- 53 The specified file is a remote file
- 54 A ctrl-X interrupt occurred
- 55 A bad mnemonic was encountered
- 56 An attempt was made to access a bad keyed file
- 59 The directory is in use
- 60 A seek error has occurred
- 61 Not enough memory
- 62 The requested operation has timed out
- 63 Write protected
- 64 A write error occurred formatting diskette
- 65 File descriptor has maximum number of names
- 66 Filesystem is mounted read-only
- 67 No more record lock entries are available

- 68 Waiting on the specified record would establish a deadlock
- 69 Bad parameter
- 70 The file is a special file system device
- 71 Bad key
- 72 Bad file system
- 73 A bad index was given to the file system
- 74 Data buffer too small
- 75 Bad file type
- 76 Attempt to access beyond beginning of file
- 77 Key doesn't exist
- 78 Duplicate key
- 79 Keyed file operation requires a current position
- 80 No record exists corresponding to specified key
- 82 Attempt to delete primary key description
- 83 Disk is already enabled
- 84 Disk is not enabled
- 85 End of tape
- 86 Drive seek error or data transfer error
- 87 Device minor number specified was a partition
- 88 Bad argument passed to system
- 89 Device off-line
- 90 File mark found on tape
- 91 End of data has been reached
- 92 Bad buffer size for a bfs write
- 93 Printer buffer overflow has occurred
- 94 Printer is out of paper
- 95 Printer is off-line
- 96 Async I/O is in progress
- 97 No cartridge
- 98 Tape media error
- 99 Originally mounted diskette not found
- 100 Disk was left mounted, override disabled
- 101 No clist
- 102 No disk in floppy disk drive
- 103 Unknown printer mnemonic
- 104 Duplicate directory
- 105 Too many directories given
- 106 TTY input buffer overflow
- 107 Printer out of ribbon

-108	Printer I/O transmission error
-109	Not a buffered device
-110	The printer is busy
-111	Parity error
-112	Attempt to execute an illegal instruction
-113	Bpt/trace
-114	Memory management error
-115	Odd address error
-116	Divide by zero error
-117	Check instruction
-118	Overflow condition caught
-119	Privilege violation
-120	Unknown trap
-121	The file is not locked
-122	Incorrectly initialized
-123	Bad port protocol specification
-124	Printer not initialized
-125	Too many printers configured
-126	Illegal serial device
-127	Illegal parallel device
-128	Printer interrupted
-129	Attempt to open an inactive ghost
-130	Bad slew channel definition
-32767	Nonexistent memory was specified for a system call
-32768	The current system call was terminated

### BFS Subroutine Library Error Codes

-200	The requested function is not available
-201	A bad logical unit number was specified
-202	No more logical units available for this process
-203	Parameter out of range
-204	File not locked
-205	File is locked
-206	The logical disk is not defined
-207	Can't open /etc/mounttab
-208	The requested operation requires a BASIC program file
-209	Program is public program, BUSY

-210 BASIC program file is too small  
-211 Data buffer too small, no memory  
-212 BASIC program has not been loaded  
-213 The file already exists  
-214 Bad public program id number  
-215 The file was not added to shared memory  
-216 File already addr'ed  
-217 Program not addred  
-218 Not a disk file  
-219 Bad date specified  
-220 Bad time specified  
-221 Bad time/date format  
-223 No SIZ= clause given  
-224 Terminal driver failure  
-225 I/O operation timed out  
-226 Control x hit  
-227 Console already opened  
-228 Audit file already started  
-229 Write via input  
-230 An attempt was made to access a bad keyed file  
-231 The operation requires BASIC program  
-232 ESCAPE and/or ^C interrupt occurred  
-233 ^Y interrupt occurred  
-234 Can only start a device  
-235 Terminal device is busy  
-236 Ghost I/O type conflict  
-240 The tape device is busy  
-241 The tape position is unknown  
-242 Illegal call or usage

### **BFS Spooling Error Numbers**

-251 Spooling is off - this one is for basic  
-252 Bad parameter in default file  
-253 Undefined default printer  
-254 Undefined printer  
-255 Undefined class name  
-256 Bad parameter in class file  
-257 Number of copies out of range  
-258 Illegal delete flag  
-259 Illegal notify flag

