intel®

NDS-II ISIS-III(C) USER'S GUIDE

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

Order Number: 122098-002

P.O. Box 125
Las Piedras, Puerto Rico 00671
(809) 733-3030 / 746-8080 / 746-8484
TELEX 325 2574

Intel Puerto Rico Inc.

ISIS-MDS.COM
EBAY JUNE 16, 2016

Dear Customer:

Thank you for your order. You have purchased a quality Intellec Microcomputer Development System upgrade designed to provide an efficient software development work station on your NDS-II.

Your new ISIS Cluster Board will work with Series II, Series III, Series IV or Model 800 development systems. Before installing your cluster board, please read the iMDX 581 Installation, Operation, and Service Manual (#122293) packaged with your board.

The NDS-II ISIS-(C) User's Guide Supplement (#122098), is also included with your cluster board. This manual provides operating instructions for any ISIS Cluster work station, regardless of the type of host system in which the Cluster Board resides. Likewise, the Product Release Notes in this package apply to all ISIS Cluster work stations.

We believe that this ISIS Cluster board is a reliable product which will provide trouble free operation. However, if you should experience any difficulties, call the nearest Intel Service Center.

If you have any questions, call your local Intel Sales Office. Intel offers a comprehensive network of support services, including a world-wide maintenance force of customer engineers, regional product specialists, and factory support experts. In addition, training classes on a wide variety of development tools are offered in the U.S., Europe and Japan.

Sincerely,

Pablo R. Rodriguez,
General Manager

Intel

product release notes

The following terminals have been tested and found to be interface compatible with the ISIS Cluster Board; configuration files are provided for these terminals. Customers are advised to select terminals to meet their own special environmental specifications.

Hazeltine, Model 1510
Televideo, Model 910+, 925, 950
Lear Seigler, Model ADM 3A
Adds Viewpoint, Model 3A+
Qume, Model 102

ISIS Cluster Board Power Consumption

The iMDX-581 power requirement us +5VDC @ 4.5A maximum; typical power consumption is +5VDC @ 2.6A

DJC and iNDX r2.8

Special instructions for using DJC and ISIS with iNDX r2.8 can be found in ISIS-III(N)/III(C) Product Release Notes (#121940). This document is shipped with the iMDX-455 Work Station Upgrade Kit, the iMDX-450 Network Resource Manager, and the iMDX-450-U28 NDX-II Software Update.

NDS-II ISIS-III(C) USER'S GUIDE

Order Number: 122098-002

Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

Intel retains the right to make changes to these specifications at any time, without notice. Contact your local sales office to obtain the latest specifications before placing your order.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

The following are trademarks of Intel Corporation and its affiliates and may only be used to identify Intel products:

BITBUS	i_m	iRMX	Plug-A-Bubble
COMMputer	iMMX	iSBC	PROMPT
CREDIT	Insite	iSBX	Promware
Data Pipeline	int_el	iSDM	QueX
Genius	int_elBOS	iSXM	QUEST
i	Intelevision	Library Manager	Ripplemode
$\hat{i}$	$int_eligent$ Identifier	MCS	RMX/80
I^2ICE	$int_eligent$ Programming	Megachassis	RUPI
ICE	Intellec	MICROMAINFRAME	Seamless
iCS	Intellink	MULTIBUS	SOLO
iDBP	iOSP	MULTICHANNEL	SYSTEM 2000
iDIS	iPDS	MULTIMODULE	UPI
iLBX			

Ethernet is a trademark of Xerox Corportion.

A1291 / 1184 /700 / DD / KH

SOFTWARE

REV.	REVISION HISTORY	DATE	APPD.
-001	Original issue. Support **V2.0** of ISIS-III(N) and the new features: —up to sixteen workstations —multiple Winchester disk storage capacity at the Network Resource Manager —QUEUE, SPACE, and WHO Software required: —ISIS-III(N) **V2.0** or later —ISIS-III(C) **V1.0** or later —iNDX.G11 **V2.0** or later —RUN **V2.1** or later —AEDIT **V1.0** or later —IXREF **V1.3** or later	3/83	P.D.
-002	Support **V2.2** of ISIS-III(N) and the new feature of Series IV as an ISIS Cluster host. Software required: —ISIS-III(N) **V2.2** or later —ISIS-III(C) **V2.2** or later —iNDX.G11 **V2.8** or later —RUN **V2.1** or later —AEDIT **V2.0** or later —IXREF **V1.3** or later	6/84	P.D.

PREFACE

The *NDS-II ISIS-III(C) User's Guide* provides operating instructions unique to the ISIS-III(C) operating system on a Network Development System-II (NDS-II). ISIS-III(C) executes on a Series IV, Series III, Series II, and Model 800 Cluster workstation. This manual assumes that you

- Have read the *NDS-II Network Development System Overview* (121761).
- Understand how to use the Intellec Microcomputer Development System Host.
- Have read the *Intellec® Series IV Operating and Programming Guide* (121753) for Series IV workstations, the *Intellec® Series III Microcomputer Development System Console Operating Instructions* (121609) for Series III workstations, or the *ISIS-II User's Guide* (9800306) for Series II or Model 800 workstations.
- Have read the *NDS-II ISIS-III(N) User's Guide* (121765) for information on the ISIS-III(N) operating system for Series III, Series II, and Model 800 host workstations.

For installation and service information for Series IV Cluster workstations refer to the *Series IV iMDX 580/582 ISIS Cluster Board Package Installation, Operation, and Service Manual* (134650). *iMDX 580 ISIS Cluster Board Package Installation, Operation, and Service Manual* (122100) provides installation and service information for Series III, Series II, and Model 800 Cluster workstations.

System operation requires version 2.2 or later of the ISIS-III(C) operating system software.

This manual contains seven chapters and three appendixes:

- Chapter 1, "Overview," profiles the NDS-II ISIS Cluster workstation.
- Chapter 2, "NDS-II Directory Structure," describes the NDS-II distributed file system and how it differs from the ISIS file structure.
- Chapter 3, "Guidelines for Creating and Maintaining NDS-II Files," offers guidelines and suggestions for establishing and tailoring directory trees to a particular application or project using the NDS-II file structure.
- Chapter 4, "Cluster Workstation Commands," defines and provides examples of Cluster workstation commands necessary to access the network.
- Chapter 5, "Cluster Workstation System Calls," describes the changes to existing ISIS system calls and new system calls designed to take advantage of the network capabilities.
- Chapter 6, "System Considerations," offers suggestions to improve system performance.
- Chapter 7, "Error Messages," lists and explains all new ISIS-III(C) error messages and indicates appropriate user action.
- Appendix A, "Summary of Error Messages," lists all error codes and messages issued by ISIS-III(C).
- Appendix B, "Summary of ISIS-III(C) Command Syntax," lists ISIS-III(C) commands and their syntax alphabetically.
- Appendix C, "Summary of ISIS-III(C) Devices," lists the network devices that are supported by the ISIS-III(C) operating system.

Related Publications

For more information on the NDS-II Network Development System refer to the following manuals:

- *NDS-II Network Development System Overview* (121761)
- *NDS-II Network Resource Manager User's Guide* (134300)
- *iMDX 455 Network Workstation Upgrade Kit Installation and Checkout Manual* (121882)

For installation and service information for a Series IV Cluster workstation refer to the following:

- *Series IV iMDX 580/582 ISIS Cluster Board Package Installation, Operation, and Service Manual* (134650)

For installation and service information for a Series III, Series II, or Model 800 Cluster workstation refer to the following:

- *iMDX 580 ISIS Cluster Board Package Installation, Operation, and Service Manual* (122100)

For more information on the ISIS-III(N) operating system for a Series III, Series II, or Model 800 host operating system refer to the following:

- *NDS-II ISIS-III(N) User's Guide* (121765)

For information on a Series IV workstation refer to the following:

- *Intellec® Series IV Microcomputer Development System Overview* (121752)
- *Intellec® Series IV Operating and Programming Guide* (121753)
- *Intellec® Series IV ISIS-IV User's Guide* (121880)

For information on a Series III workstation refer to the following:

- *Intellec® Series III Microcomputer Development System Product Overview* (121565)
- *Intellec® Series III Microcomputer Development System Console Operating Instructions* (121609)
- *Intellec® Series III Microcomputer Development System Programmer's Reference Manual* (121618)
- *ISIS-II User's Guide* (9800306)

For more information on Series II or Model 800 workstations refer to the following:

- *ISIS-II User's Guide* (9800306)

Notational Conventions

UPPERCASE	Characters shown in uppercase must be entered in the order shown. You may enter the characters in uppercase or lowercase.
italic	Italic indicates a meta symbol that may be replaced with an item that fulfills the rules for that symbol. The actual symbol may be any of the following:
directory-name	Is that portion of a *pathname* that acts as a file locator by identifying the device and/or directory containing the *filename*.

filename	Is a valid name for the part of a *pathname* that names a file.
pathname	Is a valid designation for a file; in its entirety, it consists of a *directory* and a *filename*.
V*x.y*	Is a generic label placed on sample listings where the version number of the product that produced the listing would actually be printed.
[ ]	Brackets indicate optional arguments or parameters.
{ }	One and only one of the enclosed entries must be selected unless the field is also surrounded by brackets, in which case it is optional.
{ } . . .	At least one of the enclosed items must be selected unless the field is also surrounded by brackets, in which case it is optional. The items may be used in any order unless otherwise noted.
. . .	Ellipses indicate that the preceding argument or parameter may be repeated.
punctuation	Punctuation other than ellipses, braces, and brackets must be entered as shown. For example, the punctuation shown in the following command must be entered: `SUBMIT PLM86(PROGA,SRC,'9 SEPT 81')`
`input lines`	In interactive examples, user input lines are printed in white on black to differentiate them from system output.
<cr>	Indicates a carriage return.
shading	Shading in Chapters 4 and 5 highlights the important differences between ISIS-III(C) software and previous versions of ISIS-II.

CONTENTS

CHAPTER 1 OVERVIEW PAGE

The NDS-II Network 1-1
The NDS-II Workstation 1-1
Cluster Workstation 1-1
Host Workstation 1-3
NDS-II Terminology 1-3
Hardware 1-3
Communications Controller Board Set 1-3
Intellink™ Communications Module 1-3
Files and Directories 1-3
Data Files 1-5
Directory Files 1-5
Local Files 1-5
Private Files 1-5
Public Files 1-5
Remote Files 1-5
Shared Files 1-5
Hierarchical File Structure 1-5
Remote Job Execution 1-5
Private Workstation 1-6
Public Workstation 1-6
Job Queues 1-6
Network Access 1-6
Device Access 1-6
File Access 1-6
Software Preparation 1-6
Logging On 1-7
Entering Commands 1-8
Logging Off 1-8
Superuser 1-9
Username and Password 1-9

CHAPTER 2 NDS-II DIRECTORY STRUCTURE

ISIS-II File Structure 2-2
NDS-II File Structure 2-2
Types of Files 2-2
User File Protection 2-2
Pathnames 2-2
Accessing a File 2-3
Creating Directories 2-3

CHAPTER 3 GUIDELINES FOR CREATING AND MAINTAINING NDS-II FILES

Directory Maintenance Guidelines 3-1
Logical System Root 3-1
Volumes 3-1
Directory Files 3-1
Home Directory 3-3
Using the Network 3-3

CHAPTER 4 CLUSTER WORKSTATION COMMANDS PAGE

Differences Between ISIS-III(C) and ISIS-III(N) Operating Systems 4-1
Differences Between ISIS-III(C) and ISIS-II Operating Systems 4-1
Directory Structure 4-1
Hard Disk 4-2
Remote Printer 4-2
Accessing the NDS-II Network 4-2
Command Program Compatibility 4-2
Remote Job Execution 4-2
Command Categories 4-2
Network Access Commands 4-3
Directory Maintenance Commands 4-3
File Maintenance Commands 4-3
Wild-Card Filenames 4-3
8080/8085 Program Execution Commands 4-4
Remote Job Execution Commands 4-4
Public and Private Workstations 4-4
Creating and Naming Queues 4-5
Program Control and Code Conversion Commands 4-5
Entering Commands 4-5
Command Syntax 4-6
Specifying Files 4-7

CHAPTER 5 CLUSTER WORKSTATION SYSTEM CALLS

Differences Between ISIS-III(C) and ISIS-II 5-1
Revised System Calls 5-1
New System Calls 5-1
Summary of System Calls 5-1
System Call Syntax and Usage 5-2
PL/M-80 Calls 5-2
Assembly Language Calls 5-2

CHAPTER 6 SYSTEM CONSIDERATIONS

Mainframe Link 6-1
ISIS-III(C) Command Files 6-1
Home Directory 6-1

CHAPTER 7 ERROR MESSAGES

New ISIS-III(C) Error Messages 7-1

APPENDIX A SUMMARY OF ERROR MESSAGES

ISIS-III(C) Error Messages (8080/8085 Mode) A-1

Console Command Interface Errors
(8080/8085 Execution Mode) A-2

APPENDIX B
SUMMARY OF ISIS-III(C)
COMMAND SYNTAX

APPENDIX C
SUMMARY OF ISIS-III(C) DEVICES

INDEX

FIGURES

FIGURES	TITLE	PAGE
1-1	NDS-II Network Development System Using an Intellink™ Communications Module	1-2
1-2	NDS-II Network Development System Connected to an Ethernet Cable	1-4
1-3	ISIS Cluster Configuration Status	1-7
2-1	ISIS Directory Structure	2-2
2-2	Hierarchical File Structure	2-3
3-1	Model of NDS-II File System	3-2
4-1	Sample Network File Structure	4-13
4-2	Hierarchical File Structure	4-35

TABLES

TABLE	TITLE	PAGE
4-1	Network File Access Rights	4-8
4-2	ISIS-III(C) Default Configuration	4-11
A-1	Nonfatal Error Numbers Returned by System Calls	A-3
A-2	Fatal Errors Issued by System Calls	A-3

CHAPTER 1
OVERVIEW

This chapter provides an overview of the Network Development System-II (NDS-II), describes the functions of the Cluster workstation and the ISIS-III(C) operating system, and explains the principal features of the Network.

The NDS-II Network

The NDS-II is an Ethernet-based interconnection of development systems. This connection allows multiple users to share concurrent access to a central hard disk(s), public workstations, and an optional spooled line printer. The NDS-II Network offers a remote network hard disk sub-system.

For more information on the NDS-II Network, see the *NDS-II Network Development System Overview* (121761).

The NDS-II Workstation

Cluster Workstation

An ISIS Cluster workstation is a user-supplied terminal that functions as a development system by using the resources of the NRM and the Cluster processor board. The Cluster workstation must be connected to an Intellec development system that acts as its "host." The Cluster workstation and the host communicate through the iMDX 580 ISIS Cluster Board that resides in the cardcage or an expansion chassis of the host.

Communication Controller Boards that reside in the NRM and the host enable the host and the Cluster workstation to communicate with the NRM.

The development system workstations are connected to the NRM via the Intellink Communications Module (see Figure 1-1). The development system workstations also can be connected to the Network via an Ethernet cable. A single transceiver connects the Intellink Communications Module to the Ethernet cable (see Figure 1-2).

The ISIS-III(C) operating system enables the Cluster workstation to logically connect to the NRM. After the Cluster workstation successfully boots off of the host, ISIS-III(C) is invoked at the Cluster workstation. ISIS-III(C) allows the Cluster workstation to function as a development system on the Network. ISIS-III(C) allows the Cluster workstation to access the following network resources:

- Network-based hierarchical file system compatible with ISIS format
- Controlled access to the network file system through user identification and verification
- Distributed job control (DJC) that allows user-specified batch jobs to be executed on idle development systems that have been made public with the IMPORT command
- Access to a network line printer queue (:SP:)
- A utility to monitor the network communication system

The user at the Cluster workstation does not have access to local files or monitor DEBUG facilities.

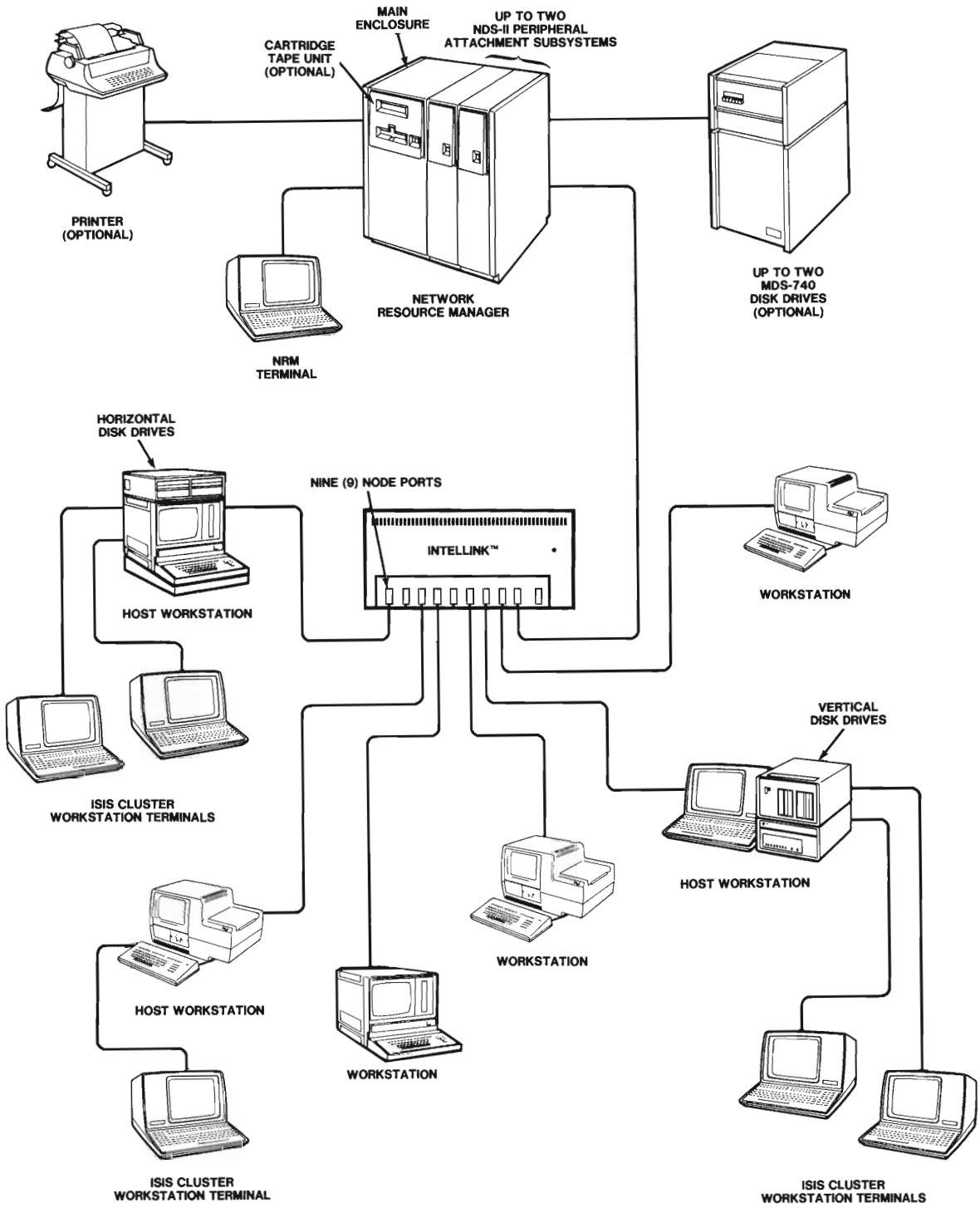


Figure 1-1. NDS-II Network Development System Using an Intellink™ Communications Module

122098-1

Host Workstation

Series IV, Series III, Series II, and Model 800 Intellec development systems can function as ISIS Cluster hosts. Series IV host workstations operate under the iNDX operating system. Series III, II, and Model 800 host workstations execute under the ISIS-III(N) operating system. All Cluster workstations operate under the ISIS-III(C) operating system.

