Intermec
Technologies Corporation

4985 Network Controller

PROGRAMMER'S GUIDE

PN: 977-028-006
Revision A
May 1996

" **NOTICE**

This publication contains information proprietary to Intermec Technologies Corporation. It is being supplied to you with the express understanding that the information contained herein is for the benefit of the contracting party only, and may not be copied, distributed, or displayed to third parties without the express written consent of Intermec Technologies Corporation, and shall be returned to Intermec Technologies Corporation upon written request. If a purchase, license, or nondisclosure agreement has been executed, the terms of that agreement shall govern this document.

This publication is furnished for information only, and the information in it is subject to change without notice. Although every effort has been made to provide complete and accurate information, Intermec Technologies Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Disclaimer of Warranties. The sample source code included in this document is presented for reference only. The code does not necessarily represent complete, tested programs. The code is provided **“AS IS WITH ALL FAULTS.” ALL WARRANTIES ARE EXPRESSLY DISCLAIMED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.**

We welcome your comments concerning this publication. Although every effort has been made to keep it free of errors, some may occur. When reporting a specific problem, please describe it briefly and include the book title and part number, as well as the paragraph or figure number and the page number.

Send your comments to:
Intermec Technologies Corporation
Publications Department
550 Second Street SE
Cedar Rapids, IA 52401

NORAND and PEN*KEY are registered trademarks of Intermec Technologies Corporation.

This publication printed on recycled paper.

Acknowledgments

MS-DOS is a registered trademark of Microsoft Corporation.

FCC Computer Compliance

" **NOTICE**

This equipment meets Class A digital device limits per Part 15 of FCC Rules. These limits protect against interference in a commercial area. It emits, uses, and can radiate radio frequency energy. If you do not install and use the equipment according to its instructions, it may interfere with radio signals. Using it in a residential area is likely to cause interference. If this occurs, you must correct the interference at your expense.

CONTENTS

SECTION 1

SECTION 2

SECTION 3

SECTION 4

SECTION 5

SECTION 6

SECTION 7

APPENDIX A

APPENDIX B

APPENDIX C

APPENDIX D

FIGURES

TABLES

Section 1

Introduction

The 4985 Network Communication Controller provides the communication link between a NORAND® Hand-Held Computers (HHCs) and your host computer. Two serial communication channels service the network. The host interface channel is an RS-232 interface, while the Local Area Network (LAN) is an RS-485 interface.

Once powered on, the 4985 Network Communication Controller requires little or no operator intervention. You can enter changes to the host interface portion using the dip switches (located at the back of the controller) or a host program that directs the internal *soft* changes. This document aids in the writing of the interface program for your host computer.

At the initial installation, configure the 4985 Network Controller for the network it should operate. There are three banks of dip switches for functional configuration of the controller. Each bank contains eight switches. The first bank defines the type of communications, the second defines the modem type connected to the network, and the third bank defines the controller's unique address in your overall network.

Norand supports NORAND modems. You can enable or disable the modem configuration by setting on or off the appropriate DIP switch.

The file maintenance system provides an HHC boot mechanism. Your host computer downloads a boot file list during the file processing stage of communication. After completing file processing, all of the files specified in the list can be downloaded to the HHC on the RS-485 network.

About the Controller

This controller requires little or no operator intervention during normal operation. Once the initial configuration is complete, controller operation becomes transparent to the user.

The power-up sequence verifies proper operation of the Read-Only Memory (ROM), Random Access Memory (RAM), and partial testing of the serial communication controller. Upon successful completion of the power-up self-test, the controller sounds a single beep. Passing the power-up self-test also implies proper operation of the internal Central Processing Unit (CPU) and bus interface logic. A double beep indicates the host port is ready to communicate.

Should the controller beep multiple times during power-up, the controller has failed a portion of the startup test sequence. Consult the beep counts table in Appendix A for descriptions of the error codes.

After successful power up, the controller is ready to communicate with the device attached to the host interface port. If the host port device is a modem, the controller may configure the modem and wait for the remote system to call.

You can access a debug facility for analysis of controller hardware and configuration.

Host Interfaces

You may need a host communication program and support programs to talk to the controller. Norand provides two host interfaces: Asynchronous and Secondary ADCCP.

The Secondary ADCCP interface only communicates with the NORAND 4980 Network Controller or the 4920 PC Communication Package and can communicate locally or remotely using modems.

The asynchronous interface connects the controller directly to your host computer. This connection provides a *gateway* for data to pass between the HHCs and your host using logical channels and record types.

This guide primarily covers the asynchronous interface which is 4980-compatible. The primary difference between the 4980 and 4985 is the 4985 Controller has only two ports and lacks a display, keypad, and disk drive. Since one port is dedicated to the host interface, there is only one port remaining for communicating with HHCs.

The different record types, sent back and forth between the controller and the host, help tailor the communications session to meet that specific session's needs.

Section 2

Getting Started

Network Configurations

The host port attaches to a host computer, a PC, or another controller. This connection must use one of two datalink protocols: ASYNC or Secondary ADCCP.

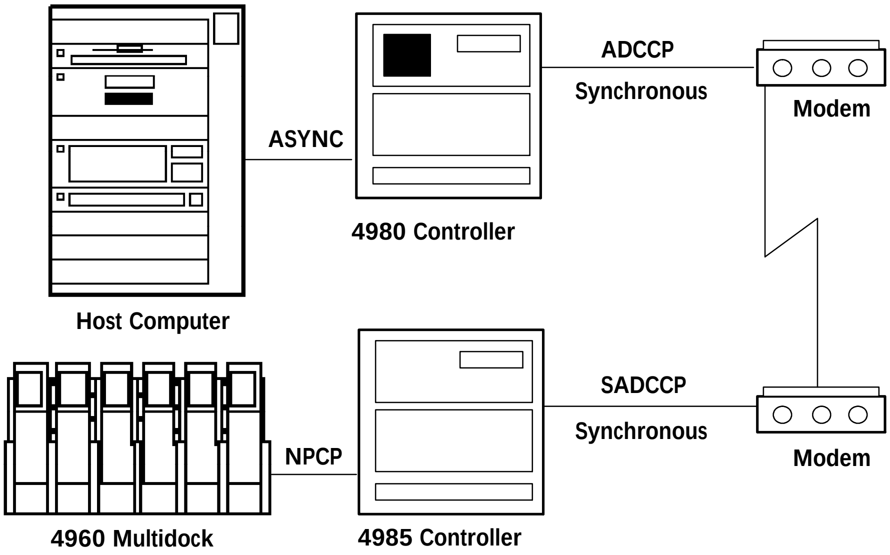


Figure 2-1
Host-to-4985 Using 4980

Figure 2-1 demonstrates the host computer directly connecting to a 4980 Network Controller, wired to a synchronous modem. A telephone line or lease links the two identical modems. Both modems must be configured exactly alike for smooth communications. The 4985 Network Controller connects to the modem with the host port.

Figure 2-2 shows the 4980 Network Controller on the left connected to a host computer. The 4985 Controller on the right is at a remote site, hooked up with at least one six-pack multidock.

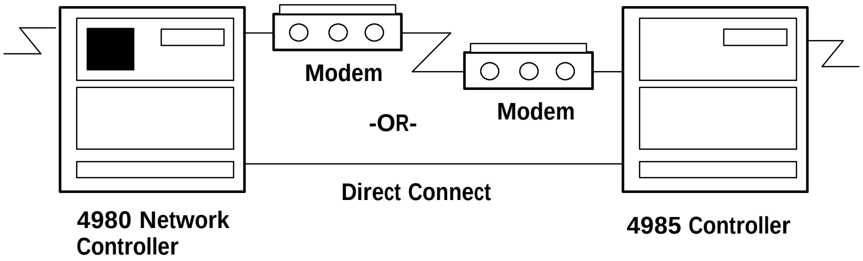


Figure 2-2
4980 Network Communications Controller Connection to 4985 Controller

Figure 2-3 shows a setup using the 4920 Communications Package for a PC connection to the 4985 Controller.

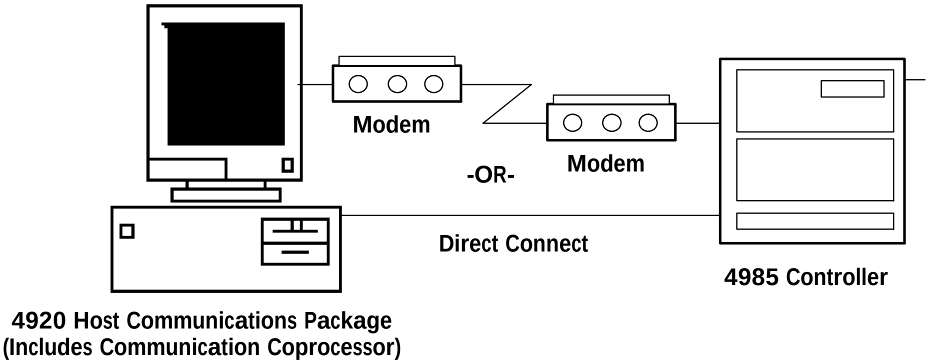

Modem
Modem
-OR-
Direct Connect
4985 Controller
4920 Host Communications Package (Includes Communication Coprocessor)

Figure 2-3
4920 Host Communications Package Connection to 4985 Controller

Figure 2-4 has the most Hand-Held Computers (HHCs) that can communicate through one 4985 Controller. The NORAND® Auxiliary Power Units (APUs) connect in series with the multidocks.

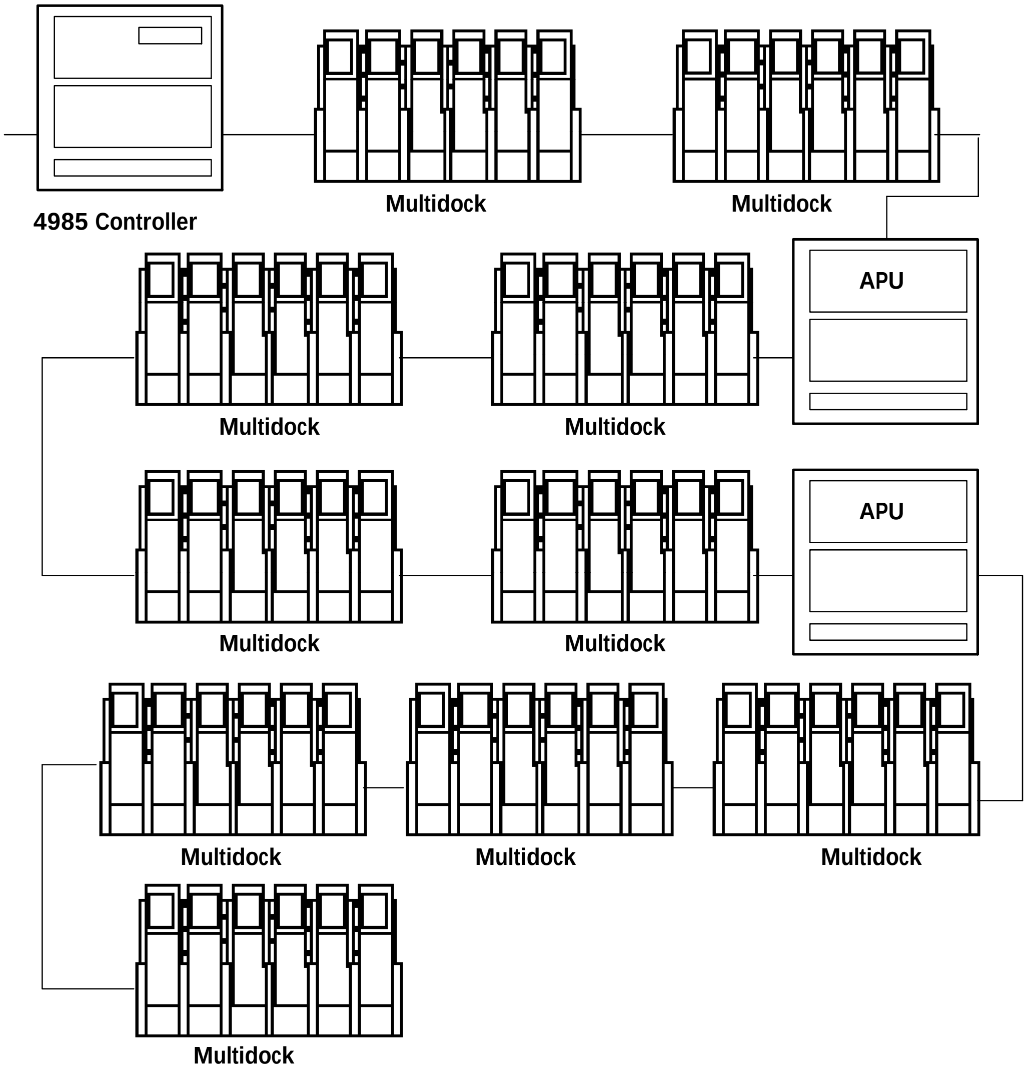


Figure 2-4
4985 Multidocks with Maximum Configuration

" **NOTE:** *Add APUs to the network when using more than two multidocks. This ensures adequate power for the multidocks to recharge the HHCs when placed in these docks. One APU powers four multidocks. The controller powers two multidocks directly.*

Figure 2-5 differs from Figure 2-4 by connecting the two APUs together instead of chaining the second APU in series to a multidock. The second APU connects directly to the first APU.