-260	Priority out of range
-261	Illegal requeue flag
-262	Illegal spool flag
-263	Illegal wait flag
-264	Undefined form name
-265	Bad page length
-266	Bad page width
-267	Bad slew line number
-268	Bad lines per inch
-269	Bad pitch
-270	Bad translation table name
-271	Bad single sheet
-275	Bad parameter in forms file
-280	Illegal translation file name
-281	Printer is open by another task
-282	General printer error
-285	Can't create spool file
-286	Can't open printer queue
-287	Can't read from this printer
-288	Bad normal timeout default
-289	Bad status timeout default
-290	Bad slave field

### **Ghost Error Codes**

-300	The ghost task is not ready to communicate
-301	Both bfs and ghost are trying to do the same type of i/o
-302	Ghost can be opened only once
-303	Not enough free space for requested i/o
-304	Siz= set by ghost and bfs sent to much data
-305	Somebody did not respond in time

## Local Area Network Errors

- 512 Invalid LAN network event log file
- 513 Can't obtain a dynamic socket
- 514 Can't obtain well known socket
- 515 Bad status from a send datagram
- 516 Bad status from a receive datagram
- 517 Can't open configuration table
- 518 Can't read configuration table
- 519 No log file name or no event type  
specified for monitoring
- 520 Can't open log file
- 521 Can't read log file
- 522 Can't open server ID file

## Virtual Circuit Errors

- 523 Illegal inter\_comm list size
- 524 Larger than 542 bytes
- 525 No more virtual circuits available
- 526 Illegal virtual circuit number
- 527 No open connection exists
- 528 Virtual circuit busy
- 529 No asynchronous op exists
- 530 No datagram arrived
- 531 Socket driver detected an error
- 532 Timeout occurred, closed circuit
- 533 Nonrecoverable error occurred
- 534 Virtual circuit in congestion
- 535 Invalid parameter value
- 536 Illegal datagram type detected
- 537 Invalid parameter
- 538 Sequence error detected
- 539 Invalid function value
- 540 Invalid size value
- 541 Invalid length value
- 542 Invalid option value
- 543 Buffer list encoding error
- 544 Protocol error detected
- 545 No such reference
- 546 Duplicated connection request
- 547 Negotiation failed

-548 Abnormal disconnect detected  
-549 Not enough space to receive data  
-550 Can't cancel asynchronous operation  
-551 CCT\_CONTROL parameter address = 0  
-552 CCT\_CONTROL illegal operation  
-553 Illegal maximum virtual circuit number  
-554 Timeout, circuit remains open  
-555 Network error, circuit remains open  
-556 Buffer length is negative  
-557 Virtual circuit is closing  
-558 ERR datagram received  
-559 EINTERRUPTED encountered  
-560 EINTERRUPTED, operation not completed  
-561 Operation was canceled  
-562 Kernel Virtual Circuit system error  
-563 Specified function does not exist

### **Remote Server Manager Errors**

-600 Server ID already defined  
-601 Server ID is not defined  
-602 Unknown server option  
-603 Server ID table is full  
-604 Error in create/open/read/write SIF  
-605 RSM failed in fexecl of server  
-606 RSM failed to fork server  
-607 Server name is too long ie > 127  
-608 Shared server status check failed  
-609 RSM could not open kernel memory  
-610 RSM restarted a failed shared server  
-611 RSM failed to send data to server

### **Gateway Errors**

-620 Gateway failed to start  
-621 Gateway out of buffers  
-622 Gateway error expanding net table  
-623 Gateway medium undefined  
-624 Gateway B4 block bad