The iNDX and ISIS-III(N) operating systems enable all network workstations to access shared network files, the network line printer, and distributed job control.

For any tasks that do not require network resources, the host workstation functions as a standalone development system even when the Cluster workstations are logically connected to the Network. When network resources are required, the host workstation can be logged on to the Network. While logically connected to the Network, the host operates as a standard development system.

NDS-II Terminology

The *NDS-II Network Development System Overview* (121761) has a complete glossary of NDS-II terminology. A more detailed description of some terms frequently encountered in ISIS-III(C) follows.

Hardware

The *iMDX 455 Ethernet Workstation Upgrade Kit Installation and Checkout Manual* (121882) describes the hardware connection of the host development system workstation and the NRM.

The *iMDX 580 ISIS Cluster Board Package Installation, Operation, and Service Manual* (122100) describes the hardware connection of the Cluster board and the ISIS-III(N) host workstation.

For installation and service information for a Series IV Cluster workstation refer to *Series IV iMDX 580/582 ISIS Cluster Board Package Installation, Operation, and Service Manual* (134650).

Communications Controller Board Set

The NDS-II Communications Controller Board Set, located in the development system workstation, and the Network Resource Manager enable all of the workstations to communicate with each other through the Intellink Communications Module.

Intellink™ Communications Module

The Intellink Communications Module is a self-contained hardware box that provides an Ethernet compatible interconnection. It supports collision detect, 10 Mbps operation, and has an independent power supply. It can be used as a standalone system or it can be connected to a full network.

Files and Directories

The distributed file system of NDS-II introduces several ways to describe network files. These new terms are defined as follows.

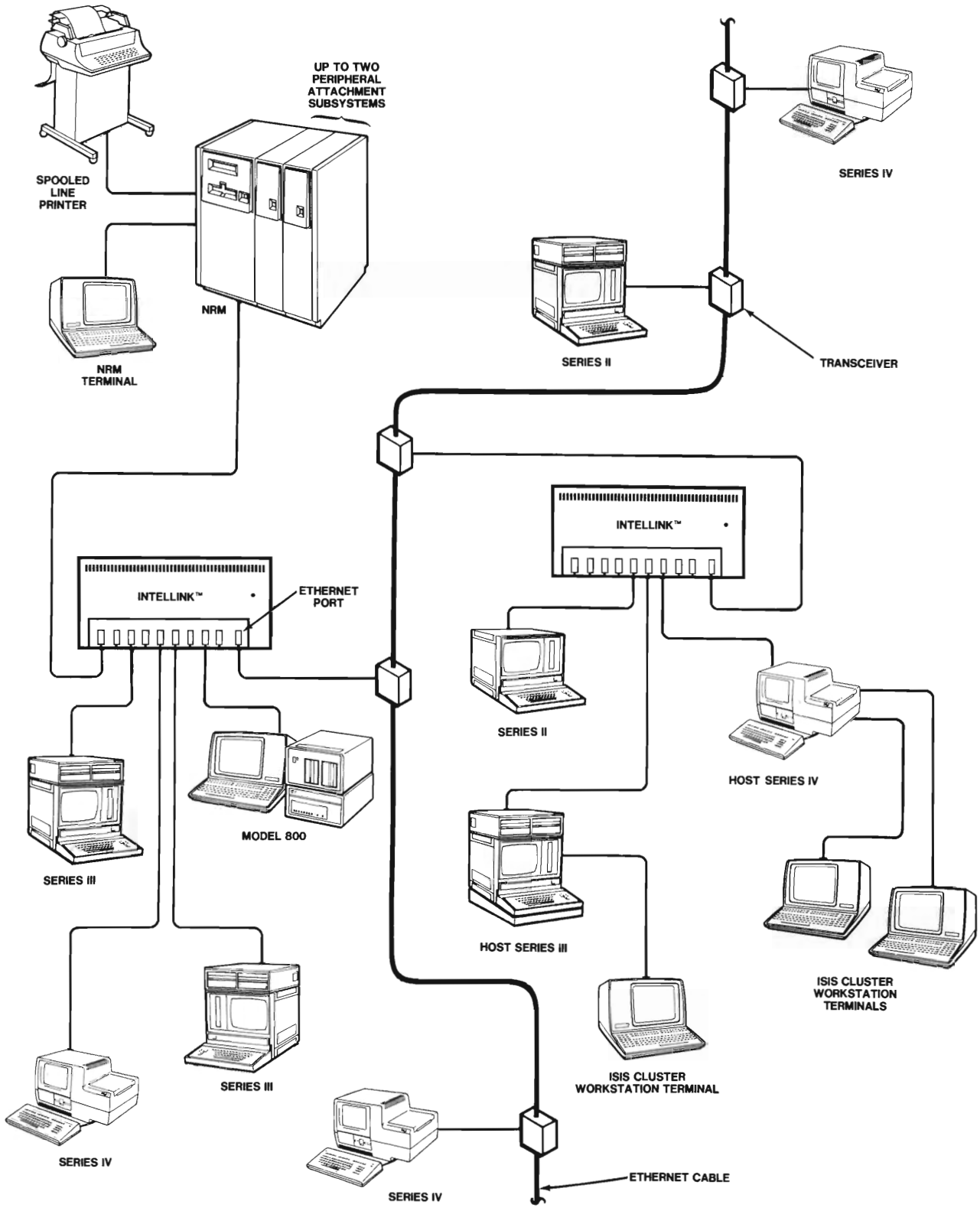


Figure 1-2. NDS-II Network Development System Connected to an Ethernet Cable

122098-2

Data Files

A data file can contain only data.

Directory Files

A directory file is a logical collection of files stored on a disk. A directory file may contain data files or other directory files.

Local Files

Local files, data and directory, reside on the flexible disks located in the flexible disk drives local to a development system workstation. ISIS Cluster workstations can not access local files.

Private Files

A private file is any local or remote file accessible only to the file owner.

Public Files

A public file is any remote file that is accessible to the owner and any other network users.

Remote Files

Remote data and directory files reside in the public network file system at the Network Resource Manager. Only remote files are accessible to an ISIS Cluster workstation.

Shared Files

Shared files reside only in the public file system. All shared files in a directory must have unique names. The pathname to the directory always uniquely defines the directory and any shared file in the system. Any shared file within a directory can be declared accessible to the other network users by setting the appropriate access switches. (See Chapter 4 of this manual.)

Hierarchical File Structure

The NDS-II public file structure is a hierarchical (or inverted tree) file structure. It can contain more than one directory and many data files. Chapters 2 and 3 describe the hierarchical file structure in more detail.

Remote Job Execution

The NDS-II distributed job control (DJC) allows jobs to be submitted from any network workstation and executed remotely at a development system workstation. The NRM uses job queues to control the remote job execution. The NRM recognizes two types of workstations: private and public.

Private Workstation

A private workstation is a workstation that is logged on to the Network and that executes its own jobs exclusively. It can also send jobs to the NRM and have them executed by other workstations. ISIS Cluster workstations are always private.

Public Workstation

A public workstation is a workstation that is logged on to the network and that executes jobs sent from job queues by the Network Resource Manager. Any idle development system workstation can be declared public with the IMPORT command. ISIS Cluster workstations cannot be made public. A public workstation accepts jobs from the NRM only. It cannot send jobs to the NRM. See Chapter 4 of this manual.

Job Queues

Job queues can be created or deleted with the QUEUE command. See Chapter 4.

Network Access

The host operating system of a Series IV development system is iNDX. The iNDX operating system enables the ISIS-III(C) users at Series IV Cluster workstations to access the network.

ISIS-III(N) is the operating system that provides Series III, Series II, and Model 800 workstations with an ISIS environment and network access. The ISIS-III(N) software enables ISIS-III(C) users at Series III, Series II, and Model 800 Cluster workstations to access such network resources as NRM files, a network line printer, and distributed job control.

Device Access

Appendix C lists the network devices that are accessible from a Cluster workstation. Local devices cannot be accessed from a Cluster workstation.

File Access

The ASSIGN command allows you to access remote files using ISIS pathnames. ASSIGN allows you to represent a network directory pathname with a directory identifier. See Chapter 4 for a detailed description of the capabilities and restrictions of the ASSIGN command.

Software Preparation

The *Series IV iMDX 580/582 ISIS Cluster Board Package Installation, Operation, and Service Manual* describes the software installation procedure (e.g., copying the directory file ISIS.SYS to the NRM's system drive) necessary for supporting a Series IV ISIS Cluster workstation on the network. For information on software installation for supporting a Series III, Series II, or Model 800 ISIS Cluster workstation on the network (e.g., copying ISIS-III(N) files from the host to the NRM) refer to the *iMDX 580 ISIS Cluster Board Package Installation, Operation, and Service Manual.*

Logging On

To access network directories or network devices the user must be identified to the Network Resource Manager. The logging on process logically connects any network workstation to the Network Resource Manager and allows access to the network file system.

Before the user can log on at a Cluster workstation, ISIS-III(C) must be initialized. To initialize the Series IV host operating system (iNDX) power up the Series IV, or press the reset button. To initialize the host operating system ISIS-III(N) at a Series III, Series II, or Model 800, power up the development system and disk drives, insert the ISIS-III(N)/III(C) system disk into drive 0, and press reset at the host. The ISIS Cluster configuration and the results of the firmware diagnostics appear at the host (see Figure 1-3).

In Figure 1-3 one of the Cluster workstations is currently disconnected, and the low RAM test failed on the E Cluster workstation. If a diagnostic is not passed, the failure is flagged on the screen of the host (**F**).

CAUTION

If board E fails, the whole Cluster set is impacted. When such a failure is flagged (e.g., **F**), it is recommended that you immediately replace board E. If board E is disconnected, the Cluster set will not function.

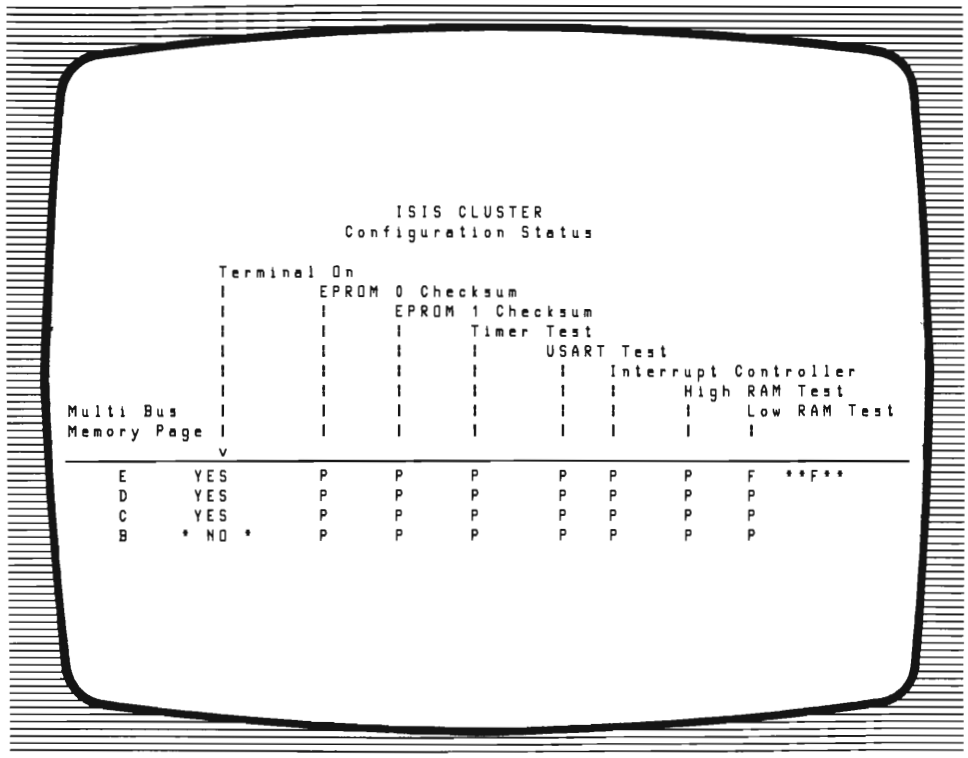


Figure 1-3. ISIS Cluster Configuration Status 122098-3

After the Cluster configuration and diagnostic results appear, the following sign-on prompt appears at the host:

system-id
system prompt

where

system-id	is iNDX's sign-on message at a Series IV host.
	is ISIS-III(N)'s sign-on message at a Series III, Series II, or Model 800 host.
system prompt	is a right angle bracket (>) for the iNDX operating system host.
	is a hyphen (-) for the ISIS-III(N) operating system host.

At the Cluster workstation, the following ISIS-III(C) sign-on message and LOGON prompt appear:

```
ISIS CLUSTER BOOT Vx.y
Copyright year(s) Intel Corporation

ISIS III(C) Vx.y
PLEASE LOGON
USERNAME-
```

If ISIS-III(C) is not successfully initialized after pressing the RESET button at the host, an ISIS Cluster error message appears on the screen of the Cluster workstation. For example,

```
LOW RAM TEST (0-32K) --FAILED
```

Entering Commands

You can enter a command at the host whenever the command level prompt is displayed. Refer to the *NDS-II ISIS-III(N) User's Guide* (121765) for details on ISIS-III(N) commands. For more information on iNDX commands refer to the *Intellec® Series IV Operating and Programming Guide* (121753).

After a successful bootstrap at the host of the ISIS-III(C) operating system, the ISIS-III(C) sign-on message and LOGON prompt appear. You can enter a command at the Cluster workstation whenever the ISIS-III(C) prompt is displayed and you have logged on.

Each ISIS-III(C) command is entered as a command line and must be terminated by a carriage return or a line feed. Pressing the RETURN key after entering a command line automatically enters a line feed. See Chapter 4 of this manual for more details on the ISIS-III(C) commands.

Logging Off

The LOGOFF command logically disconnects any workstation from the Network Resource Manager. See Chapter 4 for the LOGOFF command.

Superuser

Superuser is a username assigned to the person designated as the network administrator. Certain commands can only be entered by the Superuser at the NRM terminal.

Username and Password

The username/password combination enables a user to access the NDS-II network resources. Usernames and passwords can be added or deleted by the Superuser at the Network Resource Manager. See the *NDS-II Network Resource Manager User's Guide* (121883) for more information.

CHAPTER 2
NDS-II DIRECTORY STRUCTURE

This chapter describes the NDS-II distributed file system and the differences between it and the ISIS file structure. The NDS-II distributed file system offers a hierarchical file structure that provides:

- Multiple user access to shared network data and directory files
- Owner controlled access (World and Owner) to the network files
- A list of files that reside in the network directories
- The ability to create new directory and data files while other users are accessing the network files
- Flexibility in file maintenance
- An archiving facility for files stored on the shared disk
- Time and date stamping of network files at the NRM
- Optional cartridge tape backup of network files at the NRM

In previous versions of ISIS-II, files and collections of files were tied to the media (disk) and the disk drive (physical device) where the files were stored. Because they were tied, the terms disk, directory, directory identifier, logical device name, physical device name, and disk drive were functionally identical and could be used interchangeably.

The additional functionality of the NDS-II network requires redefinition of those terms. In this manual, the terms are used as follows:

- Disk—the media where directories of files can be stored
- Directory—a logical collection of files stored on a disk
- Directory Name—a user-specified label for a directory (SYSTEM.DSK)
- Directory Identifier—a label (:F*n*:) that can be used to represent a network directory pathname (/WINCH1.VOL/PROJA.DIR) or another directory identifier (:F3:)
- Disk Drive—a machine used to access a directory stored on a disk
- Physical Device Name—a label assigned to a physical device (line printer, drive 1, console). ISIS Cluster users cannot access physical devices directly.
- Logical Device Name—a label (:BB:, :CO:) assigned to a logical device (byte bucket, console output)

NOTE

A username, password, or remote directory name that has special characters can be used in ISIS-III(C) if the special characters are enclosed within apostrophes. The apostrophes literalize characters that are not usable normally in the ISIS environment.

A directory identifier may be assigned to a network directory pathname or another directory identifier. The default values for the directory identifier name assignments are listed under the ASSIGN command in Chapter 4. These directory identifiers can be altered with the ASSIGN command.

ISIS-II File Structure

The ISIS-II directory is based on a flat directory structure with one directory per disk (see Figure 2-1).

To access FILEB.EXT, you must enter :F1:FILEB.EXT. Every directory has one identifier (:F*n*:) and every file within the directory is a data file. Each disk or volume can contain only one directory.

NDS-II File Structure

The NDS-II file structure is a hierarchical (or inverted tree) file structure. It can contain more than one directory file and many data files (see Figure 2-2).

The Superuser names the volumes and assigns access rights to these volumes. (See the *NDS-II Network Resource Manager User's Guide.*)

Types of Files

Each volume can contain as many files (directory or data) as available storage will allow. A directory file may contain other directory files or data files. Data files contain only data. A data file cannot contain a directory file.

User File Protection

All remote network files in the hierarchical file structure have Owner, World, and Superuser Access Rights that are controlled by access switches. These switches can be controlled (turned on and off) with the ACCESS command described in Chapter 4 of this manual.

All files of an ISIS Cluster workstation are remote.

Pathnames

The files (data and directory) can be traced down through the file structure by a pathname. The pathname identifies every volume and directory from the logical system root to the data file. The pathname for the file pointed to by the arrow in Figure 2-2 is /WINCH1.VOL/DIRB/FILE3.EXT.

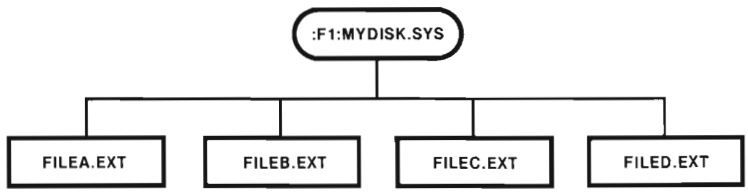


Figure 2-1. ISIS Directory Structure 121765-2

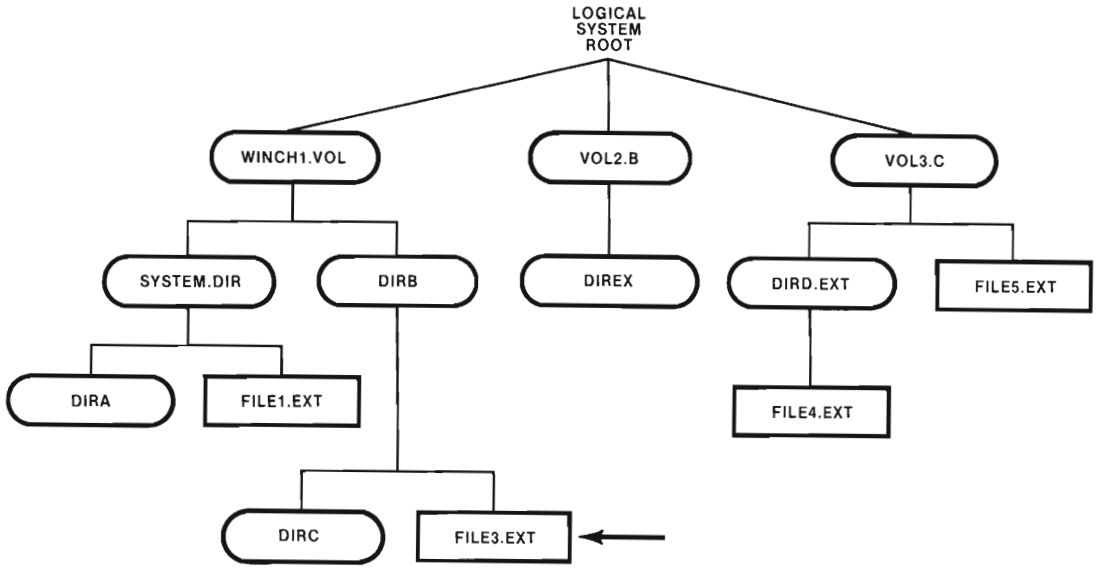


Figure 2-2. Hierarchical File Structure 121765-3

The pathname /WINCH1.VOL/DIRB/FILE3.EXT is a fully qualified pathname because the slash (/) acts as a delimiter between the names of the volume and the directories in the path along the "branches" of the "tree."