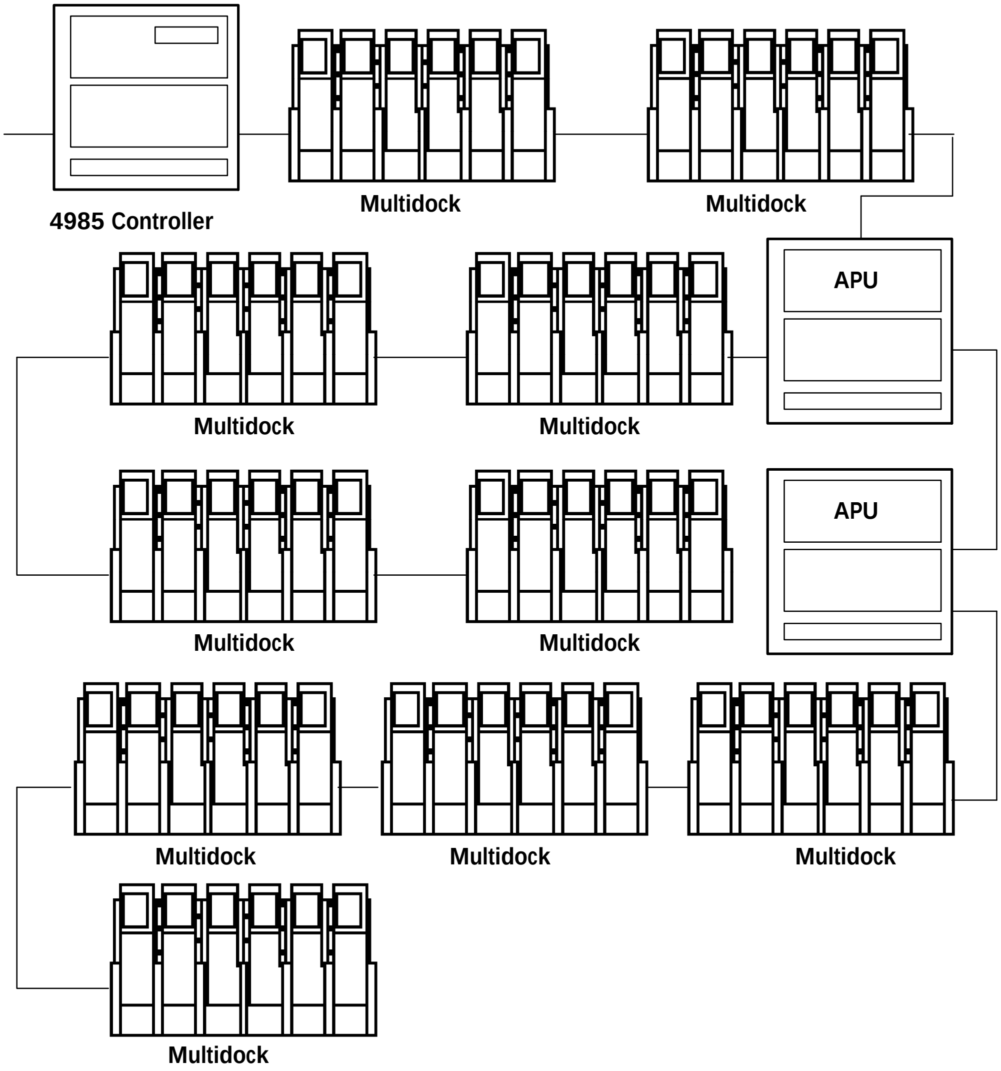


Figure 2-5
Optional Wiring with 2 APUs Connected

If hooking up the maximum number of multidocks allowed, use the wiring diagram in Figures 2-4 or 2-5. Take into consideration your power outlets and physical setup.

4985 Configuration Switches

Three 8-position DIP switches allows a user to customize the 4985 for their particular system application. Switches “1” through “5” of DIP bank #2 apply only to ADCCP host applications. DIP bank #3 configures the controllers identification number for a large system. The controller converts this 8-bit ID to an ASCII decimal number in the 49850000 to 49850255 range. This ID goes to the host computer at the start of the host-to-controller communication session. The default value is “49850000” and “49850255” is reserved.

" **NOTE:** *Details of the three Data Input (DI) bank switches start on page 3-1.*

Figure 2-6 shows switch settings for connecting the 4985 Network Controller to the 4980 Network Controller or the 4920 Communication Package using Secondary ADCCP and a NM9600 Modem.

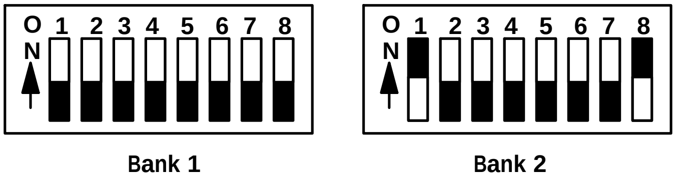


Figure 2-6
Default Switch Settings

Bank 1:	Switch 1:	Command mode is off
	Switch 2:	Communications mode uses Secondary ADCCP
	Switch 3-5:	Communications uses external clocking to determine speed
	Switch 6:	Reserved.
	Switch 7-8:	Sets controller to handle three channels active to HHCs concurrently.

" **NOTE:** *All reserved switches or those not applicable **must** be in the off position.*

Bank 2:	Switch 1:	Auto-configure the modem
	Switch 2-3:	Auto-dialing is off

Switch 4:	Auto-answer is off
Switch 5:	Controller sends AT&F command to modem before configuring.
Switch 6:	Reserved.
Switch 7-8:	Three logical channels enabled.

Communication Session

This describes the host-to-HHC communication session. Session examples illustrate what is happening, physically and logically.

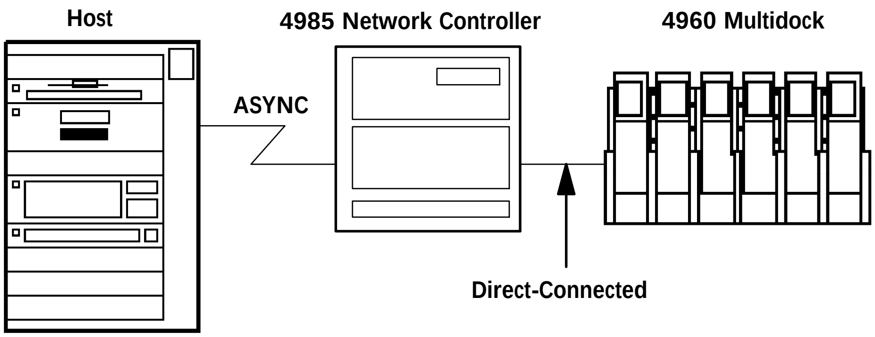


Figure 2-7
Typical Host-to-4985 Setup

The host communicates with the 4985 Controller using the NORAND asynchronous proprietary protocol. The multidock connects to the LAN port on the 4985 Controller and the host to the HOST port.

The primary controller use is to facilitate the host-to-HHC communication session. The controller provides the physical connection necessary for communication, and maintains the session on a logical channel. This allows you more options to expand your communications network.

A logical channel between an HHC and your host always passes through one or more controllers on a logical channel.

The logical channel ID prefixes all records sent from the controller to the host. It is a 1-byte field if you have disabled device channels and a 2-byte field if you have enabled device channels.

In this case, logical channels, specified by a single HHC channel ID, prefix each record sent from the controller.

Logical Channel	Record Type	Data

Figure 2-8
1-Byte Identifier

The 4985 Network Controller uses the 2-byte identifier option for compatibility. It consists of four parts.

Device Channel	Logical Channel	Record Type	Data

Figure 2-9
2-Byte Identifier

A device channel is zero as the 4985 is directly connected, the ASYNC interface is used, and an HHC channel will be “1”, “4”, or “7.” The device channel identifies which controller the HHC channel belongs to.

After making the physical connection, the communication session for all HHC types is identical. The HHCs’ protocol is transparent to the host computer.

The communication session for an HHC consists of:

- Upload phase
- Download phase
- End-of-Session status from the HHC

Typically, the HHC sends its ID in the first upload record. The host uses this information to tie a set of download data as specific HHC.

EXAMPLE: The host receives upload records from an HHC on a logical channel until the upload is complete. The host then receives download data requests on the same logical channel from the HHC. The host, using the HHC’s terminal ID, organizes the requested download data and sends the download data to the HHC on the logical channel until finished. At this point, the HHC reports the session status by sending an end-of-session record to the host.

Configuration information sent to the controller from the host is sent in the system initialization record at the start of communication.

The host can fine-tune the system and port configurations to meet the needs of a particular communications session.

Device Channels

Some things to remember:

- A logical channel number and record type prefix all controller data.
- HHC data is in a contiguous sequence on a logical channel.
- The host telecommunications program must maintain state variables and file pointers indexed by the logical channel number.

Help

The controller runs several self-diagnostic programs during power-up, ensuring the equipment is in normal working order. If a problem during power-up, the controller beeps several times and the unit does not operate.

The most common problems are often easiest to detect, and in many cases, correct. These include loose cable connections, loss of AC power, bad or no telecommunication, and sometimes faulty equipment. The HHC display sometimes helps to diagnose a problem.

Beeping

The controller performs a series of power-up self-tests each time it powers on. A single beep indicates that all power-up tests passed. Two beeps indicate the host port has reset and is ready to communicate. A long beep or a short pattern of beeps repeating for longer than 30 seconds indicate a problem with the internal hardware. The unit must go to a service center for repair. These beeps are diagnostic tools for the service technician. See Appendix A for a list of beep counts and their problem areas.

Connections

During the installation and startup testing, the system should have made all connections to the controller and its peripherals. If problems do occur, ensure that the system has fully seated connectors. Ensure that power connections are attached securely to wall outlets.

Testing

Diagnostics can run through a command interpreter on the RS-232 port using an attached ASCII terminal. These tests require a technician to access the highly technical test findings.

Modem Indicator

Modems frequently have LEDs to show events as they happen. If a LED should light but does not, the modem manual and this information can help determine where the trouble exists. The modem could be bad, or the connections to either the controller or to the telephone lines could be defective. The modem indicators, if any, can help isolate a telecommunication (TCOM) problem.

Cleaning

Periodic cleaning maintain the appearance and reliability of the controller. When cleaning, inspect for damage or wear. Clean the controller with a soft cloth dampened with a quality glass cleaner. *Do not use sol vent solutions.*

Specifications

Communication Protocol Support:

RS-232	Secondary ADCCP (9.6Kbps) or Host Async (38.4Kbps)
RS-485	NPCP (500Kbps)

Environmental:	Operating temperature:	32° to 122° Fahrenheit 0° to 50° Celsius
Memory:	512K Internal RAM 128K Internal ROM	
Microprocessor:	80C186 @ 8 MHz	
Physical:	Height:	9.5 inches (241 mm)
	Width:	13 inches (330 mm)
	Depth:	7.5 inches (191 mm)
	Weight:	5 lbs
	Chassis:	0.063 inch-thick (1.6 mm) #5052H32 aluminum
Power Capacity:	12 HHCs maximum	
Power Supply:	External:	+15 Volts @ 3.5 Amps
Serial Interfaces:	RS-232	1 port
	RS-485	1 port

Hardware

B *CAUTION:* ***Applying cleaners directly to any part of the controller will damage the controller. Do not use alcohol, benzene, carbon tetrachloride, or any other solvents to clean any portion of the controller. Use a soft, lint-free cloth moistened in a quality glass cleaner to clean the exterior of the controller.***

B *CAUTION:* ***Using the controller in an hostile environment that includes temperature extremes or where moisture can enter the unit will damage the controller. Operate the controller indoors, sheltered from precipitation.***

" NOTE: *Ensure that a properly wired, isolated-ground AC outlet powers the power supply for the controller. Properly installed isolated ground outlets and wiring produce a relatively ElectroMagnetic Interference (EMI) or* noise *free environment. This ensures proper operation of sensitive electronics such as computers and telecommunications devices.*

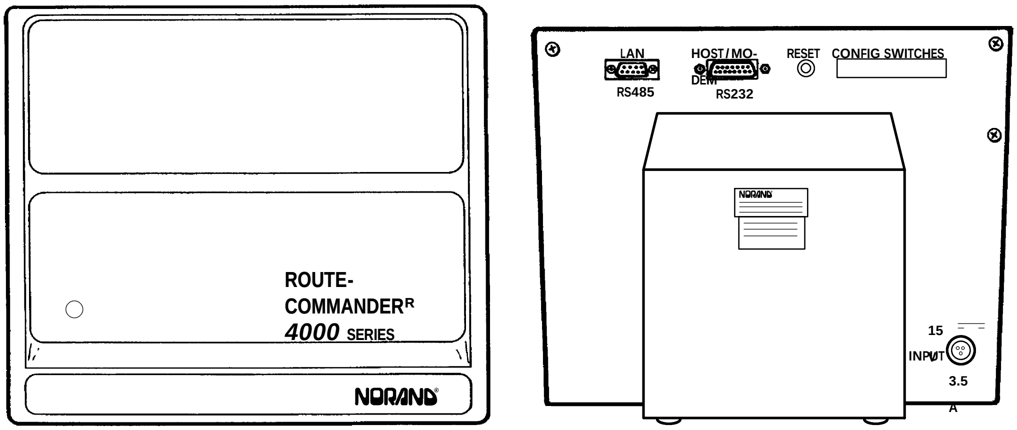
Unpacking Equipment

Inspect the shipping containers for signs of damage before unpacking the equipment. If any damage, you must immediately report that fact to the carrier.