## IAN Socket Driver Errors

- 651 No socket available
- 652 Specified Socket does not exist
- 653 Destination socket doesn't exit or is not open
- 654 Read Timeout
- 655 User buffer count out of range
- 656 No Driver buffer available
- 657 Socket in use by another process
- 658 Board not active or not available
- 659 Board does not exist
- 660 Bad pointer passed to driver
- 661 Bad special function
- 662 Receiver not reading socket
- 663 Ownership violation
- 664 Duplicate address
- 665 Board disable
- 666 Socket service interrupted by CTL-C or CTL-Y
- 667 Sum of buffer lengths out of range
- 670 Hardware generated error or retry count exhausted
- 671 Length too long for receive socket
- 672 Uninitialized socket
- 673 Invalid transmit control length
- 674 Invalid socket number
- 675 Receive socket busy
- 676 Destination host specification bad

## Remote File Server Errors Shared by COPY Server

- 690 Can't get memory for data buffer
- 691 Can't open system password file
- 692 User name not passed on open
- 693 User name not in password file
- 694 Unknown error on start up
- 695 Login of user failed
- 696 Illegal MCB command for remote file server
- 697 Unrecognized MCB command to RFS

## **Remote Copy Server Errors**

- 699 alloc failure in remote copy server
- 700 failure to create event count in copy server
- 701 failure to open event count in copy server

## **LAN Initialization Errors**

- 720 NET initialization failed
- 721 Fork/exec of RSM failed
- 722 Fork/exec of MONITOR failed
- 723 Fork/exec of GATEWAY failed
- 724 Board exists with station # of 0
- 750 MAGNET key needed, not installed
- 751 B4NET key needed, not installed
- 752 Can't get key from /etc/level file

## **Communication Errors**

### **Bisync Line Driver Errors**

- 760 X25 link reported an error
- 800 Read status word
- 801 No such command
- 802 Illegal command in current state
- 803 Device already open
- 805 Device not closed on release
- 806 No level file
- 807 Bad level file
- 808 Bad public key
- 809 Bad kernel line driver
- 810 Bad kernel memory allocation

## Synchronous Server Errors

- 890 Port is busy
- 891 LAN socket is not available
- 892 DD is already started
- 893 DD file is not found
- 894 DD undefined
- 895 SCS can not create new process
- 896 SCS received an illegal command
- 897 SCS is dying

## NOTES

**NOTE**

## APPENDIX A

### MINOR SYSTEM CONFIGURATION CHANGES

Some of the system configuration parameters should be changed when you are configuring an additional printer or allowing for additional concurrent users (usually coincident with an additional terminal). These parameters make additional resources, particularly memory, available to the system to maintain high performance when additional devices are used.

The configuration file for the normal system startup is contained in the file "/etc/conf" on the boot partition. This is the file that must be modified to support the added system load.

The configuration file is originally set up to support only a minimal system. It was probably modified by your service representative or dealer for your configuration. The changes described in this appendix are intended for further additions you make following the initial installation of the system.

Full information on configuring the system is contained in the BOSS/IX Technical Reference Manual (M6226).

Follow these steps to modify your configuration file:

1. Log on as the system administrator.
2. To display your current system configuration, execute this command:

ADMIN>vconf -os

(Note that this assumes your system is currently booted with the normal configuration.)  
The list of configuration parameters and values fills most of the terminal screen.

3. Find these parameters in the display, and note the current value.
  - o Number of system buffers
  - o Number of processes allowed
  - o Maximum number of lu's
  - o Maximum number of open files
  - o Maximum number of shared text segments
  - o Number of dynamic LAN sockets
  - o Number of LAN buffers
  - o Maximum number of file locks
  - o Maximum number of eventcalls
  - o Maximum number of printers
4. Mount the boot filesystem, "/dev/boot".
5. For each printer being added, you need to increase the values of the following parameters in the indicated amounts:
  - o Number of processes, increase by 1
  - o Maximum number of lu's, increase by 6
  - o Maximum number of open files, increase by 2
  - o Number of printers, increase by 1
  - o Number of directory caches, increase by 2  
(MAI 300 only)

6. For each concurrent user being added, you need to increase the values of the following parameters in the indicated amounts:
  - o Number of system buffers, increase by 4 on the MAI 2000, increase by 64 on the MAI 3000
  - o Number of processes, increase by 6
  - o Number of shared text segments, increase by 4
  - o Maximum number of file locks, increase by 5
  - o Maximum number of lu's, increase by 20
  - o Maximum number of open files, increase by 8
  - o Maximum number of eventcalls, increase by 5
  - o Number of directory caches, increase by 8 (MAI 3000 only)

7. Add the recommended increases for each parameter to its current value. Using these values, enter the following command:

```
ADMIN>vconf /mnt/etc/conf -save buffers=#  
tsegs=# locks=# procs=# lus=# opens=#  
eventcalls=# printers=# dirc=#
```

where # is the size you have calculated for each parameter.