Accessing a File

A workstation using ISIS-III(C) cannot directly access tree structured directory files. The ASSIGN command assigns directory identifiers to the pathnames. (See the ASSIGN command description in Chapter 4 of this manual.)

To access the FILE3.EXT in Figure 2-2, you must assign a directory identifier (:F*n*:) to the fully qualified pathname of the directory with the ASSIGN command.

```
-ASSIGN 1 TO /WINCH1.VOL/DIRB‹cr›
```

The pathname for this file is now :F1:FILE3.EXT.

Creating Directories

To create a new directory in the hierarchical tree structure, use the CREATE command described in Chapter 4 of this manual. The CREATE command allows users to add new directory files to existing directory files by specifying either the fully qualified pathname (naming all of the branches of the tree) or the directory identifier assigned to the existing directory file.

CHAPTER 3
GUIDELINES FOR CREATING AND MAINTAINING NDS-II FILES

This chapter offers guidelines that will help you take advantage of the versatility and flexibility of the NDS-II distributed file system.

Directory Maintenance Guidelines

Guidelines you can use to create and maintain your directory and data files:

1. Minimize the number of directories in each volume by subdividing the directories by project and function.
2. Keep all "system files" in one directory file.
3. Create a separate directory of directory files for individual user "miscellaneous files."

The model in Figure 3-1 illustrates a typical development environment.

Logical System Root

The top of the diagram in Figure 3-1 represents the logical system root of the NDS-II distributed file system. This root is where all volumes are connected together. You cannot add data or directory files to the logical system root.

The root stores directory information for all volumes in the NDS-II system.

Volumes

Volumes can be configured during the System Generation process (see the *NDS-II Network Resource Manager User's Guide*). The volumes can be changed dynamically whenever the system is in operation. A volume can contain directory or data files. In this model, one of the volumes, WINCH1.VOL, corresponds to the Winchester disk at the Network Resource Manager.

Directory Files

Directory files reside within each volume. Directory files can contain other directory files or data files. Figure 3-1 shows a volume (WINCH1.VOL) with six directory files, ISIS.SYS, PROJB.DIR, PROJA.DIR, JOHN.DIR, SHEILA.DIR, and LEE.DIR.

The ISIS.SYS directory file contains all the files that a programmer would normally look for in a default directory:

- System files (basic system files ISISC.BIN, ISIS.CLI, ISIS.OV0, ISIS.OV1, ISIS.OV2)
- Command programs (COPY, RENAME, DIR, etc.)
- Text editors (CREDIT, AEDIT)
- Compilers & Translators (PASC86.86, PLM86.86, ASM86.86, FORT86.86, etc.)
- Linkers and Locators (LINK86, LOC86)
- Other tools
- Latest stable version of your prototype software

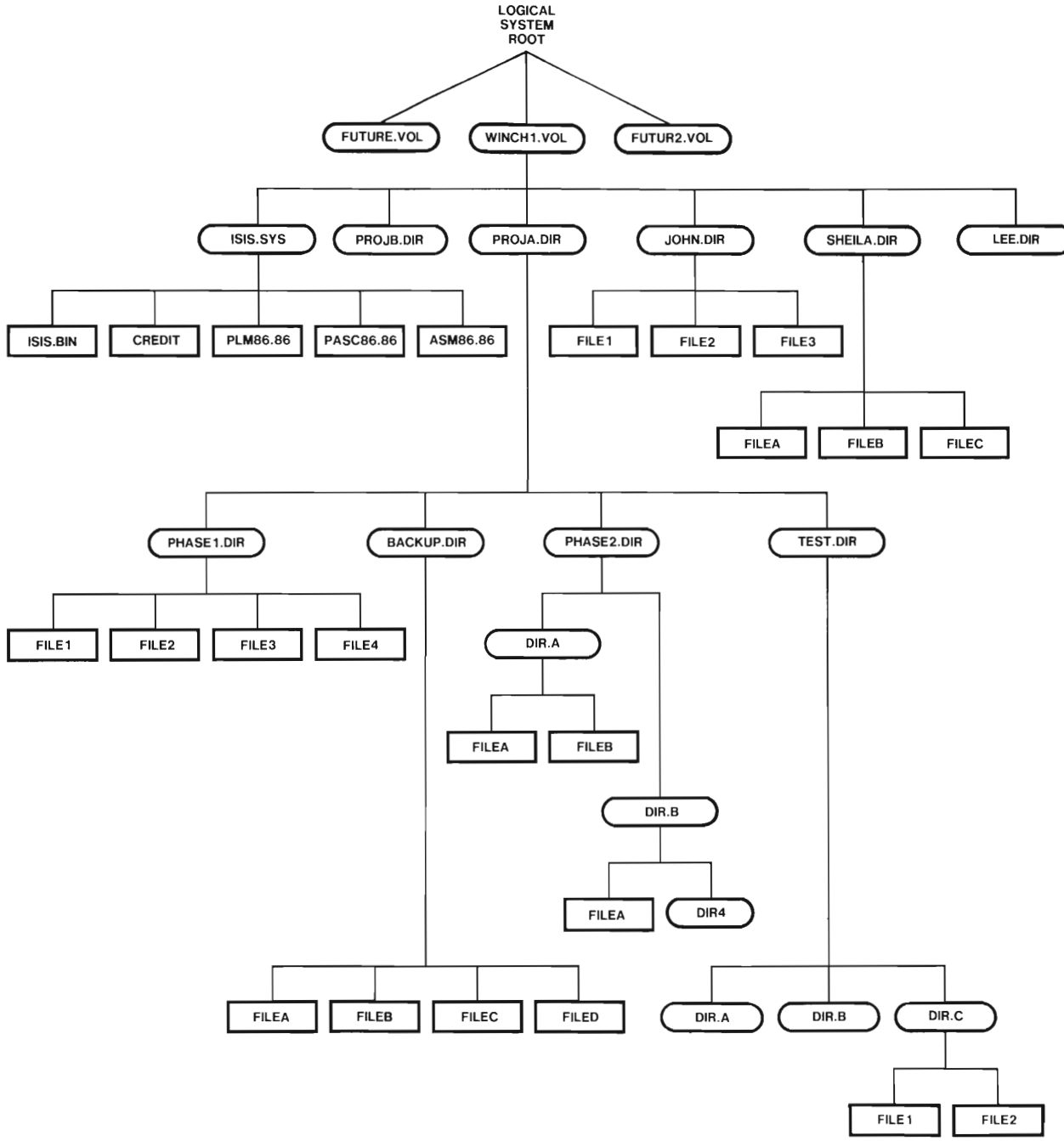


Figure 3-1. Model of NDS-II File System 122098-4

The Superuser owns the ISIS.SYS directory file. The World access rights are READ only to prevent accidental modification or deletion by system users.

Each project directory, PROJA.DIR and PROJB.DIR, contains directories that subdivide the project into phases or sections, depending on the application. Figure 3-1 shows four directories:

- PHASE1.DIR—contains various stages of prototype software and all .SRC, .LST, .OBJ and other work files related to this phase of the project.
- PHASE2.DIR—similar to PHASE1.DIR, except that this directory is further subdivided for programmer convenience.
- BACKUP.DIR—stores backup copies of all relevant files related to the project.
- TEST.DIR—contains separate directories of tests for various aspects of the project. These directories each contain individual test files.

The directory files listed can be controlled, as needed, by individual programmers.

The last directory files in this example, JOHN.DIR, SHEILA.DIR and LEE.DIR, are assigned to individuals:

- name.DIR—each person on the Network has a directory that can be further subdivided into directory files or that can be used to store status reports, memos, trip reports and other miscellaneous work that is not directly related to the project.

Home Directory

A unique home directory can be assigned to each user when the Superuser defines the user. See the *NDS-II Network Resource Manager User's Guide* for more details on assigning home directories.

For convenience, a SUBMIT file called ISIS.INI can be created in the home directory of the user. Commands in ISIS.INI will be executed automatically at LOGON time. Consequently, directory identifier assignments to commonly used pathnames can be set up automatically without user involvement at LOGON. A maximum of ten devices (:F0: to :F9:) can be assigned.

Using the Network

1. To access the files on the Network you must
   - Use a valid username/password combination to log on
   - Know the name of the volume or directory you want to access
   - Have access rights to the directory and data files you will be working with

   The Superuser of the system is the source of this information. (See the *NDS-II Network Resource Manager User's Guide.*)
2. With the information listed in number one, you can log on the Network using the LOGON command (see Chapter 4 of this manual).

3. To access the network directory files, you must assign directory identifiers to the directory names. Using Figure 3-1, assume you will be working in the PROJA.DIR and the PHASE1.DIR.

   a. The default device :F0: is automatically assigned to the directory file ISIS.SYS at LOGON time. This assignment allows you to drop the directory prefix for commands, compilers, etc., that reside in :F0:.

   b. Assign the directory identifier :F1: to the project directory file PROJA.DIR by typing the following:

      ```
      -ASSIGN :F1: TO /WINCH1.VOL/PROJA.DIR<cr>
      ```

   c. Assign the directory identifier :F2: to the directory PHASE1.DIR by typing the following:

      ```
      -ASSIGN :F2: TO /WINCH1.VOL/PROJA.DIR/PHASE1.DIR<cr>
      ```

      or

      ```
      -ASSIGN :F2: TO :F1:PHASE1.DIR<cr>
      ```

4. Use the ACCESS command (see Chapter 4 of this manual) to control World access rights to your directory files and data files.

5. Create new directory files with the CREATE command (see Chapter 4 of this manual).

NOTE

Storage capacity is the only restriction on the number of directory or data files the hierarchical structure will support. Any number of directory files can be "branched" horizontally and vertically.

CHAPTER 4
CLUSTER WORKSTATION COMMANDS

This chapter describes:

- The differences between the ISIS-III(C) and the ISIS-III(N) operating systems
- The differences between the ISIS-III(C) and the ISIS-II operating systems
- New ISIS-III(C) commands associated with the NDS-II network (ACCESS, ASSIGN, CANCEL, CREATE, EXPORT, LOGON, LOGOFF, QUEUE, REMOVE, SPACE, SYSTAT, VERS, and WHO).
- The changes to the existing standalone ISIS commands (ATTRIB, COPY, DELETE, DIR, RENAME, and SUBMIT).

The commands of the NDS-II Cluster workstation operate under the ISIS-III(C) operating system. ISIS-III(C) logically connects the Cluster board to the Network Resource Manager. After a successful bootstrap at the host, ISIS-III(C) allows the Cluster workstation to operate as an independent development station with access to network resources such as shared disk storage and the shared line printer. See Chapter 1 for more details on the bootstrap procedure for a Cluster workstation.

The ISIS-III(C) software uses the commands and error messages described in the *Intellec® Series III Microcomputer Development System Console Operating Instructions* (121608) or in the *ISIS-II User's Guide* (9800306). (ISIS-III(C) error codes are listed in Appendix A of this manual.)

Differences Between ISIS-III(C) and ISIS-III(N) Operating Systems

All ISIS-III(C) operating system files are remote.

Differences Between ISIS-III(C) and ISIS-II Operating Systems

The ISIS-II and ISIS-III(C) operating systems differ in several ways. In this chapter, shaded text highlights the important differences between the ISIS-III(C) software and previous versions of ISIS-II.

Directory Structure

The ISIS-II directory is based on a flat directory structure with one directory per disk, as described in Chapter 2 of this manual.

To access the file FILEB.EXT using ISIS-III(C) you must type :F*n*:FILEB.EXT. Every directory has one name (:F*n*:) and every file within the directory is a data file.

The NDS-II file structure is a hierarchical (or inverted tree) file structure. It can contain more than one directory file and many data files. See Chapters 2 and 3 for more information on this file structure.

Hard Disk

The ISIS-III(C) software can not access local physical devices: a hard disk, a high speed paper tape reader or punch, a local line printer, or flexible disk drives attached to an ISIS or an iNDX workstation. However, each ISIS Cluster workstation can access the hard disk(s) attached to the Network Resource Manager (NRM). The FIXMAP, FORMAT, HDCOPY, and IDISK commands do not operate on the NDS-II NRM hard disk(s).

Remote Printer

The Network Line Printer attached to the NRM prints files as a background job. Users can spool jobs to this remote printer with the COPY command. The DELETE command removes jobs from the spooler queue (:SP:). See Appendix C for a list of devices that ISIS-III(C) supports.

Accessing the NDS-II Network

The ISIS-III(C) software supports two commands (LOGON AND LOGOFF) that allow users to access the NDS-II network. These commands are described later in this chapter.

Command Program Compatibility

The VERS command identifies and displays the version number of the different ISIS command programs. Each command program works correctly only with the corresponding version of ISIS.

Remote Job Execution

The NDS-II Network allows jobs to be exported on the Network and executed at public workstations. The EXPORT command sends a job to a queue at the Network Resource Manager to be executed at a remote (public) workstation. Use the IMPORT command at a development system workstation to create public workstations. (Cluster workstations are always private.) Use the QUEUE command to add or delete queuenames.

The job queues can be monitored with the SYSTAT command and altered with the CANCEL command. These commands are described in detail later in this chapter.

Command Categories

Five command categories are described in this chapter.

- Network Access Commands
- Directory Maintenance Commands
- File Maintenance Commands
- 8080/8085 Program Execution Commands
- Remote Execution Commands

NOTE

The command descriptions appear in alphabetical order.

Network Access Commands	
LOGOFF	Logically disconnects a workstation from the Network
LOGON	Logically connects a workstation to the Network and executes the initialization file ISIS.INI

Directory Maintenance Commands	
ASSIGN	Assigns a directory identifier to a network directory
CREATE	Creates a new directory in the remote network file system
REMOVE	Removes an empty directory in the remote network file system

File Maintenance Commands

ACCESS	Changes or displays the Owner/World access rights of a file in the network distributed file system
ATTRIB	Changes and/or displays the write-protect attribute(s) of a file
COPY	Copies a file from one directory to another
DELETE	Removes references to a file from the directory and frees disk storage space associated with that file
DIR	Displays the names of, and information about, the data and directory files listed within a directory
RENAME	Changes the name of a file
SPACE	Displays the volume information of a root directory file
VERS	Displays the version number of a command program
WHO	Displays the name of the user who is logged on

Wild-Card Filenames

The DIR, COPY, DELETE, ACCESS, and ATTRIB commands allow filenames that conform to the ISIS conventions to be specified using a wild-card construct. Either of two special wild-card characters can replace some or all of the characters in a name or extension. The wild-card characters match anything when the system searches a directory for a filename.

The two wild-card characters are

- An asterisk (*) to specify a wild-card match to any number of characters
- A question mark (?) to specify a wild-card match to a single character

Asterisks specify a wild-card match to any name and/or any extension in the directory. For example,

- ABC.*—means match any filename with the name ABC and any or no extension.
- *.PLM—means match any filename with the extension .PLM, such as A.PLM or MYPROG.PLM.
- *.*—means match all filenames in the directory.

The asterisk can also specify a wild-card match for the remainder of the name or extension except for the initial character. For example,

- AB*.HEX—means match any filename with AB as the first two characters of the name and HEX as the extension. This example would match ABC.HEX, ABXYZ.HEX, AB.HEX.
- A*B.HEX is illegal since no character except a period may follow an asterisk.

NOTE

The wild-card search matches only valid ISIS filenames. A directory name in the remote shared network file system that does not conform to these criteria will not be found in a wild-card search.

8080/8085 Program Execution Commands

SUBMIT	Enters a file that contains commands to be executed

Remote Job Execution Commands

CANCEL	Cancels a queued job that is executing or waiting to execute
EXPORT	Sends a user-specified job to a queue for remote execution
QUEUE	Creates or deletes remote job queues
SYSTAT	Lists jobs waiting in a remote job queue

The NDS-II Network Resource Manager controls the remote job execution on the network. It recognizes public and private workstations and maintains all status information about remote jobs and the workstations.

NDS-II provides distributed job control that allows user-created jobs to be executed at public workstations. Commands are also available to monitor and to cancel jobs queued for execution.

Public and Private Workstations

The NDS-II NRM recognizes two types of workstations: private workstations, and public workstations. When a workstation is first powered up, it is a private workstation. This workstation operates as a normal network development system. In addition, it can send jobs to the NRM for execution by public workstations in the network. ISIS Cluster workstations are always private.

A public workstation accepts jobs from the NRM for remote execution. A private development system workstation can be turned into a public workstation by executing the IMPORT command. To restore a public workstation to a private workstation, press Control-C at the keyboard.

Creating and Naming Queues

When a private workstation is turned into a public workstation with the IMPORT command, you must specify which queues the workstation will service. If the specified queue does exist, the public workstation is added to the list of servers for the specified queue. If the specified queue does not exist, an error message is displayed.

The network job queues provide a way to match the type of job with the type of workstation necessary to execute the job. The NRM does not have predefined job queues. You can create job queues with the QUEUE command. This flexibility allows a customized match of queue names and the workstations serving the queues.

Because the NDS-II network may contain different types of development system workstations (Series IV, Series III, Series II, or Model 800's), it may be necessary to determine which workstation should get what types of jobs. A properly designed queue naming convention is necessary to help insure that only a workstation with the correct hardware will execute a job. For example, use SERIESII as the queue name for 8085 programs and SERIESIII as the queue name for 8086 programs.

The NRM sends jobs from the queues to public workstations as the workstations become available. It is not possible to indicate exactly which workstation, of those serving a particular queue, will execute a remote job. For example, if an 8086 program is mistakenly sent to a job queue served by a Series III workstation and a Model 800 workstation, the job may be sent to the Model 800 workstation to execute. A Model 800 workstation does not have an 8086 processor to execute the job—therefore, a fatal error will occur and the job will be aborted.

Program Control and Code Conversion Commands

The following manuals describe the program control commands (for Librarian, Linker, and Locator) and code conversion commands (for hexadecimal to/from object module format conversion):

- *MCS®-80/85 Utilities User's Guide for 8080/8085-Based Development Systems* (121617)
- *iAPX 86,88 Family Utilities User's Guide for 8086-Based Development Systems* (121616)

Entering Commands

ISIS-III(C) enables users at ISIS Cluster workstations to enter commands to the NDS-II network. However, the host's operating system must be initialized before ISIS-III(C) can be invoked.

To initialize the Series IV host operating system (iNDX), power up the Series IV or press the reset button. To initialize the host operating system ISIS-III(N) at a Series III, Series II, or Model 800, power up the development system and disk drives, insert the ISIS-III(N)/III(C) system disk into drive 0, and press reset at the host. The ISIS Cluster configuration and the results of the firmware diagnostics appear at the host (see Figure 1-3).

Then the host's sign-on message and logon prompt appear at the host:

system-id
system prompt

where

system-id	is iNDX's sign-on message at a Series IV host.
	is ISIS-III(N)'s sign-on message at a Series III, Series II, or Model 800 host.
system prompt	is a right angle bracket (›) for the iNDX operating system host.
	is a hyphen (-) for the ISIS-III(N) operating system host.

You can enter a command to the host development system whenever the command level prompt is displayed. Each command is entered as a command line and must be terminated by a carriage return or a line feed. Pressing the RETURN key after entering a command line automatically enters a line feed.

After the host operating system signs on, the following ISIS-III(C) sign-on message and LOGON prompt appear at the Cluster workstation:

```
ISIS CLUSTER BOOT Vx.y
Copyright year(s) Intel Corporation
ISIS-III(C) Vx.y
PLEASE LOGON
USERNAME-
```

You can enter a command at the Cluster workstation whenever the ISIS-III(C) prompt (-) appears.