Remove equipment and materials from the shipping containers and examine all items to become familiar with their physical characteristics. Check for signs of damage before you proceed.

Install mounting brackets in pairs, on 6-inch centers for the controller and the Auxiliary Power Unit (APU), and on 16-inch centers for the multidocks.

We do not provide mounting hardware. Use 1/4-inch hardware suitable for your wall or mounting surfaces (such as concrete block, poured concrete, drywall, plywood, sheet-metal, countertop, shelving, or a table).



Front

Back

Figure 2-10
4985 Network Controller

General Equipment

The controller interfaces between a group of HHCs and another controller, or a larger computer, such as a PC or mainframe. It is a low-cost controller for remote networks supporting 12 or fewer HHCs.

At each remote site, a controller connects to the HHCs by cables and a peripheral device, such as multidock or single dock. In turn, the controller connects to other network controller devices, a PC, modems, or a larger computer (host site) through cables or telephone link.

This provides flexible, software-controlled 2-way communication of data, messages, and program changes between a host computer and many HHCs. The controller acts as a buffer (temporary storage) and control device to ensure orderly communication and minimal errors.

The controller provides communication power and charging voltage for up to twelve 4000 Series HHCs. Add an APU to expand the network beyond the power supply capabilities of the controller. Connect up to two APUs and ten multidocks in series to one 4985 Network Controller.

Optional Peripherals

Optional peripheral devices allow for flexibility and customization:

Single Dock

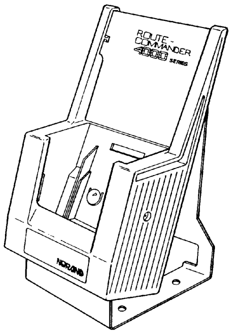


Figure 2-11
4950 Single Dock

The 4950 Single Dock accommodates one HHC. Use the single dock under specific restrictions. Special consideration for system power is required. See your Norand Systems Engineer for details.

Multidock

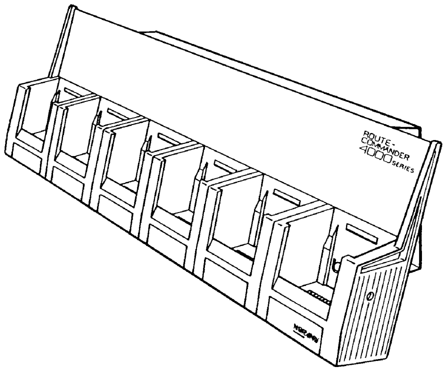


Figure 2-12
4960 Multidock

Chain the 4960 multidocks together with cables to increase the number of HHCs connected to each controller. Each rack has six docks for the 4000 Series HHCs.

Auxiliary Power Unit (APU)

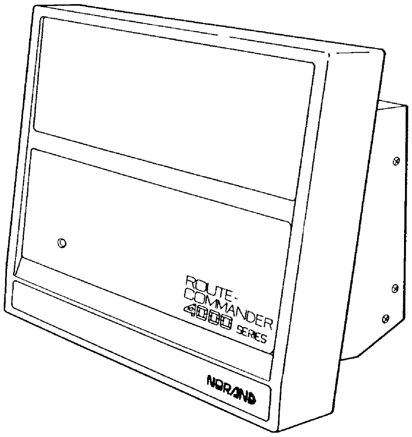
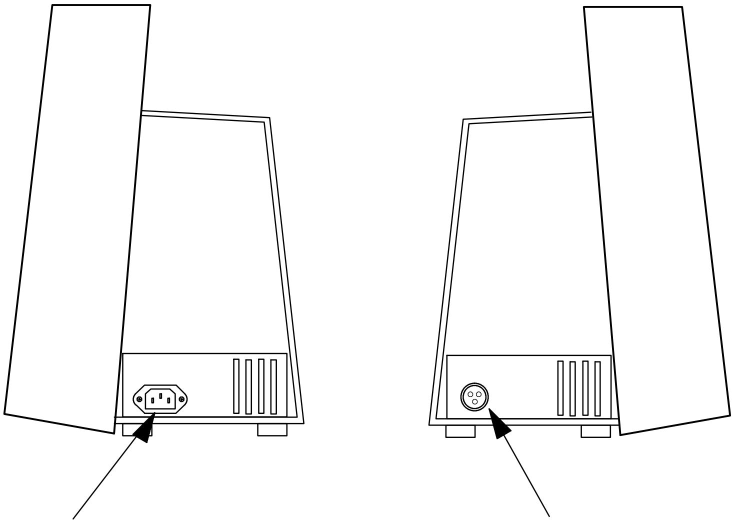


Figure 2-13
Auxiliary Power Unit

Place an APU in the network after the first two multidocks, before the next group of four multidocks. The APU furnishes charging current to the next group of multidocks and also strengthens signals on the communication lines to ensure reliable, error-free data transmission in both directions. There can be a maximum of two APUs and ten multidocks.

Power Supply



Insert your power cord from standard outlet plugs here.

Connect the network controller to this power supply outlet.

Figure 2-14
Power Supply Placement and Connectors

Place the power supply on the attached mounting shelf located at the back of the controller.

Modems

Limit the modem configuration to support these two modems: NORAND NM2400A and NORAND 9600.

Enter remote auto-dial phone numbers into the NORAND modems. Use the NORAND 9600 Modem front panel to enter the numbers. Use an RS-232 terminal to enter the phone numbers into an NM2400A. Refer to the operation documents for your modem before entering phone numbers into the modem.

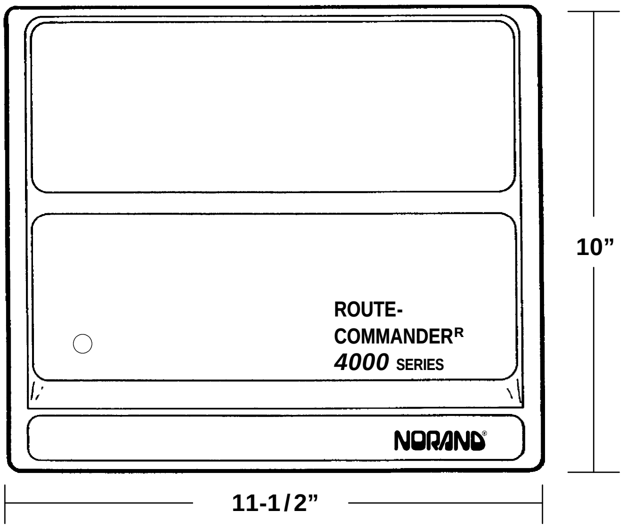


Figure 2-15
Approximate Dimensions of Auxiliary Power Unit

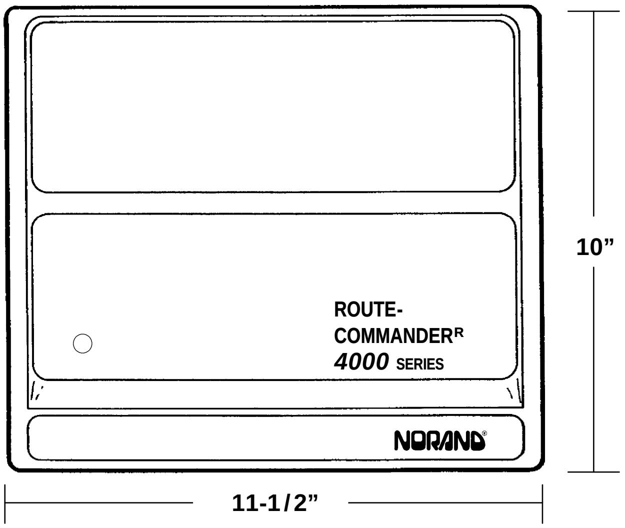
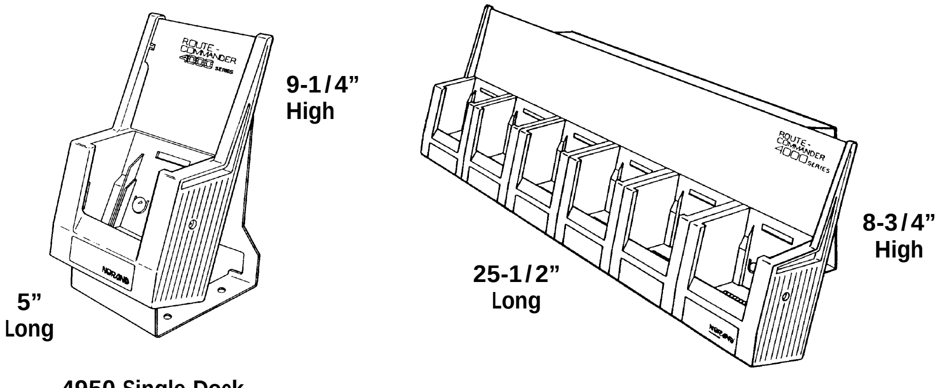


Figure 2-16
Approximate Dimensions of 4985 Network Controller

Location

Avoid extreme temperatures or humidity. Plan your space carefully before you install the mounting brackets. Figures 2-15 through 2-17 show approximate dimensions of the controller, the APU, the multidock, and the single dock.



4950 Single Dock

4960 Multidock

Figure 2-17
Approximate Dimensions of Docks

Use graph paper to draw the installation to scale in your planning.

- Minimum required spacing between a 4985 Network Controller and a multidock is 19 and 3/4 inches. Plan for 39 and 1/4 inches of space.
- Minimum required spacing between two multidocks is 26 and 1/2 inches. Plan for 51 and 1/2 inches of space.

Locate the controllers and APUs within reach of properly grounded, 3-prong AC outlets.

" **NOTE:** *Do not use extension cords with this equipment.*

Brackets

Install the system once you determine the location and plan the layout. Figure 2-18 illustrates a wall layout using a pencil, a carpenter's rule, a chalk line, and a level. After installing each pair of brackets, mount the equipment item to ensure clearances between adjacent pieces.

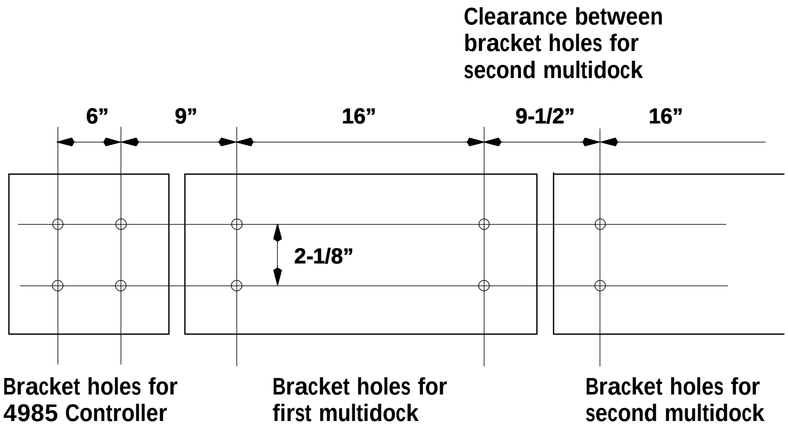

Clearance between bracket holes for second multidock
6”
9”
16”
9-1/2”
16”
2-1/8”
Bracket holes for 4985 Controller
Bracket holes for first multidock
Bracket holes for second multidock

Figure 2-18
Sample Wall Layout with Brackets

Install controller and APU brackets on 6-inch centers, using hardware appropriate to the mounting surfaces. Mount multidock brackets on 16-inch centers. You can mount these brackets on a vertical surface or wall (Figure 2-19).

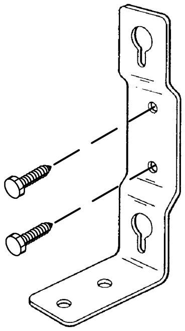


Figure 2-19
Vertical or Wall-Mounted Bracket

These brackets have two holes on the lower arm for mounting on to a horizontal surface, such as a countertop or table. When mounting on a horizontal surface, use at least an 8-inch shelf depth (Figure 2-20).

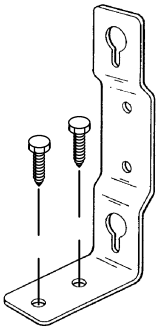
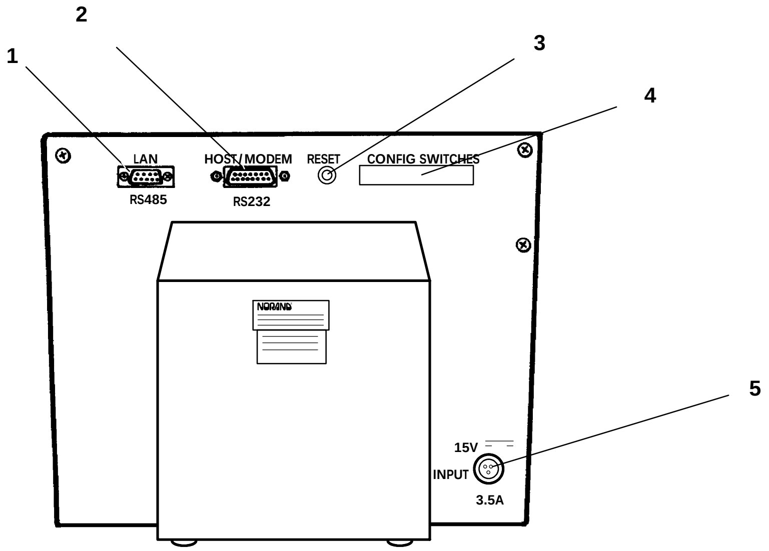


Figure 2-20
Horizontal or Shelf-Mounted Brackets

The controller, APU, and multidock all have pairs of large rivets welded to the back frame or cover. These rivets go in the mounting bracket keyholes, then slide down to lock the equipment securely in place.