For example, if your system originally matched the base configuration and you are adding one printer, the command would be (approximately):

```
ADMIN>vconf /mnt/etc/conf -save procs=21  
lus=70 opens=42 printers=2
```

7. Shutdown and reboot the system.

This completes the procedure. Note that the modified configuration does not take effect until the system has been fully shut down and re-booted, as stated in step 7.

NOTE

## APPENDIX B

### SYSTEM FILE FORMATS

#### INTRODUCTION

Several system files are in a format that allows them to be modified by the text editor, VED. Although these files are usually maintained by various utilities, there are occasions when they must be modified directly via the editor. The formats of these files are described below.

In all cases, the files are made up of one or more records, each record consisting of several fields. The fields within each record are separated by colons (:). The records within each file are separated by the carriage return character, CHR(10).

#### SYSTEM DEFINITION FILE FORMATS

This group of files contains system device and user definitions.

##### Operator Definitions File

The file "/etc/passwd" defines operator logon security, and is maintained by the Operator Information Utility. The file consists of seven fields in the following format:

name:pswd:usr#:rsvd:cods:home:ipgm

where:

name = user's logon name

pswd = encrypted password, cannot be usefully changed

usr# = user number  
rsvd = reserved, must be empty  
cods = menu access codes  
home = home directory  
ipgm = initial program

### **Port Definitions File**

The file "/etc/ports" defines the devices on the CMB and 4/8 way ports. The file consists of 6 fields in the following format:

**prt#:dfil:name:M/m:def:cmnt**

where:

**prt#** = the port number

**dfil** = the device type definition file, e.g.,  
"/etc/terminals"

**name** = the device name in the "/dev" directory

**M/m** = Major and Minor device numbers

**def** = TBD

**cmnt** = an uninterpreted comment field

## Printer Definitions File

The file "/etc/printers" defines system printers.  
The file consists of 10 fields in the following  
format:

name:pque:dvnm:hpri:type:B4:spl:rtim:wtim:slv

where:

name = the printer name

pque = the print queue directory,  
"/etc/\_qtempo"

dvnm = the printer device file name,  
"/dev/name"

hpri = the printer hold priority

type = the printer type, e.g., "tp80"

B4 = "y" or "n", for Basic Four Interface

spl = "y" or "n" for spooling turned "on" or  
"off"

rtim = duration of read timeout

wtim = duration or write timeout

slv = "y" or "n" indicating whether or not  
the printer is a slave

The printer definition file may also contain two-field records defining a printer alias. The format for these records is:

alias:pname

where alias is the alternate name (alias) of the printer and pname is the printer name as defined earlier in the file. For example:

LP:lp

gives the printer "lp" an upper case name, "LP", for access from BASIC.

## **Terminal Definitions File**

The file "/etc/terminals" defines terminals on the system. The file consists of 8 fields in the following format:

name:type:dvfl:slv:strt:itrn:otrn:rsvd

where:

name = two character terminal name, e.g., T0

type = terminal type, e.g., evdt

dvfl = the device file in "/dev"

slv = device file for slave printer

strt = terminal start process

itrn = input translation table

otrn = output translation table

rsvd = reserved

## **PRINTER PARAMETER FILE FORMATS**

This group of files contains printer parameter definitions.

## Defaults File

The "/etc/defaults" file has nine fields per record in the following format:

name:pri:form:cps:del:req:ntfy:spl:wait

where:

name = "system", a user name or a printer name

pri = priority, an integer between 0 and 9

form = the name of a form

cps = the number of copies to print

del = "y" or "n", to delete or not delete the file after printing

req = "y" or "n", to requeue or not after printing

ntfy = "y" or "n", to notify after printing

spl = "y" or "n", to spool or not

wait = "y" or "n", to wait until entire file is received before despooling, or not

Any of these fields except the first may be empty.