Command Syntax

Following is the general syntax of an ISIS-III(C) console command:

command parameters ‹ c r ›

where

command	is the name of a program.
parameters	are one or more items required by the command. When entering more than one parameter, separate them with commas or blank spaces unless otherwise noted under the individual commands. When a parameter consists of switches, separate them by spaces, not by commas.

In most cases, a command executes when the carriage return is encountered. Any exceptions are noted under the individual commands.

NOTE

Some of the ISIS-III(C) commands have the same name and a similar function as the iNDX operating system commands that run on the Series IV and the NDS-II Network Resource Manager. However, the syntax of the commands may be different. You must use the correct command syntax for the corresponding operating system.

Specifying Files

The command syntax of many commands includes the following designation:

: F *n* : *filename*

Where this format is shown, the following definitions apply unless otherwise noted under the individual command:

:F*n*: is the directory identifier of the directory or device that contains *filename*. The value *n* is an integer between 0 and 9 inclusive. If :F*n*: is not specified, :F0: is assumed.

filename is the name (and extension, if any) of the target file. Enter *filename* immediately after :F*n*: with no intervening space, as in :F1:MYPROG.

ACCESS

Syntax

```
ACCESS { :Fn: } filename [ switch ] <cr>
       { pathname/ }
```

where

:F*n*:	is the directory identifier that contains *filename*. The value *n* is an integer between 0 and 9 inclusive. If :F*n*: is not specified, :F0: is assumed.
pathname	is a fully qualified pathname to the *filename*'s directory.
filename	is the name and extension, if any, of the shared file.
switch	for the OWNER ACCESS RIGHT consists of three parts: the OWNER Identifier, the ACCESS Identifier and the RIGHT Identifier as listed in Table 4-1.

Description

The ACCESS command lists or changes the Owner or World access rights of a shared data or directory file. These switches can be used to protect files from accidental change or deletion.

Listing Current Access Rights. To list the current access rights of a shared file, type the following:

```
-ACCESS :Fn:filename<cr>
FILE NAME   OWNER  LENGTH  TYPE  OWNER ACCESS  WORLD ACCESS
WORKFL.NEW  MARK     3128  DATA      D R W           R
```

Changing Access Rights. To change the access rights of a data file or a directory file, use the ACCESS command switches. The 24 different switches consist of three characters typed with no intervening space.

The first character indicates the Owner Rights to be changed—the file owner (Owner) or the other network users (World).

Table 4-1. Network File Access Rights

Identifier	Options
OWNER	O — Owner of the file or directory W — World or public users
ACCESS	Data files: D — Delete a file R — Read a file W— Write a file Directory files: D — Delete a directory L — List a directory A — Add a directory entry
RIGHT	0 — Deny access right (reset, off) 1 — Grant access right (set, on)

The second character indicates the type of access to be altered. A file has three access rights: Read, Write and Delete. A directory also has three access rights: List, Add-entry and Delete.

The third character indicates the access right as reset (off) or set (on). A zero (0) indicates the switch is off and the access right is not allowed. A one (1) indicates the switch is on and the access right is allowed.

After the access rights of a file or a directory are changed, the new access rights are displayed.

To allow the World to read a file, set the World Read switch ON:

```
-ACCESS FILE1.EXT WR1<cr>
```

To allow the public to add to a directory, set the World Add switch ON:

```
-ACCESS MYDIR.ALL WA1<cr>
```

Any file that has the World Write set can be altered by an network user. Any file that has the Delete access right set can be deleted by any network user even if he does not have Write access to the file.

Altering Access Rights of Multiple Files and Directories. The access rights of more than one shared file or directory can be changed simultaneously. However, the access rights of the files must be changed to a common switch. To "turn off" the Owner Delete rights to files FILE1.EXT, FILE2.EXT, and FILE3.EXT, type the following:

```
-ACCESS FILE?.EXT OD0<cr>
```

Superuser Access Rights. Every network file has access rights assigned to the Superuser. These switches can only be manipulated by the Superuser. The default switch setting for the Superuser is Read only.

Possible Error Conditions

An error occurs when

- You are not logged on
- You try to access a file or a directory that does not exist
- You do not have access rights to change the access characteristics of a file or a directory

Notes

1. Use the ACCESS command with remote files only.

Examples

1. This example "turns off" the World Add access rights of a shared file.

```
-ACCESS NEWFIL.EXT WA0<cr>
```

2. This example "turns on" the Owner Read and Write access rights of any file that matches in an ISIS wild-card search for DAT *.*.

```
-ACCESS DAT*.* OR1 OW1<cr>
```

3. This example lists the current access rights of a remote file.

```
-ACCESS ACC.DIR<cr>
FILE NAME OWNER  LENGTH TYPE OWNER ACCESS WORLD ACCESS
ACC.ALL   JEANNE 10904  DATA    D    W           W
```

ASSIGN

Syntax

```
ASSIGN [{:Fn:} ] [TO y]
       {  n  }
```

where

:F*n*: or *n* — is the directory identifier. *n* can be any number 0–9. The directory identifier for the default system directory ISIS.SYS is :F0:. The directory :F0: must contain the files ISIS.CLI, ISISC.BIN, ISIS.OV0, ISIS.OV1, and ISIS.OV2.

y — is one of the following:

- A network directory pathname (/WINCH1.VOL/ PROJA.DIR/ TEST.DIR)
- Another directory identifier (:F0:—:F9:)
- A directory identifier and a pathname component (:F0:MYDIR)
- The word NULL

Description

The ASSIGN command allows a directory identifier to be mapped into the NDS-II hierarchical file system. ASSIGN provides access to the shared network files with ISIS file naming conventions. This command lists current assignments and assigns a directory identifier to a network directory pathname or another directory identifier.

Startup. At initial system bootstrap, when the RESET button is pressed, the directory identifiers are set to the default values listed in Table 4-2. These assigned directory identifiers can be changed with the ASSIGN command.

Listing Current Assignments. To list the current assignments of directory identifiers to disk drives, type the following:

```
-ASSIGN<cr>
```

Table 4-2. ISIS-III(C) Default Configuration

Directory Identifier	Directory Name
:F0:	*/root/ISIS.SYS*
:F1:	NULL
:F2:	NULL
:F3:	NULL
:F4:	NULL
:F5:	NULL
:F6:	NULL
:F7:	NULL
:F8:	NULL
:F9:	*user home directory*

This command generates a listing of all directory identifiers and their current assignments.

```
DEVICE          ASSIGNED TO
  :F0:          /WINCH1.VOL/ISIS.SYS
  :F1:          NULL
  :F2:          NULL
  :F3:          NULL
  :F4:          NULL
  :F5:          NULL
  :F6:          NULL
  :F7:          NULL
  :F8:          NULL
  :F9:          /WINCH1.VOL/JOHN.DIR
```

NOTE

:F9: is initially assigned to the user's home directory.

Changing Directory Identifier. To change the directory identifier assignment, use the ASSIGN command and follow the syntax conventions listed above. For example, to change the directory identifier assignment of :F0: to the directory /WINCH1.VOL/PROJA.DIR, after the logon prompt appears type the following:

```
username<cr>
password<cr>
-ASSIGN :F0: TO /WINCH1.VOL/PROJA.DIR<cr>
```

This allows access to the ISIS command programs and other files on the disk without typing a prefix. Any command with :F0: or no directory identifier will be loaded from directory /WINCH1.VOL/PROJA.DIR.

NOTE

Default directory :F0: must contain the ISIS system files and any other command files to be invoked from the default directory :F0: (COPY, DIR, ASSIGN, etc.).

Assigning Network Pathnames and Directories. A directory identifier can be assigned to a network directory pathname. This allows access to remote files maintained in the remote shared network directories.

In the directory tree shown in Figure 4-1, to access the file CHAP2.TXT in the MYBOOK.DIR directory, assign the directory to :F1: by typing the following after the logon prompt appears:

```
username<cr>
password<cr>
-ASSIGN :F1: TO /PROJA.VOL/MYBOOK.DIR<cr>
```

You can now access the file using the pathname :F1:CHAP2.TXT.

Up to ten network directories can be assigned with this Cluster configuration (e.g, :F0:—:F9:).

CAUTION

Basic ISIS system files must be present on the default directory.

Hardware RESET on the host development system is not specific to the host. A RESET at the host resets ALL workstations on the Cluster.

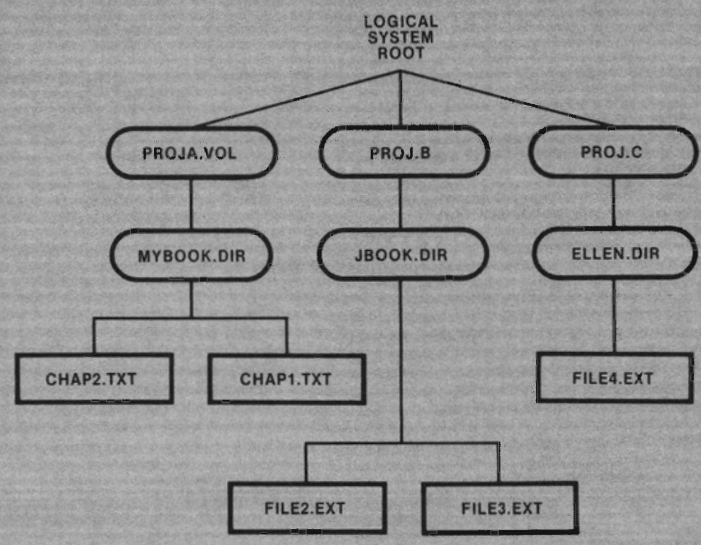


Figure 4-1. Sample Network File Structure 121786-5

Notes

1. Hardware RESET on the host development system and LOGOFF reassign the directory identifiers to the default values (see Table 4-2.)
2. Interrupt 1 will not affect directory identifier assignments.
3. When a directory identifier is assigned to a network directory file, ASSIGN will display the fully qualified pathname of the file.

Examples

1. In the following example, directory identifier :F0: is assigned to the home directory /WINCH1.VOL/JOHN.DIR.

```
-ASSIGN :F0: TO /WINCH1.VOL/JOHN.DIR<cr>
```

or:

```
-ASSIGN :F0: TO :F9:<cr>
```

2. This example lists the assignments for a network in which the directory identifier was originally assigned to /WINCH1.VOL/ISIS.SYS and has been assigned to /WINCH1.VOL/JOHN.DIR.

```
-ASSIGN<cr>
DEVICE        ASSIGNED TO
 :F0:         /WINCH1.VOL/JOHN.DIR
 :F1:         NULL
 :F2:         NULL
 :F3:         NULL
 :F4:         NULL
 :F5:         NULL
 :F6:         NULL
```

```
  :F7:          NULL
  :F8:          NULL
  :F9:          /WINCH1.VOL/JOHN.DIR
```

3. This example assigns the directory identifier :F1: to the network directory pathname /PROJA.VOL/MYBOOK.DIR in Figure 4-1.

```
-ASSIGN :F1: TO /PROJA.VOL/MYBOOK.DIR<cr>
-ASSIGN<cr>
DEVICE          ASSIGNED TO
  :F0:          /WINCH1.VOL/JOHN.DIR
  :F1:          /PROJA.VOL/MYBOOK.DIR
  :F2:          NULL
  :F3:          NULL
  :F4:          NULL
  :F5:          NULL
  :F6:          NULL
  :F7:          NULL
  :F8:          NULL
  :F9:          /WINCH1.VOL/JOHN.DIR
```

4. To assign a directory identifier to a directory within the MYBOOK.DIR directory file, type the following:

```
-ASSIGN :F7: TO /PROJA.VOL/MYBOOK.DIR/CHAP1.DIR<cr>
```

or:

```
-ASSIGN :F7: TO :F1:CHAP1.DIR<cr>
```

or:

```
-ASSIGN 7 TO :F1:CHAP1.DIR<cr>
```

5. This example shows an attempt to assign a directory identifier to a directory with more than 14 characters.

```
-ASSIGN 5 TO /PROJA.VOL/MYBOOK.DIR/CHAPTERTWENTYSEVEN<cr>
CHAPTERTWENTYSEVEN, ILLEGAL FILE NAME
```

ATTRIB

Syntax

```
ATTRIB [:Fn:] filename [attriblist] [Q]<cr>
```

where

:F*n*:	is the directory identifier of the directory where the file resides.
filename	is a file whose write-protect attribute is to be changed. The wild-card construction can be used to change and/or display the write-protect attribute of a group of files.
attriblist	is the following:
	W0 or W1 Resets (W0) or sets (W1) the write-protect attribute of a file. When set, the file cannot be opened for output or update, and cannot be deleted or renamed.
	Q Specifies query mode operation.

If two values of the attribute are specified, for example both W0 and W1, the one rightmost in the command takes precedence.

Description

The ATTRIB command changes and/or displays the write-protect attribute of a file.

When you specify the Q switch, ATTRIB displays the following messages before changing the write-protect attribute of a file.

filename, MODIFY ATTRIBUTES?

Type a Y or y to modify the file write-protect attribute. Any other response causes ATTRIB to leave the write-protect attribute unchanged for the specified file and to go on to the next file in the group. When the write-protect attribute for a file has been changed, the current write-protect attribute for the file is displayed.

If a nonexistent disk file is specified, ATTRIB displays the following:

filename, NO SUCH FILE

If a non-disk file is specified, ATTRIB displays the following:

filename, NON-DISK DEVICE

Possible Error Conditions

An error occurs when

- You are not logged on
- The file does not exist
- The directories are not assigned
- You do not have the proper access rights

Example

1. This example changes the write-protect attribute of a group of files.

```
-ATTRIB PROGA.* W1<cr>
        FILE              CURRENT ATTRIBUTES
 :F0:PROGA.SRC                    W
 :F0:PROGA.OBJ                    W
-
```

CANCEL

Syntax

```
CANCEL queuename { (jobname)    } ,...<cr>
                 { (#jobnumber) }
```

where

queuename	is the location where the job was sent to await execution.
jobname	is the filename of the job sent to the queue.
jobnumber	follows the # sign and is the hexadecimal number assigned by the system when the job is sent to the queue.

Description

Use the CANCEL command to cancel remote jobs. A job may be canceled when it is queued awaiting execution or while executing. Any number of jobs may be canceled at any one time.

Use the SYSTAT command to find the job name and number of a specific job in the queue.

NOTE

The CANCEL command cannot be used from an imported job that is executing. An error message appears.

Possible Error Conditions

An error will occur when

- You are not logged on
- The job name does not exist
- The job number does not exist
- You attempt to cancel a job you did not export
- You attempt to cancel a job that belongs to another user

Examples

1. This example cancels the job COMPLE.CSD in queue SIIJOBS.

```
-CANCEL SIIJOBS (COMPLE.CSD)<cr>
```

2. This example cancels three jobs in two different queues.

```
-CANCEL SIIJOBS (COMPLE.CSD, #27), SIIJOBS (MYPROG.CSD)<cr>
```

COPY

Syntax

```
COPY [:Fn:] infile [,...] TO {[:Fn:][outfile]} [switches] <cr>
                             {:device:       }
```

where

:F*n*:	refers to the directory identifier of the directory where the file resides.
infile	is a file (or group of files when using the wild-card construct) to be copied. The copy does not affect the contents of *infile*. If more than one *infile* is specified, they are concatenated in the order specified. When concatenating files, specify the full name and extension of each file; do not use the wild-card construct. *infile* can also be :CI:.
outfile	is a file to be created or recreated. If :F*n*: is not specified, :F0: is assumed. *outfile* must include the extension, if any. If *outfile* is not specified, :F*n*: must be specified.
:device:	is :SP:, :CO:, or :BB:.
switches	are one or more of the following:

Q Specifies the query mode. The system displays the following message before a copy is performed—COPY *infile* TO *outfile*?. A yes or y response causes the copy to be performed. Any other response causes the copy not to be performed.

B Deletes an existing file without displaying the ALREADY EXISTS prompt. The existing file is deleted and recreated with new data.

U Opens *outfile* for update instead of deleting it. The ALREADY EXISTS message is suppressed. The length is not changed unless the copy causes an increase in the size of the file.

If U and B are both specified, the U function is performed.

Description

The COPY command copies files from one directory to another.

When copying from one directory to another, the copy must be made from an input device to an output device. For example, you can copy from a file to the console output but not from the console output to a file.

If *outfile* is an existing disk file and is not write-protected, the following message is displayed:

```
outfile FILE ALREADY EXISTS
DELETE?
```

A yes or y response (followed by a carriage return) causes COPY to delete the existing file before making the copy. No change is made for any other response.

If *outfile* is write-protected, then the following message is output:

```
outfile WRITE PROTECTED
```

Wild-Card Designations. When you use wild-card designations, the following rules apply:

- Every position in the *infile* name that contains an asterisk must have a corresponding asterisk in the *outfile* name.
- Every position in the *infile* name that contains a question mark must have a corresponding question mark or asterisk in the *outfile* name.
- The wild-card characters cannot be used in directory designations (you cannot specify :F*:).

To copy files selectively with the wild-card construct, use the query mode. For example,

```
-COPY :F0:CHAP?.DFT TO :F1: Q<cr>
```

The system then displays the query message before copying each file.

Copying to Another Directory. The COPY command provides a special case for convenience when copying directory files to a different directory. If *outfile* is to have the same name as *infile*, a specific *outfile* is unnecessary. For example,

```
-COPY :F1:ABC.XYZ TO :F2:<cr>
```

is the same as specifying the following:

```
-COPY :F1:ABC.XYZ TO :F2:ABC.XYZ<cr>
```

This form can be used with wild-card designations in *infile*:

```
-COPY :F1:*.* TO :F2:<cr>
```

At the end of the listing of files that were copied, the following message is displayed if write-protected files have been encountered:

```
WRITE PROTECTED FILE ENCOUNTERED
```

The write-protected files are not copied.

Using a wild-card designation when concatenating files causes an error message to be displayed.

```
-COPY A, BC.* TO D<cr>
WILD CARD DELIMITERS DURING CONCATENATE
```

When you use the concatenate operation, *outfile* must not have the same name as *infile*. If it does, the following error message results:

```
-COPY A, B TO B<cr>
SOURCE FILE EQUALS OUTPUT FILE ERROR
```

If the rules governing wild-card designations are not followed, the following error message is displayed:

```
-COPY ABC. TO D<cr>
FILE MASK ERROR
```

Possible Error Conditions

An error occurs when

- The source file is not present
- The destination file already exists
- The destination file is write-protected
- You are not logged on
- You do not have access rights to the source file
- You do not have access rights to the destination file

Notes

1. Directory names that contain an asterisk (*) can be used. However, apostrophes (') must appear around the entire character string when accessing the file. The quotes literalize characters that are not normally usable in the ISIS environment: COPY '/WINCH1/PROJ*A/FILE1' to :F2:MYFILE < cr > .