Connectors

Multidock connectors (on each end) are identical and interchangeable. There is not an input or an output connector. Use the most convenient connector available on the multidock when connecting it to the controller, an APU, or when chaining to another multidock.



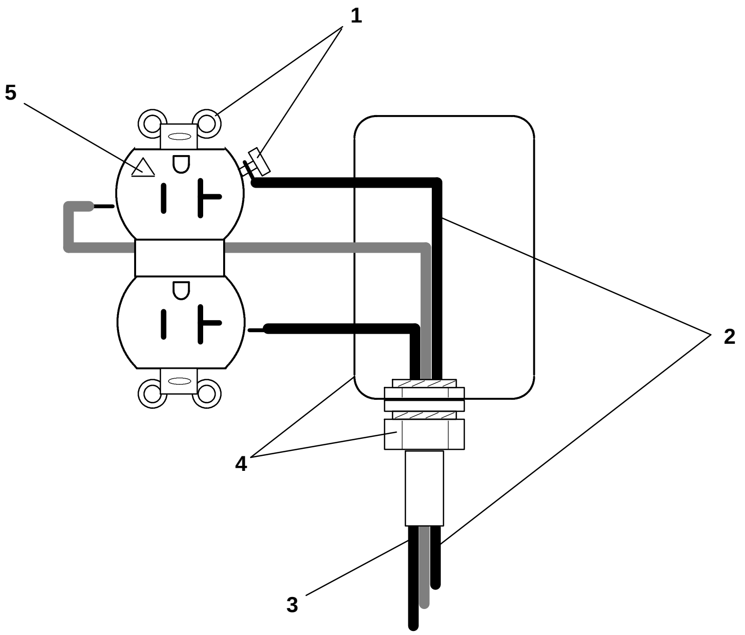
1. Multidock
2. Host or modem
3. Reset switch
4. Configuration switch
5. Power connector

Figure 2-21
Connectors on the Back of a 4985 Network Controller

" **NOTE:** *You can connect up to two multidocks. Add another APU if you want another multidock. You need special adapters for a single dock.*

AC Power

Use alternating current (AC) power for data communication equipment with *isolated ground* outlets. Properly installed isolated ground outlets and wiring produce a relatively ElectroMagnetic Interference (EMI) or *noise* free environment. This ensures proper operation of sensitive electronics such as computers and telecommunications devices.



1. Insulated from each other by design
2. Insulated ground wire
3. “Neutral” wire (white, insulated)
4. Conduit, box, metal wallplate (provides “Safety Ground”)
5. Orange triangle

Figure 2-22
Isolated Ground Outlets

An Isolated Ground (IG) installation requires special outlets such as General Electric catalog number 8200-IG2 (15 amp) or 8300-IG2 (20 amp). Wire these per applicable standards, such as National Electrical Code paragraph 250-74, (exception #4). A separate, insulated ground wire must be run, uninterrupted, from the outlets and terminate directly at an equipment grounding conductor terminal of the derived system or service.

Section 3

Installation

Start Up Operations

Start-up normally takes less than one minute.

At the time of initial installation, configure the controller for the system in which it operates. Three banks of Data Input (DI) switches are for functional configuration of the controller. Each bank contains eight switches.

- **Bank 1:** Defines host port communication type: ASYNC or Secondary ADCCP. If the host type is ASYNC, the host port parameters are: parity, data bits, speed, and channel ID.
- **Bank 2:** Functions include modem configuration, auto-answer, auto-dial, and a number of HHCs allowed to transmit concurrently.
- **Bank 3:** Defines identity code of the controller in a system configuration. Controller identification consists of a 3-digit number between “000” and “254.” There are 256 unique identity codes. The default ID is zero.

NOTE: *ID number 255 is reserved.*

Table 3-1
Bank 1 Dip Switches

Dip Switch	Meaning	Description
1	Command mode	**off:** Normal host communications. **on:** Controller powers up in command mode and awaits input commands from user on the RS-232 port. Provided as a diagnostic tool.
2	Host type	Specifies which host type is active. **off:** Secondary ADCCP **on:** Asynchronous host
3-5	Speed (bps)	ADCCP uses external clocking. 3 4 5 speed **off off off** : 19200 bps *(default)* **on off off** : 1200bps **off on off** : 2400bps **on on off** : 4800bps **off off on** : 9600bps **on off on** : 19200bps **off on on** : reserved **on on on** : reserved
6	Parity	ASYNC host only **off:** No parity **on:** Even parity
7	Databits	ASYNC host only **off:** 8 data bits **on:** 7 data bits
8		Reserved

Table 3-2
Bank 2 Dip Switches

Dip Switch	Meaning	Description
1	Modem configure flag	(Secondary ADCCP host only) **off:** Controller auto-senses the modem type before configuring the attached NM2400A or NORAND® 9600 Modem. **on:** Attached modem is not configured.
2-3	Autodial select *(see Note 1 after table)*	(Secondary ADCCP host only) Specifies which number stored in the attached modem to dial when a terminal is ready to communicate. **off off**: Autodial off **on off:** Dial stored number 1 **off on:** Dial stored number 2 **on on:** Alternate between stored number 1 or 2. The controller switches to the other number stored in the modem whenever a dial operation fails.
4	Force autoanswer *(see Note 2 after table)*	(Secondary ADCCP host only) **off:** Controller answers the phone only when a HHC is ready to communicate. **on:** Controller always answers the phone.
5	Skip modem reset	**off:** Controller sends an AT&F command to modem before modem is configured. **on:** Controller does not reset the NORAND Modem to factory default settings.
6	Reserved	
7-8	Number of logical channels on the LAN port. *(see Note 3 after table)*	This number equals the number of HHCs that can be active concurrently. **off off:** 1 channel active **on off:** 2 channels active concurrently **off on:** 3 channels active concurrently **on on:** Reserved

" **NOTE 1:** *If enabled (autodial switches 2 and 3) and the Modem Configure Flag (switch 1) is "on", the controller uses the Hayes command string "ATE0V0" to configure the modem and "ATDS" or "ATDS2" to dial.*

Also, if enabled, the controller will attempt to dial the host at power-up. This is useful for loading boot files into the controller during the initial power-up.

" **NOTE 2:** *If you wish the host computer to dial the controller to load boot files, you may perform the following steps:*
1. Set Force autoanswer to "on."
2. Press the reset button to purge files stored in the controller.
3. Download the boot files to the controller.
4. When finished, set Force autoanswer to "off."
5. Power-off the controller for 30 seconds (resets controller to original configuration).

" **NOTE 3:** *If you change the switch settings after the controller is powered on, you must either power off the controller for 30 seconds and power it back on, or press the reset switch. This action reconfigures the controller according to the new switch settings.*

" **NOTE:** *Pressing the reset button will erase all boot and other files from the controller's memory.*

Table 3-3
Bank 3 Dip Switches

Dip Switch	Meaning	Description
1-8	Controller ID	Default ID is 49850000.

These switches set an 8-bit number from hex "00" to "FF." The 8-bit number translates to an ASCII decimal number sent to the host at the beginning of a host-to-controller session.

The host may use the number to uniquely identify the controller. The host needs the ID to maintain the HHC boot file on the controller.

The ID sent to the host will range from "49850000" to "49850254," where the last three digits come from the switch settings and the first five digits are the device type.

Figure 3-1 shows the switch settings for an ID equal to “0000 0000.” The following is another way of representing the ID setting of zero: “off” “off” “off” “off” “off” “off” “off” “off.” The notation “off” and “on” represent the individual switch settings. Zero represents “off” and “1” represents “on” (such as 0000 0000).

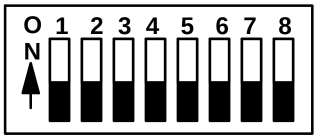


Figure 3-1
Default Switch Settings

Reset Button

■ **NOTE:** *You should always request permission from your data processing department before pressing the reset button. Pressing this button erases the files needed to “boot” the HHCs.*

File System

The controller supports a nonvolatile file system. This file system stores 4000 Series HHC boot files, program files, and data files.

Files downloaded to the controller are backed up for a minimum of one hour after the controller loses power. After that time, they could corrupt. Press the reset button before you download the files to the controller.

Download files from a host computer. For specific instructions on maintaining files with a host program application refer to the related publications listed below:

4980 Programmer’s Guide	NPN: 977-028-001
4920 and 4921 User’s Guide	
Volume A	NPN: 961-021-011
Volume B	NPN: 961-021-012
4920 and 4921 Reference Guide	NPN: 961-021-013

Section 4

File Structures

Introduction

Many of the configuration switches can be reconfigured by the host computer. These are with:

- Initialization record parameters. The host sends this record at the beginning of a communication session.
- Activation record parameters. The host sends this record to activate the LAN port.

Records from controller-to-host use a two-byte channel ID (when enabled) consisting of a one-byte *device channel* field. HHC channels belong to the active device on the corresponding device channel.

The 4985 asynchronous interface works with device channel “0”. The Local Area Network (LAN) port works with HHC channels “1”, “4”, and “7”.

Record formats with the two-byte channel option are the same as one-byte channels. The difference is that the device channel ID prepends to each record.

" **NOTE:** *Appendix D contains a quick reference to all of the following records.*

Host-to-Controller Records

Special Commands

The host may send two special commands to the controller. Records from the host, except for the special commands, must all be of the same fixed-length, and may include end-of-line pad bytes.

Type ? - Host Ready Command

The host sends this record to the controller when ready to receive data.

Table 4-1
Type ? - Host Ready Command

Position	Bytes	Description
1	1	Question mark “?”

EXAMPLE: (CR = end of record marker)
?<CR>

Type * - Reinitialization Command

The host sends this record to reset the controller when the controller is expecting a ready command.

Table 4-2
*Type * - Reinitialization Command*

Position	Bytes	Description
1	1	Asterisk “*”

EXAMPLE: (CR = end of record marker)
*<CR>

Each record sent from your host must have a record type field. There may be eight record types sent from your host to the controller. The controller may translate between ASCII and EBCDIC for EBCDIC hosts. You can enable the translation of the records using the host-to-controller initialization record. The eight record types are:

- “0” = Download Data
- “1” = End-of-Data
- “2” = Initialize
- “3” = Activate Auto-answer
- “4” = Activate Auto-dial
- “5” = Deactivate for one minute
- “6” = Host Directive
- “7” = Reboot Controller Directive

Record Layouts

This explains the function of each record type. Actual record layouts appear after the record description.

Each record from an ASYNC host must begin with a one-byte beginning-of-record delimiter (a slash "/") and a one-byte command code (ASCII) "0" through "7".

The host sends records in response to requests from the controller, except for the initialization record. The initialization record is sent to activate a controller that is inactive yet ready for activation.

The host may also send two special commands to the controller, a question mark "?" and an asterisk "*". A question mark indicates the host is ready to receive data and serves as an acknowledgement record. The asterisk resets the controller, although the controller will reset itself if the host issues an invalid record or fails to respond within the host timeout period.

The record includes an end-of-record delimiter, by default <CR> , which is configured in the initialization record.

Records from the host (not including the special commands) must all be of the same fixed-length, and may include end-of-line pad bytes.

The controller uses the initialization record sent from the host to determine these lengths:

- Record size
- End-of-line pad

Type 0 - Download Data Record

The host sends a download data record in response to a download request. Information obtained from upload data links download data with a specific HHC on a logical channel.

Table 4-3
Type 0 - Download Data Record

Position	**Bytes**	**Description**
1	1	Beginning delimiter: "/"
2	1	Command code: "0"
3	256	Download data
259	2	Optional CRC bytes (if enabled)
261+		Optional pad (up to 3 bytes)

EXAMPLE:

(CR = end of record marker)

```
/0xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<CRC><CRC>ppp<CR>
```

Beginning delimiter = “/”
Command code = Download Data “0”
Download data = “xxx”
Optional CRC bytes = 2
Optional pad = “ppp”
Optional end-of-record delimiter = <CR>

Type 1 - End-of-Data Record

The host sends an end-of-data record in response to a download request from the controller. This indicates that no more download data exists for this session or file. The host also sends end-of-data records in response to a file request to terminate file processing.

Table 4-4
Type 1 - End-of-Data Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “1”
3	256	Download data with required pad bytes if less than 256 bytes of data.
259+		Optional pad (up to 5 bytes)

EXAMPLE:

(CR = end of record marker)

```
/1xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxppppp<CR>
```

Beginning delimiter = "/"
Command code = End-of-Data "1"
Download data = "xxx"
Optional pad = "ppppp"
Optional end-of-record delimiter = <CR>

Type 2 - Initialization Record

At the beginning of a host-to-controller session, an initialization record sets the system mode parameters and default port activation parameters. The initialization record parameters are defined on page 4-20.