Here are some examples of possible entries:

system:5:stanrd:1:n:n:n:y:n

kathy:8::::::::::

lp::::::::::y:y

## **Class File**

The "/etc/class" file defines print job classes. The format is the same as for the "/etc/defaults" file, described above:

```
name:pri:form:cps:del:req:ntfy:spl:wait
```

Some sample entries are:

```
splon:5:stanrd:1:n:n:n:y:n
sploff:5:stanrd:1:n:n:n:n:n
catlog:catlog:10::::y:n
```

## **Print Forms File Format**

The "/etc/forms" file has 16 fields per record in the following format:

```
name:lnth:wdth:s11:s12:s13:s14:s15:s16:s17:s18
:
lp1:ptch:tran:feed:ptrs
```

where:

**name** = the form name

**lnth** = the number of lines per page

**wdth** = width of the form in characters

**s11** = value for slew channel 1, always set to  
1

**s12** through **s18** = values for slew channels 2  
through 8

**lp1** = number of lines per inch

**ptch** = pitch

**tran** = translation table name

**feed** = "y" or "n", for hand sheet feed or not

**ptrs** = the names of printers, separated by a  
single space, indicating the printers  
that can use the form. An asterisk  
(\*) indicates all printers.

(Note that the record will generally be displayed on a single line in the VED display.)

The system comes with a "standard" form entry in the "/etc/forms" file called "stanrd" that looks like this:

```
stanrd:66:80:1:::::::6:10:S8:*
```

Thus the name is "stanrd": length is 66 line, width is 80 characters, slew 1 is 1 (top of form), lines per inch is 6, pitch is 10, translation table is "S8", and the asterisk indicates that all printers may use this form. The empty fields are not specified and are optional. For example:

```
draft:66:80:1:::::::6:10:S8:LP P0
memo:66:80:1:::::::6:12:S7:P1
```

## NOTES

## **INDEX**

Background Processing  
(see Command Language), 5-10

Backup  
    Cartridge, 4-110  
Compare Cartridge Utility, 4-121  
Compare Cartridge Command, 5-129  
Floppy Diskette, 4-125  
Label Cartridge, 4-123  
MCS, 5-139  
MTS, 5-159  
Report Cartridge Contents, 4-119  
Report Diskette Contents, 4-134  
Utilities, 4-109

BASIC  
    Compare Programs Utility, 4-146  
    Display Program Names, 4-152  
    Encrypt Program Utility, 4-155  
    Merge Programs Utility, 4-149  
    Program Cross-reference, 4-136  
    Program Renumber Utility, 4-139  
    Search & Replace Utility, 4-142

BFS (See File System), 2-1

Change Password Command, 5-62  
Command Files (see Command Language), 5-17  
Command Interpreter (see Command Language), 5-1  
Command Language, 5-1  
    Arguments, 5-6, 5-16  
Background Processing, 5-10  
Batch Files, 5-17  
Case Sensitivity, 5-14  
Command Files, 5-17, 5-56  
Command Interpreter, 5-1  
Commands (Listed Alphabetically), 5-35  
Control Characters, 5-3  
Environment, 5-19

**Command Language (cont'd)**

Pattern Matching, 5-11, 5-17  
Pound-sign Commands, 5-19  
Process Redirection, 5-8  
Prompt, 5-2  
Punctuation, 5-14  
Redirection, 5-16  
Syntax, 5-4, 5-21  
Vocabulary, 5-4

**Date Utility, 4-160**

**Directories (see also File System), 2-4**

Access Rights, 2-20, 4-35  
Alternate Directories, 2-12, 5-41  
Change Working Directory, 4-33, 5-50  
Create Command, 5-116  
Create Utility, 4-29  
Delete Utility, 4-31  
Display Command, 5-111  
Display Utility, 4-26  
Dot Abbreviations, 2-13  
Mount Point, 2-7  
Names, 2-8  
Path Names, 2-11  
Root, 2-7  
Search Order, 2-14  
Security Utility, 4-35  
Security, 2-20, 5-85  
Tree Structure, 2-4  
Utilities, 4-25  
Working Directory, 2-12, 5-50