Examples

1. This example copies three files to one, overwriting its contents.

```
-COPY CHAP1,CHAP2,CHAP3 TO BOOK<cr>
:F0:BOOK FILE ALREADY EXISTS,
DELETE? Y<cr>
APPENDED :F0:CHAP1 TO :F0:BOOK
APPENDED :F0:CHAP2 TO :F0:BOOK
APPENDED :F0:CHAP3 TO :F0:BOOK
-
```

2. Example 1 could have been done in the following way using the update switch:

```
-COPY CHAP1,CHAP2,CHAP3 TO BOOK U<cr>
APPENDED :F0:CHAP1 TO :F0:BOOK
APPENDED :F0:CHAP2 TO :F0:BOOK
APPENDED :F0:CHAP3 TO :F0:BOOK
-
```

3. This example copies a file from directory 0 to directory 1.

```
-COPY PROA TO :F1:NEWPRG B<cr>
COPIED :F0:PROGA TO :F1:NEWPRG
-
```

4. These examples show valid uses of wild-card names with the COPY command.

```
-COPY :F1:*.* TO :F2:<cr>
-COPY :F1:A??C TO :F0:D??E<cr>
-COPY :F1:A????? TO :F0:B*.CPY<cr>
```

5. This example shows a valid use of the COPY command concatenating files on a remote directory.

```
-COPY :F6:REMTE.1,:F6:REMTE.2 TO :F6:REMTE.3<cr>
APPENDED :F6:REMTE.1 TO :F6:REMTE.3
APPENDED :F6:REMTE.2 TO :F6:REMTE.3
-
```

6. This example prints the file :F1:BOOK at the network spooled printer queue.

```
-COPY :F1:BOOK TO :SP:<cr>
```

7. This example copies a file, :F3:MYFILE.TXT, to the network spooled printer queue.

```
-COPY :F3:MYFILE.TXT TO :SP:<cr>
:F3:MYFILE.TXT COPIED TO :SP:
-
```

CREATE

Syntax

CREATE { :Fn: / pathname/ } *new directory name* ‹ c r ›

where

pathname	is an existing directory.
:F*n*:	is the directory identifier assigned to a directory.
new directory name	is a string of up to 14 alphanumeric characters.

Description

The CREATE command creates a new directory in the remote network file system. When specifying a directory identifier, the user may specify a fully qualified pathname or a pathname component prefixed by a directory identifier.

All of the components in a pathname must point to existing directories, except the new directory name. The new directory name is the only component that specifies a nonexistent directory.

When a directory pathname is prefixed by a directory identifier (:F*n*:), the name must be assigned to a valid directory path.

The user is automatically granted all owner access rights when this command creates a new directory. The access rights are

- Delete access
- List access
- Add-entry access

These access rights can be changed with the ACCESS command.

Possible Error Conditions

An error will occur when

- You are not logged on
- You do not have access rights to create a directory
- The parent directory does not exist
- The new directory name already exists
- The new directory name exceeds 14 characters

Notes

1. Use the REMOVE command to eliminate an empty directory created with the CREATE command.

Examples

1. This example creates a directory in the /VOL/NRM1/COMPILERS directory.

```
-CREATE /VOL/NRM1/COMPILERS/SRCS<cr>
-
```

2. This example creates a directory identical to the directory in example 1.

```
-ASSIGN :F5:TO /VOL/NRM1/COMPILERS<cr>
-CREATE :F5:SRCS<cr>
-
```

DELETE

Syntax

```
DELETE [:Fn:]filename [switch] <cr>
```

where

:F*n*:	refers to the directory identifier of the directory where the file resides.
filename	is the name of a file or directory to be deleted. The wild-card construction can be used to delete a group of private files.
switch	is Q, specifying the query mode. The system displays the following message before each file is deleted—*filename*, DELETE. A yes or y response causes the deletion. Any other response causes the deletion not to be performed.

Description

The DELETE command deletes specified private directory entries and shared directory entries if the World access delete switch is set.

This command effectively removes the specified file, directory, or group of files from a directory, making the space it occupied available for reassignment.

If *filename* specified in the DELETE command does not exist, the following message is sent to the console:

filename, NO SUCH FILE

If the file cannot be deleted because it has the write-protect attribute set, the following message is sent to the console:

filename, WRITE PROTECTED

NOTE

Before a directory can be deleted, all of the directory entries must be deleted.

Query Mode. Using the Q switch displays the query message before deleting each file.

The query mode allows you to delete files selectively when using the wild-card construct. For example,

```
-DELETE :Fn:CHAP?.* Q<cr>
```

The system then displays the query message for each file that matches the wild-card construct.

Examples

1. This example deletes three files.

```
-DELETE CHAP?.*<cr>
:F0:CHAP1.TXT, DELETED
:F0:CHAP2.LST, DELETED
:F0:CHAP3.SRC, DELETED
-
```

2. This example shows an attempt to delete a file with no delete access rights.

```
-DELETE PROGA.ASM<cr>
:F0:PROGA.ASM, WRITE PROTECTED
-
```

3. This example deletes a file from the spooled print queue.

```
-DELETE :SP:BOOK<cr>
:SP:BOOK, DELETED
-
```

4. This example deletes a file using the query switch.

```
-DELETE :F2:PROJ.ASM Q<cr>
 :F2:PROJ.ASM, DELETE?
-Y<cr>
 :F2:PROJ.ASM, DELETED
-
```

DIR

Syntax

```
DIR [ { FOR filename } ] [ TO listfile ] [ switches ] ‹ c r ›
      { /            }
```

The positions of these fields are not fixed.

where

filename is the file (or group of files specified with the wild-card construction) whose directory entry is to be listed. If FOR *filename* is omitted, the entire directory is listed.

/ will print all of the remote volume root directories that are available.

listfile is the name of a file or output device such as :F2:LIST.FIL, where the directory listing will be displayed. If TO *listfile* is omitted, the listing is displayed on the screen.

switches are one or more of the following, separated by spaces:

0—9 Lists the directory. If omitted, the directory :F0: is listed. If more than one directory number is specified, only the rightmost one has effect. The directory number also overrides any directory specification in FOR *filename*.

E Lists the extended directory for remote network directories including filename, number of bytes in the file, the owner's name, file type (directory or data), owner access rights, and world access rights.

SP Lists the contents of the network spooler print queue.

O Prints the file directory in a single column format. The default is a triple column format for remote network directories.

Description

The DIR command lists the contents of a specific directory.

NOTE

The wild-card search matches only valid ISIS filenames. A directory name in the remote shared network file system that does not conform to these criteria will not be found in a wild-card search.

Network Directory

```
:Fn: REMOTE DIRECTORY
FILENAME OWNER LENGTH TYPE OWNER ACCESS WORLD ACCESS
```

where

FILENAME lists the name of each data and directory file that resides in this directory.

OWNER	is *username*—the username of the file owner SYSTEM—the operating system and command files SUPERUSER—files that belong to the Superuser NOT FOUND—files assigned to a user whose username has been deleted from the system
LENGTH	is the number of bytes the file takes up in the disk.
TYPE	identifies the file as a data or directory file.
OWNER ACCESS	lists the access rights of the file owner.
WORLD ACCESS	lists the access rights for other network users set by the file owner.

Examples

1. This example lists the extended directory for a network directory.

```
-DIR 0 E<cr>
:F0: REMOTE DIRECTORY
FILE NAME      OWNER       LENGTH   TYPE    OWNER ACCESS   WORLD ACCESS
MEMOS.DIR      SANDI        10904   DIR       D L A          D L A
MYPROG.SRC     RITA          7299   DATA      D R              R
WORKFL.NEW     JEANNE        8063   DATA        R W            R
SYSTEM.LIB     SUPERUSER     3128   DATA      D R W            R
A              USER1         7299   DIR       D L A          D L A
ACC.DIR        JEANNE       10904   DATA      D   W          W
PROG.X12       DENNIS       10054   DATA      D R W            R
FMHP.CSD       AMY            210   DATA      D R              R
AEDIT          SUPERUSER    27467   DATA        R            D R W
```

2. This example displays a directory of the spooler printer queue.

```
-DIR SP<cr>
REMOTE SPOOLER
FILE NAME      OWNER     LENGTH   TYPE   OWNER ACCESS   WORLD ACCESS
9E.DIR         A         00123    DATA     D R W
```

3. This example copies the spooler directory to a file :F1:PRINT.LST, and prints it at the spooler printer queue.

```
-DIR SP TO :F1:PRINT.LST<cr>
-COPY :F1:PRINT.LST TO :SP:<cr>
```

4. This example will list all of the remote volume root directories that are available.

```
-DIR/<cr>

 VOLUME NAME   LOCATION   ACCESSIBLE
  INDX.531      REMOTE        YES
  ISIS.SYS      REMOTE        YES
```

EXPORT

Syntax

EXPORT { :Fn: / pathname/ } *filename* ['('parameter list')'] TO *queuename* [LOG / NOLOG] <cr>

where

:F*n*:	is the directory identifier assigned to the directory that contains *filename*.
pathname	is a fully qualified pathname to the directory identified by :F*n*:.
filename	is the name and extension, if any, of the shared file.
'('*parameter list*')'	is an actual value that replaces a formal parameter in the command sequence definition file. The maximum number of parameters allowed is 10. When omitting a parameter from the EXPORT file, enter a comma in its place.
queuename	is a string of up to 14 printable characters. Any character may be used except a slash (/) or a blank. Queue name identifies where the job is sent for execution.
LOG	indicates a log file (*filename*.LOG) will be created when the file executes completely. The log file records the ISIS console output during remote job execution. LOG is the default control.
NOLOG	indicates no log file is created.

Description

The EXPORT command sends user-specified jobs to queues for remote execution. EXPORT does not modify an existing source file. A temporary file with extension .CS is created and the source file is written to this file along with parameter substitution character strings. The *filename*.CS file is queued for remote execution.

Preparing a Job. The Command Sequence Definition file (.CSD file) is an input file specified in the EXPORT command that contains the sequence of commands to be executed. The file may contain formal parameters (%0—%9). When no extension is supplied, EXPORT will assume the default extension .CSD. The Command Sequence Definition file is read and copied to the Command Sequence file (.CS), with actual parameters substituted for formal ones.

The following sample file, PROG1.CSD, will assign logical device names to directory files and compile a PASCAL program called PROG1.SRC.

```
ASSIGN :F1: TO /WINCH1.VOL/SYSTEM.DIR<cr>
ASSIGN :F2: TO /WINCH1.VOL/PROJA.DIR<cr>
ASSIGN :F3: TO :F2:PHASE1.DIR<cr>
:F1:RUN :F1:PASC86 :F3:PROG1.SRC<cr>
:F1:RUN :F1:LINK86 :F3:PROG1.OBJ, :F1:P86RN0.LIB, :F1:P86RN1.LIB, &<cr>
:F1:P86RN2.LIB, :F1:P86RN3.LIB, :F1:E8087.LIB, :F1:E8087, &<cr>
:F1:LARGE.LIB TO :F3:PROG1.86 BIND<cr>
```

NOTE

It is not necessary to put a LOGON command into a .CSD file.

Parameters. EXPORT allows for up to 10 formal parameters of the format %n (the two characters must be adjacent with no intervening space), where *n* is a digit 0 through 9. These formal parameters may appear anywhere in the Command List Definition.

Actual parameters are character strings (up to 31 characters) defined by their position in the parameter list (0 is the first parameter) and separated by a comma or a blank. Actual parameters that contain delimiter characters may be entered by embedding the parameter in two apostrophes (' '). EXPORT will allow CTRL-P (Control-P), in parameters and in its input file, to quote the character that follows it. (Control-P literalizes characters in the file.) A NULL actual parameter may be specified by adjacent commas in the parameter list.

All files used by the exported job must reside in the network directory files. Files that reside on local workstation disk directories are not accessible to other workstations.

NOTE

Exported jobs are not executed until a workstation is assigned to the job queue with the IMPORT command. To cancel a job, use the same filename as the LOG file, but with the .CS extension (e.g., TEST1G.CS).

Possible Error Conditions

An error will result when

- You are not logged on
- COMM CONNECTION ABORTED: the physical connection between the Network Resource Manager and the workstation executing the job is broken

A warning message will display when

- No import workstations are currently assigned to a queue

Examples

1. This example exports the file PROG1.CSD to a queue, SERIESIIIJOBS.

```
-EXPORT /WINCH1.VOL/PROJA.DIR/PROG1.CSD TO SIIIJOBS LOG<cr>
YOUR LOGFILE NAME WILL BE PROG1.LOG
-
```

2. This example cancels the LOG file PROGO1.LOG.

```
-CANCEL SIIIJOBS (PROGO1.CS)<cr>
```

3. This example exports a file TEST.CSD to queue Q001 when no import stations are assigned.

```
-EXPORT /WINCH1/TEST.CSD TO Q001<cr>
NO IMPORT STATION CURRENTLY ASSIGNED TO THIS QUEUE
YOUR LOG FILE NAME WILL BE TEST1G.LOG.
```

LOGOFF

Syntax

```
LOGOFF<cr>
```

Description

The LOGOFF command logically disconnects the workstation from the Network and returns directory assignments to the default values. All directory assignments made with the ASSIGN command are returned to their original default value (see Table 4-2).

NOTE

The LOGOFF command automatically invokes the LOGON prompt at the Cluster workstation.

Example

1. Following is the correct use of the command:

```
-LOGOFF<cr>
PLEASE LOGON
USERNAME-
```

LOGON

Syntax

```
LOGON ‹cr›
```

Description

The LOGON command allows a user to LOGON at any time that the system displays the ISIS-III(C) prompt (-). LOGON allows the individual workstation to access the remote shared directories and the background line printer.

NOTE

The previous user is logged off automatically if another user executes LOGON.

The username and password must be established at the Network Resource Manager by the Superuser. See the *NDS-II Network Resource User's Guide.*

Immediately after ISIS-III(C) is initialized or the current user logs off, the system displays the following LOGON prompt:

```
PLEASE LOGON
USERNAME-MYNAME‹cr›
```

When the username is provided, the system prompts you for your password.

```
PASSWORD-
```

where

username	is a string of up to 14 ASCII alphanumeric characters.
password	is a string of up to 14 ASCII alphanumeric characters.

Type your password and a carriage return (‹ cr ›). For security, the characters typed here are not printed on the console.

LOGON establishes communication between the NRM and the workstation operated by the username. The NRM verifies the username/password combination and allows the username workstation to access the shared network files, to submit print requests, and to submit jobs for remote execution.

NOTE

If a home directory exists, LOGON automatically assigns :F9: to the user's home directory. The SUBMIT file :F9:ISIS.INI allows the user to automatically make directory assignments for up to 10 devices at LOGON time.

Possible Error Conditions

An error results when

- The *username* is not validated at the Network Resource Manager
- The network is not running
- The *password* is incorrect

Examples

1. Correct use of this command after the logon prompt appears is as follows:

```
PLEASE LOGON
USERNAME-JOHN <cr>
PASSWORD-PASSB <cr>
-
```

2. An example of an attempt to access the network with an invalid username is as follows:

```
-LOGON<cr>
USERNAME-JON <cr>
PASSWORD-PASSB<cr>
USER NAME NOT KNOWN
-
```

3. This example logs on USERB and logs off a user who had not previously logged off the workstation:

```
-LOGON<cr>
USERNAME-USERB<cr>
PASSWORD-PASS<cr>
PREVIOUS USER LOGGED OFF
-
```

4. This example shows an attempt to logon with an incorrect password. If another user were logged on, he would have been logged off.

```
-LOGON<cr>
USERNAME-USERB<cr>
PASSWORD-PASSBAD<cr>
USERNAME/PASSWORD MISMATCH
```

QUEUE

Syntax

```
QUEUE {ADD   } (queuename [,...]) <cr>
      {DELETE}
```

where

queuename is a string of up to 14 printable characters. Any character may be used except a slash (/) or a blank. *Queuename* identifies where the job is sent for execution.

Description

QUEUE allows the user to create and delete queuenames for remote job execution. A maximum of ten queues can be added or deleted at one time with the QUEUE command. Sixteen is the maximum number of queues that can exist at one time at the NRM.

Possible Error Conditions

An error occurs when

- You are not logged on
- You use illegal syntax
- You use QUEUE under IMPORT
- You use a nonexistent queuename
- Too many queues exist

Examples

1. This example adds the job queues SERIES2 and SERIES3 to the network.

```
-QUEUE ADD(SERIES2, SERIES3)<cr>
 SERIES2, ADDED
 SERIES3, ADDED
```

2. This example deletes the queue SERIES 2 from the network.

```
-QUEUE DELETE(SERIES2) <cr>
 SERIES2, DELETED
```

3. This example attempts to execute a default that is nonexistent.

```
-QUEUE<cr>
 ILLEGAL SYNTAX
```

REMOVE

Syntax

REMOVE { :Fn: / pathname/ } *filename*‹cr›

where

:F*n*:	is the directory identifier assigned to the parent directory of the file to be removed.
pathname	is the complete fully qualified pathname of the parent directory.
filename	is the name of the empty directory or the data file to be removed.

Description

The REMOVE command removes data and empty directory files from the parent directory. To remove a directory file from the file system, the directory file must be empty and you must have access to the directory file.

The sample file structure in Figure 4-2 has an empty directory file DIRA. To remove this file from the file system, type the following:

```
-REMOVE /WINCH1.VOL/SYSTEM.DIR/DIRA‹cr›
/WINCH1.VOL/SYSTEM.DIR/DIRA, REMOVED
```

or:

```
-ASSIGN :F3: TO /WINCH1.VOL/SYSTEM.DIR‹cr›
-REMOVE :F3:DIRA‹cr›
:F3:DIRA, REMOVED
```

Possible Error Conditions

An error will result when

- You are not logged on
- A directory file is not empty
- Any user attempts to remove a directory file and does not have delete access rights to the file

Examples

1. This example shows a successful attempt to remove the empty directory file DIRC from the file system.

```
-REMOVE /WINCH1.VOL/DIRB/DIRC‹cr›
/WINCH1.VOL/DIRB/DIRC, REMOVED
```

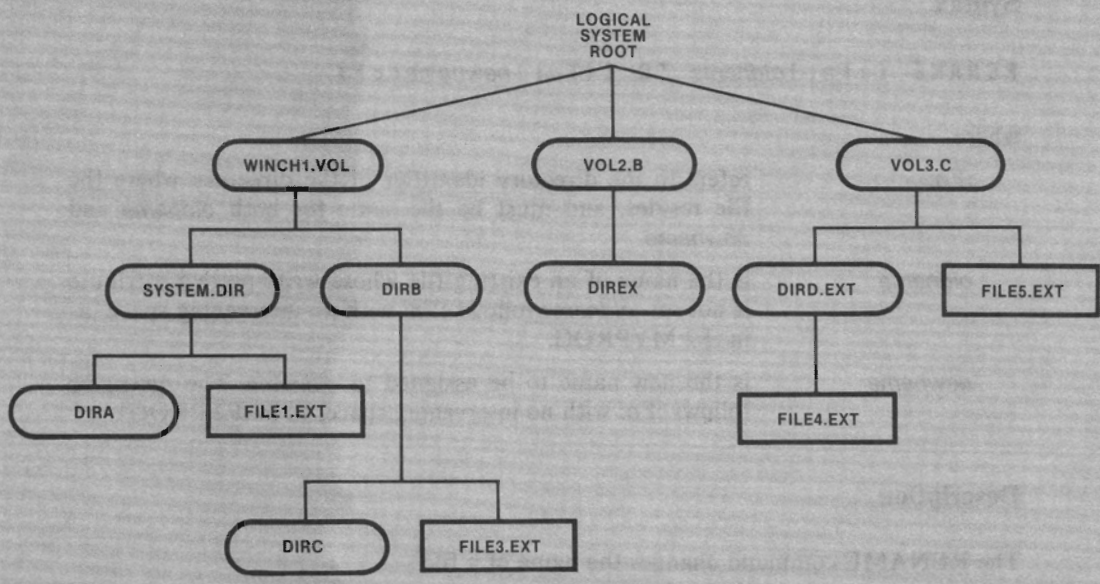


Figure 4-2. Hierarchical File Structure 121765-6

2. This example shows an attempt to remove an empty directory without the proper access rights.

```
-REMOVE /WINCH1.VOL/SYSTEM.DIR/DIRA<cr>
/WINCH1.VOL/SYSTEM.DIR/DIRA,
NETWORK FILE ACCESS RIGHTS VIOLATION
```

3. This example shows an attempt to remove a directory file that is not empty.

```
-REMOVE /WINCH1.VOL/SYSTEM.DIR<cr>
/WINCH1.VOL/SYSTEM.DIR, ATTEMPT TO DELETE A NON-EMPTY DIRECTORY
```

RENAME

Syntax

```
RENAME [:Fn:]oldname TO [:Fn:]newname<cr>
```

where

:F*n*:	refers to the directory identifier of the directory where the file resides, and must be the same for both *oldname* and *newname*.
oldname	is the name of an existing file whose write-protect attribute is not set. *oldname* follows :F*n*: with no intervening space, as in :F2:MYPROG.
newname	is the new name to be assigned to *oldname*. The *newname* follows :F*n*: with no intervening space, as in :F2:PROG1.