Table 4-5
Type 2 - Initialization Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "2"
3	256	Initialization parameters list. The parameter list terminates with a slash "/."
259+		Optional pad (up to 5 bytes)

EXAMPLE:

(CR = end of record marker)

```
/2xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/ppppp<CR>
```

Beginning delimiter = "/"
Command code = Initialization Record "2"
Parameter initialization list = "xxx"
Optional pad = "ppppp"
Optional end-of-record delimiter = <CR>

Type 3 - Activate (Auto-Answer) Record

The host sends an activate record in response to an activate request record. Nonswitched (direct) connections are auto-answer connections. See page 4-26 to build your activation parameters list.

Table 4-6
Type 3 - Activate (Auto-Answer) Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "3"
3	1	ASCII blank " "
4	255	Optional activation parameters list. This list terminates with a slash "/" and the remaining bytes filled with blanks.
259+		Optional pad up to 5 bytes

EXAMPLE:

(CR = end of record marker)

```
/3 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/ppppp<CR>
```

Beginning delimiter = "/"
Command code = Activate Record "3"
ASCII blank = " "
Activation parameters list = "xxxx"
Optional pad = "ppppp"
Optional end-of-record delimiter = <CR>

Type 4 - Activate (Auto-Dial) Record

The host responds to an activate request record with an activate record. Nonswitched (direct) connections are auto-answer connections. The controller processes auto-dial activate records for connections that are dial-out. The activation parameters lists are derived on page 4-26.

NOTE: *This is used on the 4980 Controller and is presented here for compatibility. The 4985 uses autoanswer records for NPCP.*

Table 4-7
Type 4 - Activate (Auto-Dial) Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “4”
3	1	ASCII blank “ ”
4	255	Optional activation parameters list. This list terminates with a slash “/”. When used, this field should be filled with blanks.
259+		Optional pad up to 5 bytes

EXAMPLE: (CR = end of record marker)

```
/4 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/ppppp<CR>
```

Beginning delimiter = “/”
Command code = Auto Dial Record “4”
ASCII blank = “ ”
Activation parameters list = “xxxx”
Optional pad = “ppppp”
Optional end-of-record delimiter = <CR>

Type 5 - Deactivate for One Minute Record

Use deactivate records to postpone activating the port for one minute. The controller will issue another activation request after one minute.

Table 4-8
Type 5 - Deactivate for One Minute Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “5”
3	256	Required pad bytes
259+		Optional pad up to 5 bytes

EXAMPLE:

(CR = end of record marker)

```
/5xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxppppp<CR>
```

Beginning delimiter = “/”
Command code = 5
Filler = “xxxx”
Optional pad = “ppppp”
Optional end-of-record = <CR>

Type 6 - Host Directive Record

The host sends a directive in response to a special download data request from the controller. Five subtypes of host directives are available. Types “1”, “2”, and “3” manage user data files. Type “4” sends download data to an HHC and type “6” cancels file processing.

Table 4-9
Type 6 - Host Directive Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "1" - "4" and "6": "1" = File upload directive "2" = File create or load directive "3" = File delete directive "4" = File download directive "6" = File processing abort
4	256	Directive information. The information required for this field depends on the directive type used.
259+		Optional pad up to 5 bytes

Type 7 - Reboot Controller Directive

This directive may be sent in response to any controller-to-host record and will force the controller to reboot.

Table 4-10
Type 7 - Reboot Controller Directive

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "7"
3	1	Reboot state "0" = Do not clear files from RAM disk "1" = Clear files from RAM disk
4	256	Filler bytes

Directive Types

If file processing is enabled using the system mode parameter in the initialization record, the host will receive file requests from the controller at the beginning of a host-to-controller session and after the host acknowledges a controller Directive Status Record. These requests can have one of several host directives. These four file directives for file maintenance are:

- Upload
- Delete
- Create or load
- Processing abort

" **NOTE:** *The controller sends a directive status to the host immediately after completing a host directive.*

File Upload Directive "1"

The host sends an upload directive, in response to a file processing special request, to upload a file stored on the 4985 controller (usually the directory file).

Table 4-11
File Upload Directive "1"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "5"
3	1	Directive Type: "1"
4	8	DOS compatible filename, left-justified
12	3	DOS compatible filename extension, left-justified
15	245	Filler bytes (ASCII space)

EXAMPLE:

```
/51FILENAMEEXT. . .xxxxxppppp<CR>
```

File Create or Load Directive "2"

The host sends this directive, in response to a file processing special request, to create a data file on the controller. After the controller receives a create directive, it sends the host data request records. The host then sends data records until all the data is sent. An end-of-data record from the host marks the end of the data for the file.

All directory entries use the following format:

Table 4-12
File Create or Load Directive "2"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "2"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	6	Date field (YYMMDD)
21	4	24-hour time (HHMM)
25	2	User-defined filler
27	1	User-defined file type "0" or blank indicates an HHC boot file.
28	1	Filler *(should be zero filled 0x30)*
29	1	File status: "0" = Good "1" = Undefined "2" = Feleted "3" = System
30	6	File size (in ASCII digits) *required*
36		Filler bytes

EXAMPLE: /62FILENAMEEXTddddddttttff000123456ppppp<CR>

File Delete Directive "3"

The host sends this directive, in response to a file processing request, to delete a data file from the controller's directory.

Table 4-13
File Delete Directive "3"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "3"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	245	Filler bytes

EXAMPLE: /63FILENAMEEXT. . .xxxxxppppp<CR>

File Download Directive "4"

The host sends this directive in response to a request for download data from an HHC. After receiving a download directive, the controller opens the file specified and downloads it to the HHC. The controller must contain this file (created with File Create).

Table 4-14
File Download Directive "4"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "4"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	245	Filler bytes

EXAMPLE: /64FILENAMEEXT. . .xxxxxppppp<CR>

File Processing Abort "6"

The host sends this directive to inform the controller that a file error has occurred on the host. *The controller responds by locking the files on the controller.* Normal telecommunication begins with the HHCs at this time. *No HHC cold starts are possible in a locked file system*. The files unlock after a successful file processing session.

Table 4-15
File Processing Abort "6"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "6"
4	256	Filler bytes

EXAMPLE: /66xxxx . . .xxxxxppppp<CR>

Controller-to-Host Record Layouts

Controller-to-host record types implement the file transfer protocol and status information between the controller and the host. The record type functions and a sample record layout are given here. The 1-byte channel ID supports older host programs. Activate the 2-byte channel ID in your host program. The 4921 PC Telecommunications Packages uses 2-byte channel ID, which defines the records that follow.

There may be nine status and request record types sent from the controller to your host that require a response from your host.

- "0" = Upload Data
- "2" = Data Request
- "4" = Active Status
- "6" = Special Request
- "8" = Identification
- "1" = End of Session Status
- "3" = Inactive Status
- "5" = Activate Request
- "7" = Directive Status

Type 0 - Upload Data Record

The HHC sends upload data records contiguously to the host on a logical channel.

Table 4-16
Type 0 - Upload Data Record

Position	Bytes	Description
1	1	Device channel ID
2	1	HHC channel ID: “1”, “4”, or “7”
3	1	Record type: “0”
4	256	Upload data
260	2	CRC bytes if CRC checking is enabled
262	3	Optional padding bytes

EXAMPLE: (CR = end of record marker)

```
010xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<CRC><CRC><CR>
```

Device channel ID = 0
HHC channel ID = 1
Record type = upload data
“x’s” are uploaded data
<CRC><CRC> = 2 CRC bytes for upload data

▪ **NOTE:** *CRC implementation is detailed in Appendix C.*

Type 1 - End of Session Status Record

This record indicates the status of a single HHC session. Each HHC sends an end-of-session record when finished communicating.

Table 4-17

Type 1 - End of Session Status Record

Position	Bytes	Description
1	1	Device channel ID: “0”
2	1	Channel ID: “1”, “4”, or “7”
3	1	Record type: “1”
4	1	End of session status code: “0” = Good status “1” = Abort. Port connection lost. “4” = Disconnect. The session disconnected, probably due to a datalink protocol error. “8” = HHC logic error. The HHC detected a logic error in the file transfer process.
5	4	HHC application status information
9	1	Local port identifier
10	249	Reserved for diagnostic information

EXAMPLE:

(CR = end of record marker)

```
0110G0001xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx<CR>
```

Device channel ID = 0
HHC channel ID = 1
Record type = end of session
End of session is good.
Local port identifier is 1
Optional diagnostic data is indicated by “x’s”.

Type 2 - Data Request Record

Table 4-18
Type 2 - Data Request Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “2”

EXAMPLE: (CR = end of record marker)
012<CR>

Device channel ID = 0
HHC channel ID = 1
Record type = data request

Type 3 - Inactive Status Record

Table 4-19
Type 3 - Inactive Status Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “3”

EXAMPLE: (CR = end of record marker)
013<CR>

Device channel ID = 0
HHC channel ID = 1
Record type = inactive status

Type 4 - Active Status Record

Active status records indicate a logical port status. The host port uses active status records as time-fill records whenever it is idle. The host program may discard or use these records to maintain real-time status information.

Table 4-20
Type 4 - Active Status Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “4”

EXAMPLE: (CR = end of record marker)
014<CR>

Device channel ID = 0
HHC channel ID = 1
Record type = Activate Status “4”

Type 5 - Activate Request Record

The host activates the LAN port in response to activate request records. The controller sends a request to the host to activate an enabled port after start of session and after each port disconnects. The prior connection status for this port is also in the request record. The host is expected to respond with record types “3” or “4.”

Table 4-21
Type 5 - Activate Request Record

Position	Bytes	Description
1	1	Device channel: “0”
2	1	HHC channel ID: “1”
3	1	Record type: “5”

Table 4-21
Type 5 - Activate Request Record

Position	Bytes	Description
4	1	Status code for the previous activation of the port: "0" = Good "1" = Port disconnect due to protocol error "2" = No answer "3" = Modem error "4" = Line protocol error "5" = Bad activation parameters "6" = Error activating port "7" = Auto-answer port timeout "8" = HHC error "9" = Busy " : " = Modem reported no dial tone " ; " = Modem reported no carrier " < " = Call collision. Modem reported ring.
5	4	Reserved for diagnostic information
9	252	Required padding

EXAMPLE: (CR = end of record marker)
015xyyyy<CR>

Device channel = 0
HHC channel ID = 1
Record type = Activate Request "5"
"x" = previous activation status code
"y" = diagnostic information

Type 6 - Special Request Record

Special request records lump a variety of special records into one record type. A subtype field determines the actual request type. Currently there is only one subtype defined.

Table 4-22
Type 6 - Special Request Record

Position	Bytes	Description
1	1	Device channel ID: "0"

Table 4-22
Type 6 - Special Request Record

2	1	Channel ID: “1”
3	1	Record type: “6”
4	1	Special request subtype: “1” = Obtains file processing from the host. The host responds with a file directive or an end-of-data record.

EXAMPLE:

(CR = end of record marker)
0161<CR>

Device channel = 0
Channel ID = 1
Record type = special request
Special request subtype = file processing request

Type 7 - Directive Status Record

The controller sends directive status records to the host after completing a host directive (with or without error).

Table 4-23
Type 7 - Directive Status Record

Position	Bytes	Description
1	1	Device channel ID: “0”
2	1	Channel ID: “1”
3	1	Record type: “7”
4	1	Status type: “1” - “6”: “1” = File upload “2” = File create or load “3” = File delete “4” = File download “5” = Reserved “6” = File processing short

Table 4-23
Type 7 - Directive Status Record

Position	Bytes	Description
5	1	Directive status: "0" = Good status "A" = File open error "B" = File read error "C" = File write error "D" = Maximum files exceeded on a create operation "E" = Maximum open files exceeded on a file open operation "F" = File not found "G" = Bad file handle "H" = File in use. A create operation cannot overwrite or delete a file while in use. Occurs if a file boots an HHC when host attempts to recreate the file. "I" = File not open "J" = File allocation error "K" = End of file "L" = File directory error "M" = Request aborted "O" = File system locked
6	3	Reserved for diagnostic information

EXAMPLE: (CR = end of record marker)
01710xxx<CR>

Device channel ID = 0
Channel ID = 1
Record type = Directive Status
Status type = File upload
Status = good
"xxx" is reserved for diagnostic information

Initialization Record

The host can change system parameters with an initialization record sent at the beginning of a session.

These parameters are subdivided into system, host, and port.

A slash "/" terminates initialization parameters if the initialization record is padded without spaces. The initialization record size is 258 bytes, minimum.

A parameter list has this general format: -[type]parm

[type] a one-byte field specifying the parameter type.

"parm" the parameter value.