**Disk**

Cylinders, 2-21  
Partitions, 2-23  
Sectors, 2-21  
Tracks, 2-21

**File System**, 2-1  
Access Rights, 2-20  
BFS, 2-1  
Change File Size, 2-25  
Commands (see also Commands), 2-1, 2-27  
Create Directory Command, 5-116  
Directories, 2-4  
Disk Maintenance, 2-21  
Disk Organization, 2-21  
File Access, 2-8, 2-17  
File Names, 2-8  
File Repair, 4-66  
File Types, 2-6  
Files, 2-3  
Filesystem Utilities, 4-71  
Error Analysis Utility, 4-84  
Filesystems, 2-7, 2-23  
Fixed Disk, 2-3  
Floppy Diskette, 2-3, 2-7, 2-15, 2-23  
Format Diskette Utility, 4-76  
Free Space, 2-24, 4-81  
Hierarchical Structure, 2-4  
List Directory Command, 5-111  
Maintenance, 2-19  
Mount Command, 5-144  
Mount Point Directory, 2-7  
Mount Utility, 4-73  
Mounting, 2-7, 2-15  
Organization, 2-4  
Ownership, 2-19  
Partitions, 2-23  
Path Names, 2-11  
Read-only Mounting, 2-18  
Repair Procedure, 6-10  
Repair, 2-26, 4-84  
Root Directory (/), 2-7  
Search Order, 2-14  
Security, 2-19  
Sub-directories, 2-4  
Tape Devices, 2-3  
Tree Structure, 2-4

**File System (cont'd)**  
Unmount Utility, 4-73  
Unmount Command, 5-25  
Utilities (see also Utilities), 2-26  
**Files (see also File System)**, 2-3  
Access Rights, 2-20, 5-85  
Accessing, 2-8, 2-17  
Change Owner, 5-55  
Change Size, 2-25, 4-55  
Compare Command, 5-73  
Concatenate, 5-47  
Copy Command, 5-58  
Copy Utility, 4-50  
Create Utility, 4-41  
Delete Command, 5-69  
Delete Utility, 4-44  
Differences, 5-73  
Display File Information Utility, 4-46  
Display, 5-163  
Eventcount, 5-46  
File Selection Environment, 4-20  
Filelist Utility, 4-58  
Formatted Display, 5-167  
Links, 5-43  
Lost Files, 6-21  
Move Command, 5-146  
Move Utility, 4-53  
Names, 2-8, 5-43  
Ownership, 2-19  
Pattern Search, 5-127  
Pattern Substitution, 5-52  
Rename Utility, 4-48  
Repair Procedures, 6-20  
Repair Command, 5-96  
Repair Utility, 4-66  
Security Utility, 4-64  
Security, 2-19, 5-85  
Types, 2-6

Filesystems (see also File System), 2-7, 2-23  
    Free Space, 5-75  
    Repair, 5-87  
    Usage, 5-75  
    Creating, 2-23  
    Example, 2-9  
    Mounting, 2-7, 2-15  
    Repair, 2-26

Floppy Diskette (see also File System), 2-3  
    Copy Diskette Utility, 4-78  
    Format Command, 5-71  
    Format Utility, 4-76  
    Formatting, 2-23  
    Mount, 5-144  
    Unmount, 5-25  
    Mounting, 2-7, 2-15

Magnetic Cartridge Streamer (see MCS)

Magnetic Tape Streamer (see MTS)

MCS  
    Compare to Disk, 5-129  
    Label Cartridge Utility, 4-123  
    Label Command, 5-131  
    List Contents, 5-133  
    Menu Editor Utility, 4-173

MTS  
    Compare to Disk, 5-149  
    Label Command, 5-151  
    List Contents, 5-153