Description

The RENAME command changes the name of a file.

Enter the RENAME command to change the name of an existing file to a new name that does not already exist; the system changes the directory. Wildcards cannot be used.

However, if another file with the new name already exists, the system displays the following message:

```
newname, ALREADY EXISTS, DELETE?
```

To delete the existing file, enter a Y or y followed by a carriage return. RENAME will delete the existing file and change the name of *oldname* in the directory.

If the existing file to be deleted has no delete access, or if you enter any character other than Y or y, the existing file is not deleted and the file to be renamed is not renamed.

Possible Error Conditions

An error occurs when

- *oldname* is a nonexistent file
- The directory identifier assignment has not been made
- You do not have proper access rights

Examples

1. This example changes the name of a file on directory 0 from CHAP1 to CHAP.ONE.

```
-RENAME CHAP1 TO CHAP.ONE<cr>
-
```

2. This example illustrates an attempt to rename a file with no delete access.

```
-RENAME NEWPRG.TXT TO PROGA.TXT<cr>
NEWPRG.TXT, WRITE PROTECTED
-
```

3. In this example, the new name is the name of an existing file.

```
-RENAME TEXT.BAK TO TEXT.OLD<cr>
TEXT.OLD, ALREADY EXISTS, DELETE? Y<cr>
-
```

4. This example shows an attempt to rename a file and move to another disk or directory.

```
-RENAME :F6:OLDFIL TO :F7:NEWFIL<cr>
:F7:NEWFIL, NOT ON SAME DISK
-
```

5. This example shows an attempt to change a network filename to a filename from the remote directory.

```
-RENAME :F6:NETWRK.FLE TO :F6:ALRDY.USD<cr>
ALRDY.USD, ALREADY EXISTS, DELETE?
```

6. To rename a file in the remote network file system, assign a directory identifier to the directory first.

```
-ASSIGN :F3: TO /VOLUME/MYBOOK<cr>
-RENAME :F3:CHAP1B TO :F3:CHAP2<cr>
```

SPACE

Syntax

```
SPACE  /volume name<cr>
```

where

/volume name is the volume root directory (e.g., /VOL1.A) for the given physical device.

Description

The SPACE command returns information about the amount of available space on any given disk at the specific point in time. Information is returned in the following format:

```
VOLUME GRANULARITY  = number (granularity = number of bytes/sector)
FREE BLOCKS         = number
TOTAL BLOCKS        = number
FILES AVAILABLE     = number
TOTAL FILES         = number

mm/dd/yy hh:mm:ss
```

where

mm/dd/yy hh:mm:ss is the month, day, year, hour, minutes and seconds.

number is a decimal integer.

NOTE

In a multiprogramming environment, the amount of space is being used by several users concurrently. Therefore, the information returned by the SPACE command should be considered an approximation.

Possible Error Conditions

- You are not logged on
- Volume name does not exist
- Illegal syntax of pathname is used

Example

1. In this example information for the volume whose root directory is /VOL1.A is provided.

```
-SPACE /VOL1.A<cr>
VOLUME GRANULARITY  =   512
FREE BLOCKS         = 25144
TOTAL BLOCKS        = 58995
FILES AVAILABLE     =  4796
TOTAL FILES         =  6000
1/5/83              11:31:08
```

SUBMIT

Syntax

```
SUBMIT [:Fn:]filename[(parameter[,...])]<cr>
```

where

:F*n*: is the directory identifier assigned to the directory that contains *filename*.

filename is the name (and extension, if any) of the shared file that contains the command sequence definition. If the extension is omitted, SUBMIT assumes the default extension .CSD.

parameter is an actual value that is to replace a formal parameter in the command sequence definition file. The maximum number of parameters allowed is 10. If you omit a parameter from the SUBMIT list, enter a comma in its place.

A parameter is a character string of up to 31 characters. Any ASCII character from 20H to 7AH is legal, except a comma, space, or right parenthesis. If a parameter contains a comma, space, or right parenthesis, enclose the parameter in apostrophes. To use a quotation mark inside a quoted parameter, use two apostrophes in its place. For example,

```
'TITLE(''QUOTE ('') SEARCH ROUTINE'')'
```

is used in the final command as:

```
TITLE('QUOTE (') SEARCH ROUTINE')
```

Description

The SUBMIT command causes ISIS-III(C) to take its commands from a file rather than from the console.

SUBMIT uses two files:

- A command sequence definition (CSD) file that contains the command sequence definition. Create this file with formal parameters, using the editor.
- A temporary .TMP file that contains the command sequence to be executed.

  SUBMIT creates this file with the actual parameters supplied in the SUBMIT command that replaces the formal parameters. The temporary file has the same name as the command sequence definition file but with the extension .TMP. Do not modify this file.

  SUBMIT reassigns the console input device to the .TMP file it creates and returns control to ISIS-III(C), which then executes the commands in the .TMP file. The .TMP file has a final command that restores the console input device to its former device assignment and deletes the .TMP file.

When creating the CSD file, specify formal parameters by using two characters, %*n*, where *n* is a digit from 0 through 9. You may place formal parameters anywhere in the CSD file. To enter a percent sign (%) that is not to be interpreted as a formal parameter, enclose it in apostrophes.

Any program, except a LOGON or LOGOFF, that reads its commands from :CI: noninteractively can be executed.

The CSD file can also contain commands to the programs being run. Using a SUBMIT command in a CSD file causes another .TMP file to be created. SUBMIT commands can be nested to any depth.

A CTRL-E (↑E) in a .TMP file switches the console input from the .TMP file to the initial system console, allowing interactive processing. To return control to the .TMP file, enter CTRL-E at the console. If control is *not* returned to the .TMP file, or if an error occurs after a command sequence has started processing, control returns to ISIS-III(C) and the .TMP file is not deleted.

Any program running under SUBMIT must allow two buffers in addition to the open files and buffers required by the program itself. See the *Intellec® Series III Microcomputer Development System Programmer's Reference Manual* (121618) for information on how to determine the base address of your program.

Possible Error Conditions

Do not use LOGON or LOGOFF commands in a SUBMIT file. Including either of these commands in the file causes an error message at the workstation.

```
ILLEGAL LOGON WHILE :CI:/:CO: FILE ON NETWORK
```

or

```
ILLEGAL LOGOFF WHILE :CI:/:CO: FILE ON NETWORK
```

Example

1. This example shows a PL/M-80 compilation, a LINK, and a LOCATE executed from a SUBMIT file on a two directory system. A CTRL-E is entered in the command sequence definition after the PL/M compilation so you can examine the results of the compile. The EDIT does not echo the ↑E; however, it is echoed when the SUBMIT file is executed.

   The file CMPLNK.CSD in :F1: contains the following command sequence definition. See the *MCS®-80/85 Utilities User's Guide for 8080/8085-Based Development Systems* (121617) for an explanation of controls in the PL/M-80 command. The CMPLNK.CSD file contains

```
PLM80 %0.%1 DEBUG XREF DATE(%2)
↑E
LINK %0.OBJ,SYSTEM.LIB TO %0.SAT&
PRINT(%0.MP1) MAP
LOCATE %0.SAT PRINT (%0.MP2) MAP
```

Enter the following SUBMIT command to compile, link, and locate PROGA.SRC:

```
-SUBMIT :F1:CMPLNK (:F1:PROGA.SRC, '3 OCT 81')<cr>
```

The command sequence actually executed is shown as it would be echoed on the console output device.

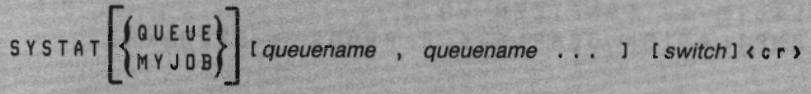
```
-PLM80 :F1:PROGA.SRC DEBUG XREF DATE(3 OCT 81)

ISIS-II PL$b1M-80 COMPILER V3.1
PL/M-80 COMPILATION COMPLETE 0 PROGRAM ERROR(S)

-↑E↑E
-LINK :F1:PROGA.OBJ,SYSTEM.LIB TO :F1:PROGA.SAT &
**PRINT(:F1:PROGA.MP1) MAP
-LOCATE :F1:PROGA.SAT PRINT(:F1:PROGA.MP2) MAP
-:F0:SUBMIT RESTORE :F1:CMPLNK.TMP(:VI:)
-
```

SYSTAT

Syntax



where

QUEUE	lists all of the names and numbers of jobs currently in the queues, or it lists a specific queue, the job status, and the names of the users who own the jobs.
MYJOB	lists the names, numbers, and status of jobs exported to a specified queue by the current user.
queuename	is the name assigned to a queue where the job was sent to await execution.
switch	is E that specifies expanded listing. All cancelled, aborted, executing, and waiting jobs are listed.

Description

The SYSTAT command will list remote job queues, and report the jobname, job number, and the job status.

Listing Active Queues. To list all of the active queues, type the following:

```
-SYSTAT<cr>
        QUEUE NAME              JOBS        IMPORT
                              WAITING     STATIONS
SIIIJOBS                        6           2
SIIJOBS                         3           1
LOWPRIORJOBS                    4           0
```

where

QUEUE NAME	is the list of all of the currently active queues.
JOBS WAITING	is the number of jobs in that queue waiting to be executed by an import station.
IMPORT STATIONS	is the number of workstations currently serving jobs from this queue.

Listing A Specific Queue. To list all of the jobs in a queue, type the following:

```
-SYSTAT QUEUE (SIIIJOBS)<cr>
    JOB NAME    OWNER       NUMBER     DATE       TIME       STATUS
  QUEUE:SIIIJOBS
  TEST.CSD      JOHN          A3       12/02/83   09:30:00   WAIT
  COMPLE.CSD    SHEILA        27       12/02/83   10:37:34   WAIT
  TEST.CSD      PETE          1A       12/01/83   04:03:12   EXECUTING
```

where

JOB NAME	is the name of the queue requested.
OWNER	is the owner of a specific file in the queue.

NUMBER	is the job number assigned by the Network Resource Manager when the job was exported. This unique hexadecimal number is assigned to avoid system confusion between two jobs of the same name with different owners. The job number can also be used in the CANCEL command.
DATE	is the date that the job was exported to the queue.
TIME	is the time that the job was exported to the queue.
STATUS	is the current status of the job, one of the following:

where

ABORTED	means the job was interrupted before it was completed. Reviewing the log file will help to determine the cause.
CANCELLED	means the job has been cancelled by the owner or the Superuser.
DONE	means the job has been executed successfully and is completed.
EXECUTING	means the job is currently executing at an import station.
WAIT	means the job is waiting in the queue to be executed by an import station.

Possible Error Conditions

An error will occur when

- You are not logged on
- A specified queue does not exist
- A job does not exist

Examples

1. This example lists all of the active queues.

```
-SYSTAT<cr>
        QUEUE NAME              JOBS          IMPORT
                              WAITING       STATIONS
SIIIJOBS                        6             2
SIIJOBS                         3             1
LOWPRIORJOBS                    4             0
```

2. This example lists all of the jobs in the SIIIJOBS queue.

```
-SYSTAT QUEUE (SIIIJOBS)<cr>
    JOB NAME    OWNER       NUMBER     DATE       TIME       STATUS
  QUEUE:SIIIJOBS
  TEST.CSD      JOHN          A3      12/02/83   09:30:00   WAIT
  COMPLE.CSD    SHEILA        27      12/02/83   10:37:34   WAIT
  TEST.CSD      PETE          1A      12/01/83   04:03:12   EXECUTING
```

3. This example lists the status of all the jobs exported by one user.

```
-SYSTAT MYJOB<cr>
  JOB NAME      NUMBER        DATE         TIME       STATUS
 QUEUE:SIIIJOBS
 COMPLE.CSD       27         12/02/83     10:27:34    WAIT
 QUEUE:SIIJOBS
 TEST.CSD         2A         12/02/83     09:30:00    WAIT
 MYPROG.CSD       19         12/01/83     08:57:04    EXECUTING
 QUEUE:LOWPRIORJOBS
```

VERS

Syntax

```
VERS command‹cr›
```

where

command is one of the ISIS command programs.

Description

The VERS command lists the version number of the ISIS-III(C) operating system command programs.

The VERS command identifies the version number of the different versions of ISIS operating system command programs. Each version works correctly only with the corresponding version of ISIS.

ISIS-III(C) operates with any command program that displays a version number followed by C and has a version number 2.2 or greater. Other Intel software such as editors and translators do not contain version numbers.

Examples

1. This example successfully lists the version number of a compatible ISIS-III(N) command program.

```
-VERS DIR‹cr›
DIR V2.2N
```

2. This example shows an attempt to list the version number of a user file.

```
-VERS MYFILE.EXT‹cr›
file does not contain a program version number
```

3. This example shows an attempt to list the version number of a file not in the directory of the target device.

```
-VERS NONFLE‹cr›
ERROR 13 USER PC 375B
```

4. This example lists the version number of the command VERS.

```
-VERS VERS‹cr›
VERS V2.2N
```

WHO

Syntax

```
WHO<cr>
```

Description

The WHO command displays the name of the user who is currently logged on.

Possible Error Conditions

An error will occur when

- You are not logged on

Example

1. This example displays the name of the user.

```
-WHO <cr>
 I AM SUSAN
-
```

CHAPTER 5
CLUSTER WORKSTATION SYSTEM CALLS

This chapter briefly summarizes the changes in the ISIS-III(C) system calls, lists all of the ISIS-III(C) routines available on the NDS-II system, and *describes only those routines which are not identical* to those described in the *ISIS-II User's Guide* (9800306).

Differences Between ISIS-III(C) and ISIS-II

Shaded text throughout this chapter highlights the differences between the ISIS-III(C) and the ISIS-II operating systems.

Revised System Calls

In ISIS-III(C), the ATTRIB, GETATT, GETD, and SPATH system calls are slightly different.

The ATTRIB and the GETATT calls only recognize the write-protect attribute for a network file.

The GETD call accesses information from the file system of the Network Resource Manager.

The SPATH call defines a directory type for NDS-II remote mass storage.

New System Calls

The CHGACS call allows the owner to change the owner or world access rights of a file.

The DETIME call returns the current date and time.

The FILINF call provides security and accounting information on a file.

Summary of System Calls

The ISIS-III(C) services that can be called by a program include

- Input/output operations for the disk directories (FILINF, OPEN, CLOSE, READ, WRITE, SEEK, RESCAN, SPATH)

NOTE

:SP: is a queuename and should not be used in programmatic calls.

- Disk directory maintenance (ATTRIB, CHGACS, DELETE, GETATT, GETD, RENAME)
- Console device assignment, error message, and date and time output (CONSOL, WHOCON, ERROR, DETIME)
- Program loading and execution and return to the supervisor (LOAD, EXIT)

For descriptions of these system calls, refer to the *ISIS-II User's Guide.*

To call ISIS-III(C), specify the services desired and the address of the parameter list. Note that an ISIS routine does not affect the user stack depth, and that a call to ISIS-III(C) destroys the contents of the CPU registers.

The system call descriptions include discussion of the operation of the call and the parameters the program must supply.

To clarify the effect of certain system calls on the files, two integer quantities (LENGTH and MARKER) are associated with each file. LENGTH indicates the number of bytes in the file. MARKER indicates the number of bytes already read or written in the file (that is, it acts as a file pointer to the current position).

System Call Syntax and Usage

ISIS-III(C) system routines can be called from a PL/M-80 or Assembly Language program. If the program makes an ISIS-III(C) system call, link the object program with SYSTEM.LIB using the LINK program.

SYSTEM.LIB, a library file supplied with the ISIS-III(C) system disk directory, contains the procedures necessary to interface the user's programs that contain ISIS-III(C) system calls with the ISIS-III(C) operating system.

PL/M-80 Calls

Any PL/M-80 program can interface to ISIS-III(C) by performing calls to procedures in SYSTEM.LIB. Programs must include external procedure declarations so the proper procedures from SYSTEM.LIB will be included with the program by LINK. These external procedure declarations may be declared as type ADDRESS, but may also be values as well as addresses of values.

Assembly Language Calls

The interface between the 8080/8085 Assembly Language program and ISIS-III(C) is accomplished by calling the desired ISIS-III(C) routine. All system calls conform to the interface described in the *ISIS-II User's Guide.*

The ISIS-III(C) entry point is defined in a routine in SYSTEM.LIB that must be included in the program. When using LINK, specify the name of the program followed by the name SYSTEM.LIB. See the *MCS®-80/85 Utilities User's Guide* for more information on LINK.

System call references can be defined in EXTRN statements before they are referenced in a program, thereby allowing the program to reference the system call routines symbolically. Only the specific system calls needed by the program need be defined.

ATTRIB

The ATTRIB call allows your program to change an attribute of a directory file.

You must pass four parameters in the ATTRIB call. These parameters are

1. *path$p*, the address of an ASCII string that contains the name of the file whose attribute is to be changed. The string can contain leading space characters but cannot contain embedded spaces. The string must be terminated by a character other than a letter, digit, colon (:), or period (.). A space can be used.
2. *atrb*, a number that indicates which attribute is to be changed. The numbers are

   0 — invisible attribute
   1 — system attribute
   2 — write-protect attribute
   3 — format attribute

   Only the write-protect attribute (2) may be set or reset on remote files. This attribute will set or reset the access right switches (Owner Write, Owner Delete, World Write, and World Delete) of the remote file.
3. *onoff*, a value that indicates whether the attribute is to be set (turned on) or reset (turned off). The value is stored in the low order bit of the low order byte. A value of 1 specifies that the attribute be set; a value of 0 specifies that it be reset.
4. *status$p*, the address of a memory location for the return of a nonfatal error number.

   Nonfatal error codes returned: 4,5,13,23,26,28,61,66,70

   Fatal error codes returned: 1,24,30,33,63,64,65

CHGACS

Syntax

```
CALL CHGACS (path$p, class, access, status$p)
```

The CHGACS call requires four parameters:

1. *path$p*, a 16-bit pointer to a string containing the pathname of the file whose access rights are to be changed.
2. *class*, a 1-byte value that specifies the class of users whose access rights are to be changed.

Value	Type
0	owner of file
1	world
2-255	reserved

3. *access*, a 1-byte value that specifies the type of access to be granted to the class of specified file users. If all bits are set to 0, the access of the specified user to the file is denied. If any bits are set to 1, access is granted as follows:

Bit	Access
0	delete (data), delete (directory)
1	read (data), display (directory)
2	write (data), add-entry (directory)
3	update (read and write)

4. *status$p*, a 16-bit value that specifies the address of a memory location for the return of a nonfatal error.

Standard Exception Codes

The following exception codes that are returned to the user are equivalent to Series III UDI exception codes:

E$OK (0000H)
E$FNEXIST (0021H)
E$FACCESS (0026H)
E$SUPPORT (0023H)

Description

The CHGACS call allows the user to change the owner or world access rights of a file. Files whose access rights are to be changed are assumed to be open before the CHGACS call is made.

NOTE

The privilege to use the CHGACS function on remote files is granted only to the owner of the file. The granting of this privilege to other users is operating system dependent. If the privilege is not granted, the sytem returns the error E$FACCESS. The error E$FNEXIST indicates that the file does not exist. If the user attempts to change the access rights of a non-disk file, the error E$SUPPORT is returned. The access rights of the file are changed immediately but do not affect other connections to the file until they are detached.