System Mode Parameters

You can customize the controller configuration using system parameters. -S[system mode]

The system mode is a 1- to 5-digit numeric (base 10) field adjusting the operational mode for the controller. Mode parameters have a value and are turned on by *adding in* the values for each specific mode. The following mode parameters are supported in the initialization record:

```
/* defined bits in "S" record mode */
/* system mode parameters */
/* LABEL           DECIMAL         BINARY               */
   MODE_OFF        0            /* 0000 0000 0000 0000 */
   EDBDIC_ON       1            /* 0000 0000 0000 0001 */
   STRANS          4            /* 0000 0000 0000 0100 */
   SBOOT           8            /* 0000 0000 0000 1000 */
   DEBUG_ON        16           /* 0000 0000 0001 0000 */
   NCC_CHAN_ON     32           /* 0000 0000 0010 0000 */
   RMT_ID_ON       64           /* 0000 0000 0100 0000 */
   FILES_ON        128          /* 0000 0000 1000 0000 */
   RMT_A_RQ_ON     256          /* 0000 0001 0000 0000 */
   LCL_ID_ON       512          /* 0000 0010 0000 0000 */
   FORCED_AA_ON    1024         /* 0000 0100 0000 0000 */
   DIRECT_CNCT_ON  2048         /* 0000 1000 0000 0000 */
   DEACT_RQ_ON     4096         /* 0001 0000 0000 0000 */
   CRC_ON          8192         /* 0010 0000 0000 0000 */
```

EXAMPLE: If a system mode is 104 (8 + 32 + 64), then boot processing, 2-byte channel IDs, and remote controller IDs are enabled.

If no mode is defined, then the default system mode is 8 (LAN port defined by default).

Table 4-24
System Mode Parameters

Value	Meaning	Description
0	Mode disabled.	All mode parameters set off.
1	Force EBCDIC host.	Controller determines if host is EBCDIC or ASCII if this field is not added in.
4	Transparent mode on all ports.	Applies to EBCDIC hosts. When added in, ASCII or EBCDIC data translation is disabled on all ports. Data translation can be enabled or disabled for individual ports.
8	Boot 4000 Series HHCs in a LAN connected to the controller.	Enabled when a LAN port is installed and a list of boot files exists on the controller's system disk.
16	Debug mode.	Reserved for Norand diagnostic use.
32	2-byte channel option.	When added in, all records sent to the host are preceded by a 2-byte channel identifier. The 2-byte channel option is required if any of the controllers in the network are remote.
64	Forwards ID records from remote controller to the host.	Must be *on* if the 2-byte channel option is enabled.
128	Controller file processing.	When added in, the host is prompted with file requests at the beginning of a host-controller communications session.
256	Forwards port activation requests from remote controllers to the host.	Enable only if necessary for the host computer to send activation parameters to ports on remote controllers. Normally, the host does not need to see activation requests from remote controllers, unless the remote controller is directly cabled to a host controller that is directly connected to the host computer.

Table 4-24
System Mode Parameters

Value	Meaning	Description
512	Send Local controller ID on power up to host.	When added in, the local controller begins a communications session with the host by sending an ID record. This host then responds with an initialization record. Should be disabled, as it prohibits the host from using initialization record parameters to tailor the host-controller interface. When enabled, the EBCDIC host option must be set for EBCDIC host computers.
1024	Force autoanswer for remote controllers.	When enabled, a remote controller always answers the phone, even when no HHCs are ready to communicate. If not enabled, a remote controller will not answer the phone unless an HHC is ready.
2048	The secondary controller is direct-connected to the primary.	Applies to remote controllers. If enabled, a controller connected to an HHC port on a host controller always maintains the link to the host controller.
4096	Deactivate Request ON *(ASYNC only)*	If *on*, the controller periodically sends a deactivate request to the host for each enabled auto-answer port not currently active, such as a port waiting for a phone call. The host can then time window port configurations. Not useful with the 4985.
8192	Cyclic Redundancy Check (CRC) ON. *(ASYNC only)*	If *on*, the controller appends two CRC characters to the end of the upload data (type 0) records. Host appends two characters to the end of download data blocks (type 0). See Appendix C.

Host Parameters

You can customize the controllers host parameters in the initialization record.

Table 4-25
Host Parameters

Parameter	Default	Description
-A[mode]	0	1 to 5 digit numeric field setting mode for host port, such as -A0.
-C[parity]	0	1 digit numeric field setting parity and number of data bits. Valid values are: “0” = No parity/8 data bits(-C0) “1” = Odd parity/7 data bits (-C1) “2” = Even parity/7 data bits (-C2)
-D[stopbits]	1	1 digit numeric field setting stop bits. Valid values are: “1” = One stop bit (-D1) “2” = Two stop bits (-D2) “3” = Three stop bits (-D3)
-E[delimiter]	13	1- to 2-digit numeric field specifying collating sequence number of character marking end of all records sent from controller to host. A delimiter of “0” specifies no end-of-record delimiter.
-F[block size]	0	Numeric field ranging from 258 to 1029; specifies the minimum controller-to-host record block size. Does not include the end-of-record delimiter character. Records shorter than the block size are padded to the block size and the delimiter character is appended. Forces the controller to send fixed-length blocks to the host.
-G[pad char]	32	1- to 3-digit numeric field specifying the characters that pad short blocks when a minimum block size is specified.
-H[ready char]	63	1- to 3-digit field specifying the host “ready” directive.

Table 4-25
Host Parameters

Parameter	Default	Description
-I[read timeout]	30	2- or 3-digit numeric field specifying the maximum seconds the controller waits for a response from host. If no response received within set time, the controller will reset and await an initialization record from the host. Valid values are 10 to 120.
-J[host-delay]	0	1- or 2-digit numeric field specifying delay in hundredth of seconds before controller sends next message to host after receiving a ready character from host. Valid values are 0 to 50.
-K[char-gap]	10	1- or 2-digit numeric field specifying maximum intercharacter gap time in whole seconds allowed in a block of data sent from host. Valid values are 9 to 40.
-L[cts-gap]	5	1- to 3-digit numeric field specifying maximum time in tenths of seconds the controller waits for CTS after RTS. Valid values are 0 to 100.

EXAMPLE: `/2-E0-I60-K30/`

Port Parameters

Port parameters change the activation defaults for the HHC port receiving the initialization record. These changes remain in effect until you reboot the controller. The initialization record gets port parameters after the system and host initialization parameters.

Port parameters have this general format:
`@[port], [port-type], activation parameters`

`-[port]` 1 digit number specifying HHC port. The 4985 only has one port so the default is “1.”

-[port-type] 1 digit number specifying port type. Valid port types are:

- 4 LAN (NPCP) *for LAN port only*. For example: @1,4 (port 1, NPCP)

-"Activation parameters" same as those parameters allowed in an activation record for the port type specified. These are defined starting on page 4-26.

The above string would appear in the host-to-controller initialization record (type 2).

Activation Parameters

The host sends activation parameters to activate the HHC port. The host-to-controller record layouts subsections specify these record layouts. You can specify optional parameters in activation records to customize a port configuration. The host can send some of the parameters only in an activation record.

The parameter list has this general form: -[type][parameter]

-[type] 1-byte character specifying the parameter type.

[parameter] a string of digits or characters.

▪ **NOTE:** *All optional parameters begin with a dash.*

Table 4-26
Type 4 - LAN Parameters

Parameter	Default	Description
-C[channels]	3 channels	Specifies number of HHCs that communicate concurrently. Valid values are 1, 2, or 3 (-C3).
-T[timeout]	60 seconds	Specifies a "no activity" timeout period in seconds. If timeout period expires, port deactivates. Valid values are from 20 to 120 (-T60).

EXAMPLE: Async Host

```
/2 -S128 -C0 @1,4,-C3/
```

-S128	enable file processing
-C0	specifies 8 data bits and no parity.
@1,4,-C3	specifies three HHC communication channels on LAN (port 1)
/	end of string terminating initialization data. The slash is required only if record is not padded with spaces.

Section 5

Other Features

Controller-Based File Processing

“File Processing” refers to a method of transferring from the host computer updated application and data files to a local or remote controller. These files would be stored in the controller’s RAM disk. The new files will overwrite or replace the existing files on the controller. The controller can store up to 40 data files on the system. The total file size area must not exceed 360,000 bytes. The data files may contain files to boot Hand-Held Computers (HHCs), download data, or warm-boot files for HHCs using the File Server method.

To successfully implement file processing, you must:

- Understand the controller disk directory structure and format of the NCDIR.NCC and NETRPL.LST control files.
- Implement host directive, special request, and directive status records on your host computer.
- Know how to upload the NCDIR.NCC control file from the controller to your host computer.
- Perform version checking of HHC application software program changes done on the host computer against each HHC for proper version numbers.
- Know how to download to HHCs, without a proper version, a “download request” file. The “download request” file must have the proper filenames located on the controller.

File processing first starts with the knowledge of what files are currently on the controller and valid for file processing. This information is stored in the NCDIR.NCC file. During the file processing phase of communications where the host application and the controller exchange information, the host should retrieve the NCDIR.NCC file first to compare the contents of the controller file system with the current host files. Here, any files that may have been changed or need to be removed from the controller can be processed.

NCDIR.NCC

The NCDIR.NCC file contains zero, one, or multiple records on all valid files currently loaded on the controller. This information includes filenames, date created, file size, and file status. The record size is 32 bytes. The record layout is as follows:

Table 5-1
NCDIR.NCC Record Layout

Position	Bytes	Description
1	8	MS-DOS compatible filename *(required)*
9	3	MS-DOS compatible file extension *(required)*
12	6	Creation date (YYMMDD)
18	4	Creation time (HHMM, military 24-hour clock)
22	2	Pad
24	1	Pad - “0” or blank HHC boot file
25	1	Translation type
26	1	File status: “0” = good “1” = undefined “2” = deleted “3” = system
27	6	File length (in bytes) *required*

Pad the last data record for a file with blanks if the file length is not an even multiple of 256. Do not include the pad in the file length field in the directory entry for the file.

EXAMPLE: An entry in the NCDIR.NCC file (* represents blank spaces)

```
APP*****EXE9308191134**000110667
```

The NCDIR.NCC file should be the first file sent from the controller to the host. It should also be the last file sent back to the controller from the host to update the controller. Once the host receives the NCDIR.NCC file, the host must be able to break the file into the records as defined earlier and compare the contents with the files on the host.

After retrieving the NCDIR.NCC file and validating or marking the files to be deleted, created, or downloaded to the controller, the host can begin sending the proper sequence records to the controller for file processing. This sequence of commands can be summarized into the following:

- For each record the host wants uploaded from the controller, the host would send an upload directive with the filename.
- If the host is downloading a new file to the controller, the host would send a download directive or a “create or load” directive.
- The host, after each directive record, should be aware of the status records from the controller and act appropriately based on the status. The last file sent to the controller should be the new updated NCDIR.NCC file.
- 4000 cold booting changes can be affected by file processing. The 4000, when cold booting, uses the NETRPL.LST file containing a list of files that are sent to it when the 4000 requests a boot session. If the host did not send a NETRPL.LST file, the controller will create a file listing all files in the controller as boot files. Normal operations do not affect the NETRPL.LST file.

Use an “Abort” record to abort the file processing session and lock the 4985 file system. The file system unlocks after a successful file processing session.

Pad the last data record for a file with blanks if the file length is not an even multiple of 256. Do not include the pad in the file length field in the directory entry for the file.

All directory entry fields must contain ASCII character data.

Fill all fields in a directory entry when creating a directory entry, except for the file status field. The file status field indicates the current status of the file.

Enter these fields:

- Filename
- Translation type
- File length

The filename and extension entries must be MS-DOS compatible, left-justified, and padded with blanks. If not using an extension, pad the extension with blanks.

The translation field applies only to file data sent from a host. The file length field must contain the exact length of the file in bytes, be right-justified, and padded to the left with leading zeros.

You can fill in the date, time, fill, and type fields. However, future NORAND® software releases may use the specified fields.

EXAMPLE: Directory entry: `MYFILE DAT9606222322 000001064`

MYFILE	Filename: MS-DOS compatible
DAT	File extension: MS-DOS compatible
960622	Date created: June 22, 1996
2322	Time created: 11:22 pm
00	User defined filler
0	File type - HHC boot file
0	Filler
0	File status: Good
1064	File length is 1064 bytes

If enabled on the controller (file processing), the host can update controller files at the beginning of each host-to-controller session. The controller provides four directives for this purpose:

- 1 Upload
- 2 Create or load
- 3 Delete
- 4 Abort

After the host sends the initialization record, the controller sends file requests to the host to initialize file processing and will continue to send file requests to the host after each file processing transaction concludes until the host responds with an end-of-data record, terminating file processing, and beginning data communications.

Host-to-Controller File Processing Related Commands

NOTE: *See Section 4 for details on record types and definitions.*

Initialization Record (Type 2)

The host sends the initialization record to the controller at the beginning of each communication session. The host can activate file processing in the controller using the system mode parameter (-S) in the initialization record. The entire parameter list is located in Section 4 on page 4-20.

EXAMPLE:

```
/2 -S224/
```

Initialize controller with the following controller options enabled:

- Controller file processing
- Forward ID records from remote controller to host
- 2-byte channel option

Host Directive Record (Type 6)

The host sends a directive record only during the file processing phase of a communication session. This record requests information from the controller or file uploads with the controller. The entire parameter list is located on page 4-1.

A host directive may be sent by the host in response to a special request from the controller. Types “1” to “3” manage user data files (or HHC boot files) stored on the controller. (If file processing, a system mode parameter is enabled and the host is prompted with file requests at the beginning of a host-to-controller session.) Type “4” sends a file stored on the controller to an HHC requiring download data.

Host Directive Subrecords (Types 1-3)

File Upload (Subtype 1)

An upload directive may be sent in response to a file request to upload a data file from the controller.