Operator Information Utility, 4-183

Port Configuration Utility, 4-164

Printing (see also Spooler), 3-1  
Form Definitions, 3-10  
Change Form Utility, 4-95  
Maintain Print Queue Utility, 4-93  
Printer Maintenance, 5-100  
Printer Parameters Utility, 4-98  
Printer Status Utility, 4-97  
Problems, 6-24  
Submit Job Command, 5-106  
Submit Print Job Utility, 4-90  
Utilities, 4-89  
Problem Solving, 6-1  
Filesystem Repair, 6-10  
Memory Dump, 6-5  
Premature Shutdown, 6-3  
System Crash, 6-3

Redirection (see Command Language)

Restore

Cartridge, 4-115  
Floppy Diskette, 4-130  
MCS, 5-135  
MTS, 5-155  
Report Cartridge Contents, 4-119  
Utilities, 4-109

Set Date, 5-62

Spooler

Advantages, 3-4  
Auto Form Feed, 3-15  
Bypassing, 3-14  
Change Form Utility, 4-95  
Commands, 3-3, 3-16  
Configuration, 3-12  
Default Files, 3-2, 3-5, 4-98  
Description, 3-3  
Despooler, 3-3  
Display Queue Command, 5-104  
Form Definition, 3-10  
Form Parameters, 3-11  
Forms File, 4-98  
Introduction, 3-1

**Spooler (cont'd)**

Maintain Print Queue Utility, 4-93  
ON/OFF, 3-13  
Parameters, 3-5  
Printer Maintenance Command, 5-100  
Printer Status Command, 5-110  
Printer Status Utility, 4-97  
Problems, 6-24  
Submit Print Job Command, 5-106  
Submit Print Job Utility, 4-90  
Utilities, 3-3, 3-16, 4-89

**Text Editor, 5-208**

**Time Utility, 4-160**

**Utilities**

Accessing, 4-5  
Backup, 4-109  
BASIC Cross-reference, 4-136  
BASIC Program, 4-135  
BASIC Renumber, 4-139  
BASIC Search & Replace, 4-142  
Change File Size, 4-55  
Change Print Form, 4-95  
Change Working Directory, 4-33  
Command Access, 4-7  
Compare BASIC Programs, 4-146  
Compare Cartridge Utility, 4-121  
Copy Diskette Utility, 4-78  
Copy File, 4-50  
Create Directory, 4-29  
Create File, 4-41  
Date and Time, 4-160  
Delete Directory, 4-31  
Delete File, 4-44  
Directory Display, 4-26  
Directory Security, 4-35  
Directory, 4-25  
Display BASIC Programs, 4-152

## Utilities (cont'd)

Encrypt BASIC Program, 4-155  
File Information, 4-46  
File Repair, 4-66  
File Security, 4-64  
File Selection Environment, 4-20  
File Selection, 4-17  
File Utilities, 4-39  
Filelist Maintenance, 4-58  
Filesystem and Disk, 4-71  
Filesystem Error Analysis, 4-84  
Filesystem Repair, 6-13  
Format Diskette, 4-76  
Free Space Analysis, 4-81  
Input Prompts, 4-13  
Label Cartridge, 4-123  
Maintain Print Queue, 4-93  
Menu Access, 4-5  
Menu Editor, 4-173  
Menu Prompts, 4-11  
Menus, 4-9  
Merge BASIC Programs, 4-149  
Mount Filesystem, 4-73  
Move File, 4-53  
Nonoperational Keys, 4-16  
Operator Information, 4-183  
Pattern Matching, 4-23  
Port Configuration, 4-164  
Printer Parameters, 4-98  
Printer Status, 4-97  
Printer, 4-89  
Prompts, 4-10  
Rename File, 4-48  
Report Cartridge Contents, 4-119  
Report Save Diskette Contents, 4-134  
Restore from Cartridge, 4-115  
Restore from Diskette, 4-130

**Utilities (cont'd)**

- Save and Restore, 4-109
- Save to Cartridge, 4-110
- Save to Diskette, 4-125
- Special Function Keys, 4-14
- Standard Procedures, 4-5
- Submit Print Job, 4-90
- System, 4-159
- Toggle Prompts, 4-11
- Unmount Filesystem, 4-73
- Verification Prompts, 4-14

NOTE