Example

1. *PL/M-80 CHGACS Call*

```
CHGACS:
  PROCEDURE (path$p, class, access, status$p) EXTERNAL;
  DECLARE     path$p      address,
              class       byte,
              access      byte,
              status$p    address;
  END CHGACS;

  DECLARE ABC$DEF(*) BYTE INITIAL (11,':F1:ABC.DEF ');
  DECLARE OWNER$CLASS LITERALLY '0';
  DECLARE WORLD$CLASS LITERALLY '1';
  DECLARE DRW$ACCESS LITERALLY '07H';

  DECLARE (ENTRY,EXCEP,STATUS) ADDRESS;
  DECLARE AFTN ADDRESS;

  CALL OPEN(.AFTN,.ABC$DEF(1),1,0,.STATUS);

  CALL LOAD(.('ISIS.OV2 '),0,0,.ENTRY,.STATUS);

  CALL CHGACS(.ABC$DEF,OWNER$CLASS,DRW$ACCESS,.STATUS);
  IF STATUS <> THEN ERROR ...
```

DETIME

Syntax

```
CALL DETIME (dt$p, status$p)
```

The DETIME call requires two parameters:

1. *dt$p* is a pointer to user declared structure in the following form:

```
DECLARE DT STRUCTURE(
                sys$time$lo     address,
                sys$time$hi     address,
                date(8)         byte,
                time(8)         byte);
```

where

sys$time is a formatted double address that contains the date and time. If *sys$time* is zero, the system clock is read first to obtain the current date and time. If *sys$time* is not zero, it is simply decoded into ASCII date and time strings. *sys$time* contains the binary format of the current date and time if selected by the input value zero. The specified format is in seconds beginning with 1/1/78.

date has the form *mm/dd/yy* for month, day, and year. The value for hours ranges from 0 through 23.

2. *status$p* is the address of a memory location for the return of a nonfatal error number.

Standard Exception Codes

E$OK (0000H)

NOTE

On remote files the date and time that is maintained at the NRM is returned.

Description

The DETIME call decodes the *sys$time$lo* and *sys$time$hi* variables into ASCII date and time strings. It may also be used to return the current date and time in either binary address format or as a decoded ASCII string.

Example

1. *PL/M-80 DETIME Call*

```
DETIME:
  PROCEDURE (dt$p,status$p) EXTERNAL;
    DECLARE  dt$p           address,
             status$p       address;
  END DETIME;
```

```
DECLARE DT STRUCTURE(
            sys$time$lo     address,
            sys$time$hi     address,
            date(8)         byte,
            time(8)         byte);
DECLARE (ENTRY,EXCEP,STATUS) ADDRESS;
CALL LOAD (.('ISIS.OV2 '),0,0,.ENTRY,.STATUS);
DT.SYS$TIME$LO,DT.SYS$TIME$HI = 0;
CALL DETIME(.DT,.STATUS)
IF STATUS <> THEN 0 THEN CALL ERROR (STATUS)
```

FILINF

Syntax

```
CALL FILINF (file$table$p, mode, file$info$p, status$p)
```

The FILINF call requires four parameters:

1. *file$table$p* points to a structure of the form

```
DECLARE file$table      structure(
        aftn            address,
        device$num      byte,
        name(6)         byte,
        extension(3)    byte,
        device$type     byte,
        drive$type      byte);
```

where

aftn	contains the active file table number of the open file on which security and accounting information is desired.
device$num, name, extension, device$type, drive$type	contain information normally returned from the SPATH call.

2. *mode* indicates whether the file owner is to be identified. Byte 0 indicates that the owner name or identification is not to be returned. Byte 1 indicates that the owner name or identification is to be returned.

3. *file$info$p* points to the structure that the user declares to receive the file information. The structure form is

```
DECLARE FILE$INFO STRUCTURE
        (owner(15)          byte,
        length$of$file(2)   address,
        type                byte,
        owner$access        byte,
        world$access        byte,
        create$time(2)      address,
        last$mod$time(2)    address,
        reserved(20)        byte);
```

For remote files these fields are interpreted as follows:

owner — the string that identifies the system name of the file owner

lengthoffile — the length of the file in bytes

type — an indication of the usage of the file where the type is indicated by the following values:

Value	Type
0	data file
1	directory file
2	reserved
.	
.	
225	

owner$access, world$access — describe the owner and world file access rights

where

Bit	Access
0	delete (data), delete (directory)
1	read (data), display (directory)
2	write (data), add-entry (directory)
3	update (read and write)
4-7	reserved

create$time — indicates the date and time of the creation of the file. The contents of the time *address* array is the number of seconds since 1/1/78.

lastmodtime — indicates the date and the time of the last modification of the remote file. It is a 4-byte array.

4. *status$p*, the address of a memory location for the return of a nonfatal error number.

Standard Exception Codes

The following exception codes that are returned to the user are equivalent to Series III UDI exception codes:

E$OK (0000H)
E$SUPPORT (0023H)

Description

The FILINF call allows the user to determine security and accounting information about a file. Files for which the FILINF function is invoked are assumed to be opened before the FILINF call is made. If the file for which FILINF is invoked does not have the read bit on, an error occurs.

Example

1. *PL/M-80 FILINF Call*

```
FILINF:
  PROCEDURE (file$table$p, mode, file$info$p, status$p) EXTERNAL;
  DECLARE   file$table$p        address,
            mode                byte,
            file$info$p         address,
            status$p            address;
  END FILINF;

  DECLARE   FILE$INFO STRUCTURE(
            owner(15)           byte,
            length(2)           address,
            type                byte,
            owner$access        byte,
            world$access        byte,
            create$time(2)      address,
            last$mod$time(2)    address,
            reserved(20)        byte);
```

```
DECLARE     file$table          structure(
            aftn                address,
            device$num          byte,
            name(6)             byte,
            extension(3)        byte,
            device$type         byte,
            drive$type          byte);
DECLARE   (entry, excep, status) ADDRESS;
DECLARE ABC$DEF (*) BYTE INITIAL (':F1:ABC.DEF ');
DECLARE AFTN ADDRESS;
CALL SPATH(.ABC$DEF,.FILE$TABLE.DEVICE$NUM,.STATUS);
CALL OPEN(.AFTN,.ABC$DEF,1,0,.STATUS);
CALL LOAD(.('ISIS.OV1 '),0,0,.ENTRY,.STATUS);
CALL FILINF(.FILE$TABLE,1,.FILE$INFO,.STATUS);
IF STATUS <> 0 THEN ERROR ...
```

GETATT

Syntax

```
CALL GETATT (file$pointer, .attrib$p, .status)
```

The information is returned in a one byte field (ATT),

where

bit 0 set	= invisible
bit 1 set	= system
bit 2 set	= write-protect
bit 3	= reserved
bit 4	= reserved, undefined (1 or 0)
bit 5	= reserved
bit 6	= reserved
bit 7 set	= format

Description

On network files, only the write-protect attribute is valid.

The GETATT call allows access to the attribute information associated with the disk file pointed to by *file$pointer*. If the specified file is nonexistent, a nonfatal error occurs.

Examples

1. *PL/M-80 GETATT Call*

```
GETATT:
 PROCEDURE (path$p,attrib$p,status$p) EXTERNAL);
  DECLARE (path$p,attrib$p,status$p) address;
END GETATT;
/* MUST BE LINKED WITH SYSTEM.LIB */
.
.
.
  DECLARE FILE(15) BYTE;
  DECLARE STATUS ADDRESS;
  DECLARE ATTRIB BYTE;
  .
  .
  .
  .
  CALL GETATT (.FILE, .ATTRIBUTE, .STATUS);
  IF STATUS <> 0 THEN ...
```

2. *Assembly Language GETATT Call*

```
            EXTRN   GETATT              ; MUST BE LINKED
                                        ; WITH SYSTEM.LIB
;
; GETATT
            LHLD    FILEP               ; FILE PARAMETER
            PUSH    H
            LHLD    ATTP                ; ATTRIBUTE POINTER
            PUSH    H
            POP     B
            LHLD    STATP               ; STATUS
            PUSH    H
            POP     D
            CALL    GETATT
            LDA     GSTAT               ; TEST ERROR STATUS
            ORA     A
;...        JNZ     EXCEPT              ; BRANCH TO EXCEPTION
                                        ; ROUTINE
FILEP:      DW      GFILE               ; POINTER TO FILE
                                        ; PATHNAME
ATTP:       DW      ATTRIB              ; POINTER TO ATTRIBUTE
STATP:      DW      GSTAT               ; POINTER TO STATUS
;
GFILE:      DB      ':F0:FILE.EXT '     ; FILE PATHNAME
ATTRIB:     DS      1                   ; ATTRIBUTE VALUE
                                        ; (RETURNED)
GSTAT:      DS      2                   ; STATUS (RETURNED)
```

GETD

Syntax

```
CALL GETD (did, .conn, count, .actual, .table, .status)
```

The GETD call requires five parameters:

1. *did*, an address value that contains the number for the directory from which entries are to be returned. Valid integer values are 0-9 representing :F0:-:F9:.
2. *conn*, an address value that must be initialized to zero when the first display request of a sequence is made. Do not change the value; changing it will cause an error. The system then assigns a value that is returned in subsequent requests for directory entries. This value acts as a pseudo-connection for all subsequent calls.
3. *count*, an address value that contains the number of entries to be returned. *count* = 0 indicates termination of the current display request sequence and releases the pseudo-connection. The program must make a final call to GETD with *count* = 0 to release the connection, except when it is released automatically as described under *actual*.
4. *actual*, an address value that contains the number of directory entries returned by the system. When the number of directory entries is less than *count*, the last directory entry has already been returned and the pseudo-connection will be released automatically.
5. *table*, the address of a memory structure where directory entries are returned. The structure form is

```
DECLARE ENTRY STRUCTURE(
    RESERVED1 (1)    BYTE,
    FILE$NAME (9)    BYTE,
    RESERVED2 (6)    BYTE);
```

   where

   file$name is a nine-byte field with two subfields left justified and zero filled. The first six bytes represent the name; the remaining three bytes are the file extension.

6. *status*, the address of a memory location for the return of a nonfatal error number.

   Nonfatal error codes returned: 3, 4, 5, 13, 23

   Fatal error codes returned: 1, 24, 30, 33

Description

The GETD call allows access to information in the file directory defined by DID in the Network Resource Manager file system.

Reserved fields returned by GETD are different from one ISIS-III(C) system call to another. Do not use reserved fields in a user program.

Programs that use this interface will execute correctly on a standalone development system (Model 800, Series II, or Series III) or a workstation attached to an NDS-II network. GETD requires that an ISIS overlay and an NDS-II overlay, ISIS.OV0, be

loaded into the top of a 64K memory space beginning at 0E800H with the LOAD system call before calling GETD. The overlay space can be overwritten when this call is completed.

Access the overlay entry point by linking to SYSTEM.LIB. Avoid overwriting the overlay when accessing the desired directory.

Examples

1. *PL/M-80 GETD Call*

```
GETD:
 PROCEDURE (did, conn$p, count, actual$p, table$p, status$p)
   EXTERNAL;
  DECLARE (did, conn$p, count, actual$p, table$p, status$p) ADDRESS;

 END GETD;

 DECLARE (dummy, status, conn, actual) ADDRESS;

 DECLARE TABLE (50) STRUCTURE
                   (reserved (1) byte,
                    filename (9) byte,
                    reserved (6) byte);

 CONN = 0;

 CALL LOAD (.(':F0:ISIS.OV0 '),0,0, .DUMMY,.STATUS);

 CALL GETD (1,.CONN, 50, .ACTUAL, .TABLE (0), .STATUS);

 IF STATUS <> 0 THEN ERROR ...
```

2. *Assembly Language GETD Call*

```
;
; GETD
              EXTRN     GETD        ; MUST BE LINKED
                                    ; WITH SYSTEM.LIB
; ISIS.OV0 MUST BE LOADED BEFORE CALLING GETD
              LXI       H,0
              SHLD      CONNP       ; INITIAL 0
              LHLD      DID         ; DIRECTORY
              PUSH      H
              LXI       H,CONNP     ; CONNECTION POINTER
              PUSH      H
              LHLD      COUNT       ; COUNT
              PUSH      H

              LHLD      ACTP        ; ACTUAL
              PUSH      H

              LHLD      BUFFP       ; BUFFER POINTER
              PUSH      H
              POP       B
              LHLD      GSTAT       ; STATUS
              PUSH      H
              POP       D

              CALL      GETD
              LDA       DSTAT       ; TEST ERROR STATUS
              ORA       A
```

```
                JNZ         EXCEPT      ; BRANCH TO EXCEPTION
                                        ; ROUTINE
;
;
DID:            DW          1           ; DIRECTORY IDENTIFIER
CONNP:          DW          0           ; DIRECTORY CONNECTION
COUNT:          DW          8           ; ENTRY COUNT
ACTP:           DW          DACT        ; POINTER TO ACTUAL
BUFFP:          DW          DBUF        ; POINTER TO BUFFER
GSTAT:          DW          DSTAT       ; POINTER TO STATUS
DACT:           DS          2           ; COUNT OF ENTRIES READ
                                        ; (RETURNED)
DBUF:           DS          128         ; DIRECTORY BUFFER
DSTAT:          DS          2           ; STATUS (RETURNED)
;
```

SPATH

The SPATH call allows your program to obtain information relating to a specified file. The information returned by this call includes the device number, filename and extension, device type, and if a disk file, the drive type.

You pass three parameters in the SPATH call. These parameters are

1. *path$p*, the address of an ASCII string that contains the name of the file for which information is requested. The string can contain leading spaces but cannot contain embedded spaces. The string must be terminated by a character other than a letter, digit, colon (:), or period (.). A space can be used.
2. *info$p*, the address of a 12-byte memory location in which the system will return the information. After the call is completed, the buffer will contain the following information:

   Byte 0 — Device number
   Bytes 1 through 6 — Filename
   Bytes 7 through 9 — Filename extension
   Byte 10 — Device type
   Byte 11 — Drive type
3. *status$p*, the address of a memory location for the return of a nonfatal error number.

   Nonfatal error codes returned: 4,5,23,28

   Fatal error code returned: 33

The possible values for the contents of the *info$p* device number are the following:

0 — directory 0*
1 — directory 1*
2 — directory 2*
3 — directory 3*
4 — directory 4*
5 — directory 5*
6 — teletype input
7 — teletype output
8 — CRT input* (:VI:)
9 — CRT output* (:VO:)
10 — user console input
11 — user console output
12 — teletype paper tape reader
13 — high speed paper tape reader
14 — user reader 1
15 — user reader 2
16 — teletype paper tape punch (teletype)
17 — high speed paper tape punch
18 — user punch 1
19 — user punch 2
20 — line printer
21 — user list 1
22 — byte bucket (a pseudo input/output device)*
23 — console input* (:CI:)
24 — console output* (:CO:)
25 — directory 6*
26 — directory 7*
27 — directory 8*
28 — directory 9*
29 — spool printer queue (:SP:)*

* — supported by ISIS-III(C)

The filename and extension are the ISIS filename, e.g., SAMPLE.SRC, without the period.

The device type specifies the type of peripheral with which the file is associated. The possible values for this field are the following:

0 — sequential input device*
1 — sequential output device*
2 — sequential input/output device
3 — random access input/output device*

* — supported by ISIS-III(C)

The drive type field specifies the type of drive controller if the device type field is 3. If the device type is anything except 3, the drive type is undefined. The possible values for a device type of 3 are the following:

0 — controller not present
1 — two-board double density
2 — two-board single density
3 — integrated single density
4 — reserved
5 — NDS-I remote mass storage (valid for NDS-I only)
6 — NDS-II remote mass storage (valid for NDS-II only)*
7 — reserved
8 — unassigned*

* — supported by ISIS-III(C)

CHAPTER 6
SYSTEM CONSIDERATIONS

This chapter describes useful ways to improve system performance and to avoid problems.

Mainframe Link

The Mainframe Link program RCOM80 running on the workstation can access only files at the local workstation; it cannot access files stored in the shared hard disk directories; therefore, it cannot work with Cluster workstations. You must access the Mainframe Link program from a host development system workstation.

ISIS-III(C) Command Files

The system disk contains command program files of the same name as the ISIS-III(C) commands (DIR, COPY, DELETE, etc.). Do not interchange these files with any command program files from other versions of ISIS; the commands will not operate correctly. Use the VERS command to verify the version number and compatibility of the command program file (see Chapter 4).

Home Directory

When a new user is defined by the Superuser at the NRM, the user is assigned a home directory. Users can build a SUBMIT file called ISIS.INI in their home directory. Commands in ISIS.INI will be executed automatically at LOGON time. Consequently, directory identifiers to commonly used pathnames can be set up automatically without user involvement at LOGON time.

CHAPTER 7
ERROR MESSAGES

This chapter describes error codes and messages unique to the NDS-II operating system software. For descriptions of other ISIS-III(C) error codes and messages, see Appendix A of this manual.

New ISIS-III(C) Error Messages

61. DEVICE NOT ASSIGNED.

The user has attempted to open a file on a logical device that has not been assigned to a directory. Use the ASSIGN command to map the device to the directory.

63. SYNCHRONIZATION ERROR.

Communication messages between ISIS and the Network Resource Manager are not synchronized. Reboot the ISIS software.

64. NETWORK COMM ERROR.

The Network Resource Manager comm board is malfunctioning or missing.

65. LOCAL COMM ERROR.

The Series II comm board is malfunctioning or missing.

66. ILLEGAL ATTRIBUTE FOR REMOTE FILE.

User has attempted to change the system, the format, or the invisible attribute on a network file. The write attribute is the only attribute supported on remote network files. Attributes are changed with the ATTRIB command or the ATTRIB system call.

70. NETWORK FILE ACCESS RIGHT VIOLATION.

User attempted to open a network file for read access without read access rights to the file. File access rights can be changed with the ACCESS command.

71. ILLEGAL OPERATOR ON SHARED FILE.

User attempted to write to a file that was being accessed by more than one user. When all other users have exited from the file, try again.

72. MAXIMUM NUMBER OF FILES ON A DEVICE EXCEEDED.

User attempted to create more than the maximum number of files allowed for a particular device. The maximum number of files per device is determined when the System Generation procedure is run.

73. ATTEMPT TO DELETE A NON-EMPTY DIRECTORY.

User attempted to delete a network directory that contains files. Delete all files in the directory before deleting the directory.

74. ILLEGAL NETWORK PATHNAME SYNTAX.

User specified a fully qualified DFS pathname in a CREATE command and did not begin the pathname with a slash (/). All fully qualified pathnames must begin with a slash (/).

75. NONTERMINATING PATH ELEMENT IS NOT A DIRECTORY.

User specified a fully qualified pathname in a CREATE command where one of the first elements of the pathname was a data file. Only the final element of the pathname can be a data file.

76. ATTEMPT TO CREATE A CONNECTED NETWORK FILE.

User assigned :F*n*: to /ROOT/A and then attempted to create /ROOT/A. No two directories can have the same pathname.

77. USERNAME/PASSWORD MISMATCH.

User attempted to log on with a username that is valid but an incorrect password for this user.

78. USERNAME NOT KNOWN.

User attempted to log on with an unknown username.

79. FILE ERROR ON SYSTEM FILE.

The system accessed the user definition file. This file is a system file only.

80. NETWORK FILE DETACHED, DEVICE DISMOUNTED.

While a user was accessing a file, the device on which the file resided was dismounted or removed at the Network Resource Manager.

81. MAXIMUM REMOTE ATTACHES EXCEEDED.

User attempted to exceed the maximum number of files that could be attached at one time. The user can attach only 12 remote files at once.

82. ILLEGAL PASSWORD SYNTAX.

User attempted to log on with a password of more than 14 characters.