File Create or Load (Subtype 2)

A file create or load directive may be sent in response to a file request to create a user data file on the controller. The directive must contain a user data file directory entry which includes a filename and user-defined information. The file is saved into a RAM disk. After the controller receives the create directive, the host is prompted with data request records. The host can send data records in response until all the new file data has been sent. An end-of-data record from the host marks the end of the data for the file.

File Delete (Subtype 3)

A file delete directive may be sent in response to a file request to delete a user data file from the controller user file directory.

▪ **NOTE:** *A directive status record (type 7) goes to the host immediately after a host directive has been completed.*

Controller-to-Host File Processing Related Commands

" **NOTE:** *See Section 4 on page 4-13 for record types and definitions.*

Controller-to-Host Records with 2-Byte Channel ID

Records from the controller to the host can, optionally, use a 2-byte channel ID consisting of a 1-byte “device channel” field and a 1-byte “HHC channel” field. HHC channels belong to the device active on the corresponding device channel. Device channel “0” is reserved for the controller directly connected to the host computer. Device channels “1” to “9” are for remote controllers. The record formats for the 2-byte channel option are the same as the 1-byte channel option except that the device channel ID is prepended to each record from the controller.

Directive Status Record (Type 7)

After a directive has been completed, successful or not, the controller sends a directive status record to the host containing the directive type and a 1-byte status code. The status codes for the file are on page 4-19.

Booting HHCs from Controller

The controller can boot 4000 Series HHCs in the Local Area Network (LAN) environment with a list of files kept in a file named NETRPL.LST.

“Boot” is the industry jargon for bringing a computer to normal operating condition when it has been without power. Norand has adopted the term to describe the same process for starting our 4000 Series Computers.

NETRPL.LST must be one of the directory entries in the file directory NCDIR.NCC (described on page 5-1).

NETRPL.LST is a pointer file that consists of filenames needed for *booting* the HHC.

NETRPL.LST is composed of 11-byte records. Each record contains an 8-byte filename and 3-byte extension. A blank filename terminates the list of files.

These filenames must have corresponding directory entries in the file directory NCDIR.NCC.

When the *boot option* is in effect, the controller attempts to send all files specified in NETRPL.LST to any terminal trying to boot on the attached LAN.

NETRPL.LST

The controller can boot 4000 Series HHCs in the LAN environment with a list of files kept in a NETRPL.LST file.

NETRPL.LST is a data file residing on the controller. It consists of a series of 11-byte records. Each record contains two fields, the filename and the filename extension. A blank record terminates the list of files.

When the boot option is in effect, the controller attempts to send all files specified in the NETRPL.LST file to any terminal on the LAN attempting to boot.

Table 5-2
NETRPL.LST File

Position	Bytes	Description
1	8	DOS compatible filename (right-justified, padded with blank spaces)
9	3	DOS compatible filename extension

Host-to-Controller Session Sample 1

This shows the exchange of control records between the host computer and the controller when there are no files residing on the controller.

■ **NOTE:** *Asterisks indicate spaces in the command column. Complete commands may not be shown.*

Host		Controller	Command
Init Record	-------->		/2 -S224
Host Ready	-------->		?
	<--------	File Processing Request	0161
Upload File Directive	-------->		/61NCDIR***NCC
Host Ready	-------->		?
	<--------	Data Record	010*********
Host Ready	-------->		?
	<--------	Status Record	01710
Host Ready	-------->		?
	<--------	File Processing Request	0161
File Create/Load Directive	-------->		/62NETRPL**LST9507050914*****000022
Host Ready	-------->		?
	<--------	Download Request	012
Data Block	-------->		/62APP*****EXE9507021412*****021123
Host Ready	-------->		?
	<--------	Download Request	012
Data Block	-------->		/0<data>
Host Ready	-------->		?
.			
. <repeat until all data downloaded for file>			
.			
	<--------	Download Request	012
End of Data Record	-------->		/1
Host Ready	-------->		?
	<--------	Status Record	01720
Host Ready	-------->		?
	<--------	File Processing Request	0161
End of Data Record	-------->		/1
<end of file processing phase>			
<beginning of data communication phase>			
	<--------	Activate Request	/0150

Host-to-Controller Session Sample 2

This shows the exchange of control records between the host computer and the controller when there are files residing on the controller.

" **NOTE:** *Asterisks indicate spaces in the command column. Complete commands may not be shown.*

Host		Controller	Command
Init Record	-------->		/2 -S224
Host Ready	-------->		?
	<--------	File Processing Request	0161
Download File Directive	-------->		/61NCDIR***NCC
Host Ready	-------->		?
	<--------	Data Record	010APP*****EXE9507021412*****021123 NETRPL**LST9507050914*****000022
Host Ready	-------->		?
	<--------	Status Record	01710
Host Ready	-------->		?
	<--------	File Processing Request	0161
File Delete Directive	-------->		/63NETRPL**LST
Host Ready	-------->		?
	<--------	Status Record	01730
Host Ready	-------->		?
	<--------	File Processing Request	0161
Data Block	-------->		/62NETRPL**LST9507021422*****000022
Host Ready	-------->		?
	<--------	Download Request	012
Data Block	-------->		/0APP*****EXE
Host Ready	-------->		?
	<--------	Download Request	012
Host Ready	-------->		?
	<--------	Status Record	01720
Host Ready	-------->		?
	<--------	File Processing Request	0161
End of Data Record	-------->		/11
Host Ready	-------->		?
<end of file processing phase>			
<beginning of data communication phase>			
	<--------	Activate Request	/0150

Section 6

Writing the Host Program

The communications controller offers several options designed to provide flexible options for reading records from *the communications con troller.*

- You can add an end-of-record delimiter to the end of each record. The default delimiter is a carriage return.
- You can configure the communications controller to pad all records to a fixed-block size. The host sets the pad character. The default is an ASCII space.
- The host-ready character, a question mark by default, can change to any other character, such as an XON character.
- The upload data record data portion may contain transparent data if the port configures for transparent mode with EBCDIC hosts. All other data sent to the host is character data.
- The network communications controller reads the initialization record as 8-bit data, no parity, and one-stop bit, so that the host can set the parity for the host-communications controller connection in the initialization record.

If the host is reading transparent data from the communications controller, then the host cannot read to an end-of-record delimiter. You could solve this problem by padding all records to a fixed length. Yet, this is inefficient as extra length would be added to download requests.

As an alternative, the system reads the channel ID and record type portion of each record before the rest of the record. The host could then use a table, indexed by record type, to determine the number of bytes to read for the record.

" **NOTE:** *The system sends the asterisk before the initialization record or when the controller expects a question mark to reset the controller.*

The purpose of this section is to help you write a host communications program. The pseudo-coded host program acts as a guide to writing your own program.

There are many ways to write a computer program. Each method has its advantages depending on the programming language used. Host computers may have different operating systems, device names and assignments, or linkages. Norand provides a generic pseudo-coded host program that includes most everything you need to know for your host computer to communicate with the 4985 Network Controller.

" **NOTE:** *The pseudo programs in the following subsections do not have a status screen. They do not send blank initialization and activation records to the controllers. The program does run until a communication error occurs.* ***You must thoroughly test all code before including it in your system.***

" **NOTE:** *For the latest "C" programs available on BBS, call (319) 369-3515.*

Starting the Main Routine

The main routine is a top-level routine directing the flow of the program. Physically, this routine determines if a controller is trying to communicate with it, reads a record, checks parameters, and depending on record type, performs or calls a subroutine to do the real work. When done reading records, it closes the files and ends.

There are several subroutines that check errors and parameter. If errors appear, or a field's contents are invalid, the program outputs a message to a log file or a display.

EXAMPLE: Pseudo-code program Main Routine

```
Main Routine
Open Files
Initialize counters to 0 and port/channel states to inactive.
Send Initialization Record to controller.
```

```
While Not Finished
    Send the ready character.
    Perform Controller_Read
    Decode (channel) and Record (type)
    If type is Upload_Data
        Perform Received_Upload_Data_Routine
    Else if type is End_Tcom_Status
        Perform End_Status_Routine
    Else if type is Download_Data_Request
        Perform Download_Data_Routine
    Else if type is Inactive_Status
        Update host display “port inactive”
    Else if type is Active_Status
        Update host display “port active”
    Else if type is Activate_Request
        Perform Activation_Routine
    Else if type is Special_Request
        Perform File_Processing_Routine
    Else if type is Directive_Status
        Perform Directive_Status_Routine
    Else
        Perform Invalid_Record_Routine
    Endif
End While Not Finished
Close Files
End Program
End Main Routine
```

Receiving Data

The controller-to-host record type read was an *Upload Data Record.* This means a HHC on the controller’s LAN is sending upload data to the host computer. You need to:

1. Determine which channel receives the data.
2. Ensure that the received data writes to the proper receive file.
3. Send a “READY” command to indicate to the controller to send the next data block.

You return to the Main Routine to read the next record from the controller.

The following example is a simplified version of the communication session using a three-channel LAN.

EXAMPLE: File processing session

```
Host                                              Controller

1. Get initialization parms for device
Send Init “2 -S0/”                     -------->
(do not wait for response)
“?” (ready)                            -------->   first HHC begins to transmit
                                       <-------    Chan 1 Upload Data
Set up data save area for channel 1
“?” (ready)                            -------->
                                                   second HHC begins to transmit
                                       <--------   Chan 7 Upload Data
Set up data save area for channel 7
“?” (ready)                            -------->
                                                   second HHC transmits another
                                                   block of data
                                       <--------   Chan 7 Upload Data
Add record to data save area already set up
“?” (ready)                            --------->
                                                   third HHC begins to transmit
                                       <--------   Chan 4 Upload Data
Set up data save area for channel 4
“?” (ready)                            -------->
```

Host		Controller
		second HHC transmits another block of data
	<--------	Chan 7 Upload Data

EXAMPLE: Pseudo-code program Upload Data Routine

```
Upload Data Routine
If channel is inactive
        Save HHC ID from first upload record in save area for channel
        Reset upload sequence counter for channel.
        Change channel state for channel to “uploading.”
        Write upload record with channel/HHC ID/sequence counter
        key.
Else
    Increment upload sequence counter.
    Write upload record with channel/HHC ID/sequence counter key.
Endif
End Upload_Data Routine
```

Download Data

The HHC sends a download data request through the controller to the host computer. The host program needs to send a download data record, an end-of-data record, or a download file directive in return.

EXAMPLE: Pseudo-coded Download Data routine

```
Download Data Routine
If channel state is uploading
    log good status for upload
    reset download sequence counters for channel
    change channel state to downloading
```

```
Endif
If there is more download data for the HHC
    send download data record
Else
    send an end-of-data record
Endif
Return
End Download_Data_Routine
```

" **NOTE:** *One way to associate download data with an HHC is to maintain a* ***download request*** *file which contains a list of* ***download*** *files for each scheduled HHC. The list of files may actually be a list of keys for indexed file systems.*

Ending Status Routine

When the host receives this record, the record indicates that an HHC has terminated its communications session. The session information must be recorded in a log file. The log file is helpful when starting out, as it allows you to debug your code faster.

EXAMPLE: Pseudo-coded program Ending Status Routine

```
End Status Routine:
Initialize log record.
Move "END" to log record type
Move the session key (channel, HHC ID) to log record key.
Move Upload_count to log record upload counter.
Move Download_count to log record download counter.
Move current_status to log record status.
Add the log record to the upload status.
Change session status to inactive.
Display log status of HHC telecom session.
```

```
/* close and reopen the upload file to flush the buffers to disk */
Perform Upload_flush.
/* write the log record */
Write log_rec to the log file.
Close the log file.
    Reset the channel status to inactive.
    Return.
End End_Status_Routine.
```

Handling Time-Fill Records

The controller periodically sends either an Inactive Status Record (record type “3”) or an Active Status Record (record type “4”). These records indicate the logical port’s current status, and are sent whenever the controller is idle. The idle period could occur during a telecom session with an HHC as it switches from upload to download mode, or, when the controller is waiting for the next HHC to begin its telecommunications.

When the host program receives the Inactive or Active status record, it should possibly update a status display or write status information to a log file. These records primarily tell the host program that the controller is still alive, and should not be disconnected through an automatic timeout.

Activating LAN Port

The host program activates the LAN port in response to active request records. The controller sends requests for the port whenever it disconnects. The prior connection status for this port is contained in the uploaded activate request record. The host program should respond with either an activate auto-answer record (record type 3) or an activate auto-dial record (record type 4).

Processing Special Requests

Special request records combine a variety of special records into one record type. Currently only one subtype is defined: File Request. File requests obtain file processing information from the host. The host responds to this record with either a file directive or an end-of-data record to return to normal data communications. See pages 5-8 through 5-10 for sample host-to-controller communications sessions.

Processing Directive Status Records

The controller sends a directive status record to the host after completing a host directive. Check this record for a good read before sending the next record, or for an error status. You may wish to set up a different routine for handling each error type returned.

Valid status responses for file processing directives returned from the controller are detailed on page 4-19.

Adapting Older Host Programs

Host programs designed to run with an NI311 require minimal changes, as specified below, to communicate with the controllers running in a *compatibility mode.*

The required changes generally fall into the following categories:

- **Record Types:** the controller adds a number of new record types and the function of some of the old NI311 record types has changed.
- **Logical Channels:** The controller may have up to three logical channels per physical port.
- **New Enhancements:** The controller provides a number of new enhancements which may be enabled when the host is ready to support them.