83. ILLEGAL USERNAME SYNTAX.

User attempted to log on with a username of more than 14 characters.

APPENDIX A
SUMMARY OF ERROR MESSAGES

This appendix provides a list of error codes and/or messages issued by ISIS-III(C), and some nonresident system routines. Other nonresident system routine error messages are listed in the *ISIS-II Users' Guide*.

Further explanations of the error codes can be found in other Intel manuals.

Error Codes	Manuals
1-39	*ISIS-II User's Guide*, 9800306 *Intellec® Series III Microcomputer Development System Console Operating Instructions*, 121609
40-60	*NDS-I Network Manager Operating Instructions*, 121645 *NDS-I Workstation Operating Instructions*, 121646
60-90	*NDS-II ISIS-III(N) User's Guide*, 121765
101-149	*Intellec® Series III Microcomputer Development System Console Operating Instructions*, 121609
201-240	*Intellec® Series III Microcomputer Development System Console Operating Instructions*, 121609

ISIS-III(C) Error Messages (8080/8085 Mode)

1. FATAL ERROR. TOO FEW BUFFERS WERE ALLOCATED.
2. ILLEGAL ACTIVE FILE TABLE NUMBER.
3. FATAL ERROR. ACTIVE FILE TABLE IS FULL.
4. INCORRECTLY SPECIFIED FILENAME.
5. UNRECOGNIZED DEVICE NAME.
6. ATTEMPT TO WRITE TO INPUT DEVICE.
7. FATAL ERROR. THE DISK IS FULL.
8. ATTEMPT TO READ FROM OUTPUT DEVICE.
9. DISK DIRECTORY IS FULL.
10. PATHNAME IS NOT ON SAME DISK.
11. FILE ALREADY EXISTS.
12. FILE IS ALREADY OPEN.
13. NO SUCH FILE.
14. WRITE-PROTECTED FILE ENCOUNTERED.
15. FATAL ERROR. ISIS OVERWRITE.
16. FATAL ERROR. BAD LOAD FORMAT.
17. NOT A DISK FILE.
18. ILLEGAL ISIS COMMANDS.
19. ATTEMPTED SEEK ON NON-DISK FILE.
20. ATTEMPTED BACK SEEK TOO FAR.
21. CANNOT RESCAN.
22. ILLEGAL ACCESS MODE TO OPEN.
23. MISSING FILENAME.
24. FATAL ERROR. DISK INPUT/OUTPUT HARDWARE ERROR.
25. ILLEGAL ECHO FILE.
26. ILLEGAL ATTRIBUTE IDENTIFIER.
27. ILLEGAL SEEK COMMAND.
28. MISSING EXTENSION.
29. FATAL ERROR. PREMATURE EOF.

30. FATAL ERROR. DRIVE NOT READY.
31. CANNOT SEEK ON WRITE ONLY FILE.
32. CANNOT DELETE OPEN FILE.
33. FATAL ERROR. ILLEGAL SYSTEM CALL PARAMETER.
34. FATAL ERROR. INVALID RETURN SWITCH IN A LOAD SYSTEM CALL.
35. SEEK PAST EOF.
61. DEVICE NOT ASSIGNED.
62. RESERVED.
63. SYNCHRONIZATION ERROR.
64. NETWORK COMM ERROR.
65. LOCAL COMM ERROR.
66. ILLEGAL ATTRIBUTE FOR REMOTE FILE.
70. NETWORK FILE ACCESS RIGHT VIOLATION.
71. ILLEGAL OPERATION ON SHARED FILE.
72. MAXIMUM NUMBER OF FILES ON A DEVICE EXCEEDED.
73. ATTEMPT TO DELETE A NON-EMPTY DIRECTORY.
74. ILLEGAL NETWORK PATHNAME SYNTAX.
75. NON-TERMINATING PATH ELEMENT IS NOT A DIRECTORY.
76. ATTEMPT TO CREATE A CONNECTED NETWORK FILE.
77. USERNAME/PASSWORD MISMATCH.
78. USERNAME NOT KNOWN.
79. FILE ERROR ON SYSTEM FILE.
80. NETWORK FILE DETACHED, DEVICE DISMOUNTED.
81. MAXIMUM REMOTE ATTACHES EXCEEDED.
82. ILLEGAL PASSWORD SYNTAX.
83. ILLEGAL USERNAME SYNTAX.

Console Command Interface Errors (8080/8085 Execution Mode)

201. Unrecognized switch
202. Unrecognized delimiter
203. Invalid syntax
206. Illegal disk label
208. Checksum error
209. Relo file sequence error
210. Insufficient memory
211. Record too long
212. Illegal relo type
213. Fixup bounds error
214. Illegal SUBMIT parameter
215. Argument too long
216. Too many parameters
217. Object record too short
218. Illegal record format
219. Phase error
220. No EOF record in object module file
221. Segment overflow during LINK operation
222. Unrecognized record in object module file
223. Fixup record pointer is incorrect
224. Illegal record sequence in object module file in LINK
225. Illegal module name specified
226. Module name exceeds 31 characters
227. Command syntax requires left parenthesis

```
228. Command syntax requires right parenthesis
229. Unrecognized control specified in command
230. Duplicate symbol found
231. File already exists
232. Unrecognized command
233. Command syntax requires a TO clause
234. Filename illegally duplicated in command
235. File specified in command is not a library
     file
236. More than 249 common segments in input files
237. Specified common segment not found in object
     file
238. Illegal stack content record in object file
239. No module header in input object file
240. Program exceeds 64K bytes
```

Table A-1. Nonfatal Error Numbers Returned by System Calls

System Call	Error Numbers
OPEN	3, 4, 5, 9, 12, 13, 14, 22, 23, 25, 28, 52, 54, 61, 70
READ	2, 8
WRITE	2, 6
SEEK	2, 19, 20, 27, 31, 35
RESCAN	2, 21
CLOSE	2
DELETE	4, 5, 13, 14, 17, 23, 28, 32, 61, 70
RENAME	4, 5, 10, 11, 13, 17, 23, 28, 52, 54, 56, 61, 70
ATTRIB	4, 5, 13, 23, 26, 28, 52, 54, 61, 66, 70
CONSOL	None; all errors are fatal.
WHOCON	None
ERROR	None
LOAD	3, 4, 5, 12, 13, 22, 23, 28, 34, 61, 70
EXIT	None
SPATH	4, 5, 23, 28, 61, 70
GETD	3, 4, 5, 13, 23
GETATT	4, 5, 13, 23, 28

Table A-2. Fatal Errors Issued by System Calls

System Call	Error Numbers
OPEN	1, 7, 24, 30, 33, 51, 63, 64, 65
READ	24, 30, 33, 63, 64, 65
WRITE	7, 24, 30, 33, 63, 64, 65
SEEK	7, 24, 30, 33, 63, 64, 65
RESCAN	33, 63, 64, 65
CLOSE	33, 63, 64, 65
DELETE	1, 24, 30, 33, 63, 64, 65
RENAME	1, 24, 30, 33, 51, 63, 64, 65
ATTRIB	1, 24, 30, 33, 51, 63, 64, 65
CONSOL	1, 4, 5, 12, 13, 14, 22, 23, 24, 28, 30, 33
WHOCON	33
ERROR	33
LOAD	1, 15, 16, 24, 30, 33, 63, 64, 65
SPATH	33
GETD	1, 24, 30, 33
GETATT	1, 24, 30, 33

APPENDIX B
SUMMARY OF ISIS-III(C) COMMAND SYNTAX

This appendix lists the ISIS-III(C) commands and syntax alphabetically. Chapter 4 of this manual contains complete descriptions and examples of each command.

ACCESS

```
ACCESS { :Fn:      } filename [switch] <cr>
       { pathname/ }
```

ASSIGN

```
ASSIGN [{ :Fn: }] [TO { network directory pathname,   }]  <cr>
       [{ n    }]     { another directory identifier, }
                      { directory identifier and      }
                      { pathname component,           }
                      { NULL                          }
```

ATTRIB

```
ATTRIB [:Fn:] filename [attriblist] [Q] <cr>
```

CANCEL

```
CANCEL queuename { (jobname)    } , . . . <cr>
                 { #(jobnumber) }
```

COPY

```
COPY [:Fn:] infile [, . . .] TO { [:Fn:] [outfile] } [switches] <cr>
                                { :device:          }
```

CREATE

```
CREATE { :Fn:      } new directory name <cr>
       { pathname/ }
```

DELETE

```
DELETE [:Fn:] filename [switch] <cr>
```

DIR

```
DIR [{ [FOR filename] }] [TO listfile] [switches] <cr>
    [{ /              }]
```

EXPORT

EXPORT { :Fn: / *pathname*/ } *filename* ['('*parameter list*')'] TO *queuename* { LOG / NOLOG } <cr>

LOGOFF

LOGOFF <cr>

LOGON

LOGON <cr>

QUEUE

QUEUE { ADD / DELETE } (*queuename* [,...]) <cr>

REMOVE

REMOVE { :Fn: / *pathname*/ } *filename* <cr>

RENAME

RENAME [:Fn:] *oldname* TO [:Fn:] *newname* <cr>

SPACE

SPACE /*volumename* <cr>

SUBMIT

SUBMIT [:Fn:] *filename* [(*parameter* [,...])] <cr>

SYSTAT

SYSTAT [{ QUEUE / MY JOB }] [*queuename*, *queuename*...] [*switch*] <cr>

VERS

VERS *command* <cr>

WHO

WHO <cr>

APPENDIX C
SUMMARY OF ISIS-III(C) DEVICES

This appendix alphabetically lists the devices that are accessible to the user at the ISIS Cluster workstation. Local devices are not accessible from the Cluster workstation.

Standard Devices

:BB: Byte Bucket (a universal null device) (input/output)
:CI: Current Console Device (input)
:CO: Current Console Device (output)
:F0: Directory (0) (input/output)*
:F1: Directory (1) (input/output)*
:F2: Directory (2) (input/output)*
:F3: Directory (3) (input/output)*
:F4: Directory (4) (input/output)*
:F5: Directory (5) (input/output)*
:F6: Directory (6) (input/output)*
:F7: Directory (7) (input/output)*
:F8: Directory (8) (input/output)*
:F9: Directory (9) (input/output)*
:HP: Not supported
:HR: Not supported
:LP: Not supported
:SP: Spool Line Printer Queue (network output)
:TI: Not supported
:TO: Not supported
:TP: Not supported
:TR: Not supported
:VI: Video Terminal Keyboard (input)
:VO: Video Terminal CRT Screen (output)

* indicates logical random access devices that are assigned to network directories.

#, 4-17
:F*n*:, 4-7
:SP:, 1-1, 4-2, 4-18
8080/8085 programs, 4-4

aborted job, 4-43
ACCESS, 3-4, 4-8
access
  devices, network, 1-6
  files, 1-6, 2-3, 4-11
  identifier, 4-8
  line printer queue, network, 1-1
  network, 1-1, 1-6, 4-3
  owner, 4-8
  rights, 2-2, 3-4, 4-8
  switches, 4-8
  world, 4-8
AEDIT, iii, 3-1
archiving remote files, 2-1
Assembly Language Calls, 5-2
ASSIGN, 1-6, 4-11
assignment
  console device, 5-1
  devices, 3-4
  directory identifiers, 3-4, 4-11
  directory listing, 4-11
ATTRIB call, 5-3
ATTRIB command, 4-15
attribute
  write-protect, 4-15

blocks, 4-38
boards
  Cluster, 1-1
  Communication Controller, 1-1
bootstrap, 1-7
branching
  file structures, 3-4
buffers
  SUBMIT, 4-40

cable, Ethernet, 1-1, 1-3
calls, system
  ATTRIB, 5-3
  CHGACS, 5-4
  DETIME, 5-6
  FILINF, 5-8
  GETATT, 5-11
  GETD, 5-13
  SPATH, 5-16
CANCEL, 4-17
cancelled job, 4-17, 4-43
CHGACS, 5-4
Cluster Board, 1-1, 1-3
Cluster configuration status, 1-7
Cluster workstation, 1-1, 1-2, 1-6
command
  ACCESS, 4-8
  ASSIGN, 4-11
  ATTRIB, 4-15
  CANCEL, 4-17
  categories, 4-2
  COPY, 4-18
  CREATE, 4-22
  DELETE, 4-24
  DIR, 4-26
  directory maintenance, 4-3
  entry, 1-8
  EXPORT, 4-28
  file maintenance, 4-3
  LOGOFF, 4-30
  LOGON, 4-31
  network access, 4-3
  program compatibility, 4-45, 6-1
  program execution, 4-4
  QUEUE, 4-33
  remote job execution, 4-4
  REMOVE, 4-34
  RENAME, 4-36
  SPACE, 4-38
  SUBMIT, 4-39
  syntax, 4-6
  SYSTAT, 4-42
  VERS, 4-45
  version, 4-45
  WHO, 4-46
Communication Controller Boards, 1-1, 1-3
Communications Module, 1-1, 1-3
compatible
  commands, 4-45, 6-1
configuration status, Cluster, 1-7
conventions, vi
COPY, 4-18
CREATE, 2-3, 4-22
creating
  directories, 2-3, 4-22
  job queues, 4-5, 4-33

data files, 1-5
  access, 1-6
  local, 1-5
  protection, 4-8
  remote, 1-5
DELETE, 4-24
DETIME, 5-6
device
  access, 1-6
  assignment, 3-4

logical, 2-1
physical, 2-1
diagnostic results, 1-7, 1-8
DIR, 4-26
directory, 1-3, 2-1
access rights, 4-11
creating, 2-3, 4-22
empty, 4-24, 4-34
files, 1-5
home, 3-3, 6-1
listing, 4-26
expanded, 4-26
maintenance, 3-1, 4-3
network, available, 4-11
parent, 4-34
remote, 1-5
structure, 2-1
hierarchical, 2-2
flat, 2-2, 4-1
directory identifier
assigning, 4-11
assignments,
automatic, 4-11
changing, 4-11
default values, 4-11
listing, 4-11
disk, 2-1
drive, 2-1
hard, 1-1, 4-2
distributed job control (DJC), 1-1 ,1-5

empty directory, 4-24, 4-34
error codes, 7-1
error messages, 7-1
Ethernet, 1-1, 1-3
executing job, 4-43
EXPORT, 4-28

files
access, 1-6, 2-3, 4-11
copying, 4-18
data, 1-5
deleting, 4-24
directory, 1-5, 3-1
length, 4-27
local, 1-5
maintenance, 4-3
name, 2-2
owner, 4-27
private, 1-5
protection, 2-2, 4-8
public, 1-5
remote, 1-5
renaming, 4-36
rights, access, 4-8
shared, 1-5
structure, 1-5, 2-1
system, 3-1
types, 1-5, 2-2

flat file structure, 2-2
FILINF, 5-8
flexible disk drive, 4-2

GETATT, 5-11
GETD, 5-13

hard disk, 1-1, 4-2
hardware, 1-3
hierarchical file structure, 1-5, 2-2
home directory, 3-3, 4-11, 6-1
host workstation, 1-1, 1-3

identifier
access, 4-8
directory
assigning, 4-11
changing, 4-11
default values, 4-11
listing, 4-11
data file, 4-8
directory file, 4-8
owner, 4-8
right, 4-8
iMDX 455, 1-3
iMDX 580, 1-1, 1-3
iMDX 582, 1-6
IMPORT, 1-1, 4-5
Import workstations, 4-42
iNDX, 1-3, 1-6
iNDX.GII, iii
input/output operations, 5-1
Intellink, 1-1, 1-2, 1-3
ISIS file structure, 1-5, 2-2, 4-1
ISIS-II, 4-1
ISIS-III(C), iii, 1-1, 1-6
ISIS-III(N), iii, 1-4, 1-6
ISIS.INI, 3-3, 4-31, 6-1
ISIS.SYS, 3-3, 4-11
IXREF, iii

job
aborted, 4-43
cancelled, 4-17, 4-43
creating, 4-5
done, 4-43
executing, 4-43
name, 4-17, 4-42
number, 4-17, 4-43
public workstation, 1-6, 4-4
queue, 1-6, 4-2, 4-33, 4-42
remote execution, 1-1, 1-5, 4-2
status, 4-43
waiting, 4-43
job queue
adding, 4-33
deleting, 4-33
queuenames, 4-33

librarian, 4-5
LINK, 5-2
linker, 4-5
literature, related, vi
local file, 1-5
local printer, 4-2
locator, 4-5
LOG, 4-27
LOG file, 4-28
logging off, 1-8
logging on, 1-7
logical device name, 2-1
logical system root, 2-2, 3-1
LOGOFF, 4-30
LOGON, 4-31

Mainframe Link, 6-1
maintenance
  directory, 4-3
  data file, 4-3
marker, 5-2
Module Communications, 1-1

name
  job, 4-17, 4-42
  queue, 4-33, 4-42
NDS-II file structure, 2-2, 3-1, 4-1
network, vi, 1-1
  access, 1-1, 1-6, 4-2
  capability, 1-1
  directory, 2-2
  line printer queue, 1-1, 4-2
  pathname, 4-11
  resources, 1-1
Network Resource Manager, v, 1-1, 1-5, 1-6
NOLOG, 4-28
Notational Conventions, vi
NRM
  see Network Resource Manager, v
NULL, 4-11
number
  job, 4-17, 4-43

operating system, differences, 4-1
overview, v
owner
  data file, 4-8
  directory file, 4-8
  identifier, 4-8

parameter, 4-6
parameter list, 4-28
parent directory, 4-34
password, 1-8, 3-3, 4-30
pathname, 2-2
  directory, 4-11
physical device name, 2-1
PL/M-80 system calls, 5-2
printer, remote, 1-1, 4-2
private files, 1-5
private workstation, 1-6, 4-4
program loading, 5-1
protection, file, 2-2, 4-8
publications related, vi
public files, 1-5
public workstation, 1-6, 4-4

query switch, 4-15
QUEUE, 4-5, 4-33
queue
  cancelling, 4-17
  creating, 4-5, 4-33
  import station, 4-42
  job, 4-42
  listing, 4-42
  name, 4-42
  naming, 4-5
  spooler, 4-2, 4-18
queuename, 4-33, 4-42
quotes, single, 2-1, 4-20

related publications, vi
remote
  files, 1-5
  jobs execution, 1-1, 1-5, 4-2
    aborted, 4-43
    cancelling, 4-17
    creating, 4-5
    executing, 4-43
    status, 4-43
  printer, 1-1, 4-2
REMOVE, 4-34
RENAME, 4-36
right identifier
  ACCESS, 4-8
rights
  access, 4-8
  changing, 4-8
  listing, 4-8
  owner, 4-8
  Superuser, 4-9
root
  logical system, 2-2, 3-1
RUN, iii

shared files, 1-5
software
  compatible, iii, 4-45
  preparation, 1-6
SPACE, 4-3, 4-38
SPATH, 5-16
spooled printer queue, 4-2, 4-18
standalone
  development system, 1-3
status, 4-42
SUBMIT, 4-39
summary of calls, 5-1

Superuser, 1-9, 2-2, 3-3
  access rights, 4-9
switch
  query, 4-15
syntax
  command, 4-6
  system call, 5-2
SYSTAT, 4-42
system calls
  summary, 5-1
  syntax, 5-2
  usage, 5-2
system files, 3-1, 4-12
SYSTEM.LIB, 5-2
system root
  logical, 2-2, 3-1
system routine, 5-2

terminology, 1-3

username, 1-8, 3-3, 4-30

VERS, 4-45
version
  command program, 4-45
volumes, 2-2, 3-1, 4-38

WHO, 4-46
wild card, 4-3, 4-19
workstation, 1-1
  Cluster, 1-1, 1-2, 1-6
  host, 1-1, 1-3
  import, 1-6, 4-4
  private, 1-6, 4-4
  public, 1-6, 4-4
  upgrade kit, 1-3
write-protect attribute, 4-15

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.