The other *major area of change* is in the data record. The length of the data block has increased from 128 bytes to 256 bytes.

Record Types

Record types for both the NI311 and controllers follow. Listed are differences (if any) in how you can use the records.

NI311 Record Types

NI311-to-Host

a) 0 Upload Data

b) 1 End-of-call Status

c) 2 Data Request

Host-to-NI311

a) 0 Download Data

b) 1 End-of-data

c) 2 Initialization

d) 3 Cancel auto-dial

f) 5 Deactivate auto-dial for one minute.

4985 Network Controller

The length of the data block has increased from 128 bytes to 256 bytes.

Controller-to-Host

a) 0 Upload Data

b) 1 End-of-session Status

After each HHC completes the communication session, the system sends end-of-session records to indicate the status of a single HHC session. This differs from the NI311 end-of-call record that provides the status after the system loses a port connection.

c) 2 Data Request

d) 3 Inactive Status

The system sends inactive status records to the host whenever a period of time expires without any activity on the host port. The status records indicate the status of the associated port.

e) 4 Active Status

See previous Inactive Status record.

f) 5 Activate Request

Activate request records obtain the information, necessary to activate a port. Activate requests include the status of the previous activate request for the port. The system sends these requests for a physical port whenever the port becomes disconnected. The host may respond with with one of these activate records: auto-dial, auto-answer, or deactivate-for-one-minute.

For compatibility, the activate request record processes exactly like an NI311 phone request record with the following restrictions:

- A 1-byte status field follows the record type in the activate request. The status field indicates the status of the previous activate request. This status corresponds to the NI311 end-of-call status.
- Responding to an activate request with an auto-answer activate record corresponds to responding to an NI311 phone number request with a cancel-auto-dial record. However, the controller will continue to prompt the host with activate requests each time a port disconnects.

The following record types are not NI311 compatible and do not go to the host unless the system enables the associated options.

g) 6 Special Request

Currently, one subtype follows:

1 File Requests

h) 7 Directive Status

i) 8 Controller Identification

Host-to-Controller Records

a) 0 Download Data

b) 1 End-of-Data

c) 2 Initialization

The initialization record supplies system configuration parameters.

Store initialization information in a changeable file to avoid host program changes.

The controller uses the initialization record to determine the block size of the host computer. The block size includes a beginning-of-record delimiter, record type, data, and an optional “end-of-line” pad character (a carriage return). Minimum block size is 258 bytes.

▪ **NOTE:** *All records sent to the controller, except special commands, must have the same size. Any end-of-line pad, appended-to-records, and special commands sent to the controller must be consistent.*

d) 3 Auto-answer Activate

Auto-answer activate records respond to an activate request and may include optional port configuration parameters. The auto-answer activate record replaces the NI311 cancel-auto-dial record.

e) 4 Auto-dial Activate

Auto-dial activate records respond to an activate request, and may include optional port configuration parameters. The auto-dial activate record replaces the NI311 phone number record.

Auto-answer and Auto-dial Activate records process identically. The auto-answer type is provided for backward compatibility.

f) 5 Deactivate-for-one-minute

▪ **NOTE:** *The following record types are not NI311 compatible, and cannot be sent to the controller unless the system enables the associated options.*

g) 6 Host Directive

The host may send a host directive in response to a special request or data request from the controller. Currently, five subtypes of host directives follow:

- 1 File Upload
- 2 File Create or Load
- 3 File Delete
- 4 File Download
- 6 File Processing Abort

Special Commands

In addition to the preceding records, the host may send two other commands to the controller.

- The host must send a question mark (?) to the controller to indicate that it is ready to receive data. You can change the question mark to any other character (such as XON) with an initialization record parameter. If optional pad characters follow the question mark, they must be consistent.
- The host may reinitialize the controller by sending an asterisk (*). This parallels sending a slash and an asterisk (/*) to the NI311. The host must wait at least 15 seconds before sending another initialization record after sending the asterisk.

" **NOTE:** *The system sends sends an asterisk before the initialization record or when the controller expects a question mark.*

Logical Channels

The controller has two physical ports logically numbered “0” and “1.” Logical port “0” is always the host port. Port “1” is the HHC port. There may be up to three logical channels for the HHC port.

EXAMPLE: If enabled, port “1” may have logical channels “1,” “4,” and “7.”

Records from the controller (specific to a port, such as Activate Requests) always use the logical channel that equates to the logical port

ID. Data to or from an HHC is sequential on a logical channel and will always be port "1." This occurs after establishing the physical connection on a port.

Data Record Length

The controller data record length is 256 bytes, not including channel and record type bytes. The NI311 data record length is 128 bytes.

Enhancements

The communications controller provides a number of new features:

- **File Support:** The controller allows the host computer to download boot files used to boot an HHC independent of the host computer.
- **Line Discipline:** The controller has a single LAN port which uses NPCP.
- **Booting HHCs:** The controller can boot HHCs and the host downloads boot files.
- **Changing System and Port Configurations:** The host computer can change the controller system and port configurations with initialization and activation parameters.
- **DWNLRQ File Server:** The controller has a DWNLRQ file server which services file requests from HHCs. The file server can *warm start* HHCs. The server is always active except with a locked file system.

You can add support for any of the above enhancements to an existing controller host program running in an NI311 compatibility mode without changing the general structure of the host program.

Section 7

4000 Series Warm Start System

Using the 4985 *warm start* system requires changing the Hand-Held Computer (HHC) application program. This change has two parts to implement:

- Automatic program updates
- DWNLRQ file requests

The first change requires the application to overlay itself with an updated version. The second change requires the application to interpret a command file, directing the HHC to set up for a second session. The application then requests a program update from the DWNLRQ file server.

The HHC application binds to the file server (NORAND_SERVER). Code the application to obtain new program files from the host when the files are not available from the server because the server is unavailable or the HHC attaches to a modem.

NOTE: *The 4000 Series HHC receive routine uses IPFRCV, and cannot pack 7-bit Z-space hex files. Therefore, any PL/N application implementing automatic program updates must use the 8-bit binary format. Consult the Nordev Toolkit, software kit #210-682-001.*

Complete the following items before controller-based file processing can begin. Details follow:

- Set the ID on the remote controller. The default 4985 Controller ID is “49850000.” A unique ID is not required for each controller, but is recommended.
- Create a file maintenance list for each controller ID.

- Create an HHC program load “command file” and add a program load request for each HHC. A single request can broadcast to all HHCs.
- Build a session control file.
- Set the current version number of the HHC application.
- Copy the HHC application program files into the download directory.
- Copy the filenames specified as boot files into a boot list file for the associated controller device.
- Execute session control formatting (if using 4921 Series of PC Telecommunications Packages). The boot file list, created with the device ID as the filename, goes to the controller as NETRPL.LST.
- Convert the HHC program to support program updates and DWNLRQ program requests.

Once the *warm start* system is in place, distribute a new application by copying the program files into the DWNDIR directory.

Creating File Maintenance Lists

" **NOTE:** *Many of the configuration specifics are related to the operation of the 4921 Series of PC Telecommunications Packages.*

The host communications program needs to create file maintenance lists for the controllers. This file list associates with the controller ID in the host session control file.

Create a file maintenance list for each remote controller ID, using a host session control file interface. Use a boot-type parameter to generate a boot list (NETRPL.LST) for the controller.

EXAMPLE: Host session control maintenance list

```
N     498500201 type=boot dir=bootdir
F     pacxm0p type=data dir=dnldir
F     pacx10p type=data dir=dnldir
```

F sysarc.exe

- Controller ID is “49850201”
- Type and dir parameters in the “N” record set default type (boot) and directory (bootdir), for the following files
- Pacx files (pacxm0- and pacx10p) represent modules in HHC PL/N application
- sysarc.exe file is self-extracting archive which contains HHC operating system files and kernel program

In this example, when an HHC is cold-started, it receives the SYSARC file. After the operating system file “boots” the HHC, it passes control over to the kernel program.

The kernel program immediately sets up a telecommunications session with the DWNLRQ file server to request download of the application program and data files. The boot process is complete when control passes to the application.

After the application is running, it overlays *(warm starts)* itself by passing control to a communications module. The communications module requests the new PACX files from the server. The *warm start* initiates from the host by scheduling a program load request for the HHC.

" **NOTE:** *The controller file maintenance subsystem does more than boot and **warm start** files.*

Your host communications program should provide a means for updating HHC application programs. A program version number and program load request can associate with HHC identifiers in the host session control file.

The program load requests activate when the program version on the HHC does not match the program version number associated with the HHC identifier on the host.

Program the application on the HHC to overlay itself if you want to support *warm starting.* You do not have to write the HHC application to support the DWNLRQ file server on the controller.

If the HHC application does not support the DWNLRQ file server, direct the host communication program to download a new program to each HHC with an out-of-date application.

If the HHC application supports the DWNLRQ file server, direct the host communication program to download a command file to the HHC. This command file directs the HHC to set up for a second session to receive the set of program files specified in the command file.

The DWNLRQ file server system provides a mechanism for 4000 and 6000 Series HHCs to request files from a locally attached NORAND® LAN controller. The session interface is identical to the interface that the HHCs use to communicate with a host computer. To access the server, the application must change the name of the target application to which it is binding from NORAND_HOST to NORAND_SERVER. The server expects file requests like PL/N DWNLRQ request files.

```
<DDWNLRQ00003X016>MYDATA.DAT PMBNX0P PMBNG0P
```

The server sends the files as requested until:

- A fatal communications error occurs
- A file cannot be found
- All files have been sent

The server supports MS-DOS filenames and extensions, but does not support a path name.

Setting Version Number

Set the current version number of the HHC application. A version number can enter the mode or terminal identification records in the session control file.

A version number on the mode record is global. A version number on the terminal identification record applies to a specific terminal.

The terminal identification option is for multiple application support.

Converting HHC Application Programs

Convert the HHC program to support program updates and DWNLRQ program requests.

Once the *warm start* system is in place, a new application distributes by copying the program files into the host download directory.

Due to improvements in hardware design in remote data communication technology, the 4000 and 6000 Series HHCs maintain their operating system and programs in Random Access Memory (RAM) rather than in Read-Only Memory (ROM).

Therefore, the HHCs must have the operating system, programs, and data sent to them when you *cold boot* them into service for the first time.

There are two changes required before your HHC application can utilize the host-to-controller *warm start* system.

The first change requires that the application can overlay itself with an updated version. The second change requires that the application can interpret a command file. The command file directs the HHC to set up immediately for a second session. Then the file requests a program update from a controller DWNLRQ file server.

The HHC application can bind to the controller file server (NO-RAND_SERVER). Code the HHC application to obtain new program files from the host when the files are not available from the controller's server. Either the controller has suffered a malfunction (causing it to lose the files from its memory), or the HHC attaches to a modem and uses TTY to communicate.

The receive routine for the 4000 HHC, (IPFRCV), cannot pack 7-bit z-space hex files. Therefore, any PL/N applications that implement automatic program updates must use the 8-bit binary format.

Appendix A

Power-Up Self-Tests

The controller performs self-tests each time it powers on. More than two beeps refer to problem areas. A long beep or a series of beeps lasting longer than 30 seconds indicate internal hardware problems. Note how many beeps, then refer to Table A-1. Send your unit to a service center for repair. Include the beep count for the service technician.

Table A-1
Beep Count

Beep Count	Meaning
1	power-up tests complete
2	host port ready
3	RAM test error
4	CPU test error
5	CPU timer test error
6	CPU DMA controller test error
7	CPU interrupt controller test error
8	even ROM test error
9	odd ROM test error
10	SCC test error
11	bad opcode trap
12	bad interrupt vector trap
13	heap memory exhausted

Appendix B

Communication Session

" **NOTE:** *This sample has no file processing.*

ASYNC Host Computer		Network Controller
Init “2 -S0 / ”	-------->	
“?”	-------->	
	<--------	Chan 1 Activate Request
Activate Parms	-------->	
“?”	-------->	
	<--------	Chan 1 Active Status Record
“?”	-------->	
	<--------	Chan 1 Upload Data
“?”	-------->	
	<--------	Chan 7 Upload Data
“?”	-------->	
	<---------	Chan 7 Upload Data
“?”	-------->	
	<--------	Chan 4 Upload Data
“?”	-------->	
	<--------	Chan 4 Activate Status
“?”	-------->	
	<--------	Chan 7 Upload Data
“?”	-------->	
	<--------	Chan 4 Upload Data
“?”	-------->	
	<--------	Chan 1 Data Request
Download Data	-------->	

ASYNC Host Computer		Network Controller
"?"	-------->	
	<--------	Chan 1 Data Request
Download Data	-------->	
"?"	-------->	
	<--------	Chan 7 Upload Data
"?"	-------->	
	<--------	Chan 1 Data Request
End of Data	-------->	
"?"	-------->	
	<--------	Chan 4 Upload Data
"?"	-------->	
	<--------	Chan 1 End Status
"?"	-------->	
	<--------	Chan 7 Data Request
End of Data	-------->	
"?"	-------->	
	<--------	Chan 4 Data Request
Download Data	-------->	
"?"	-------->	
	<--------	Chan 7 End Status
"?"	-------->	
	<--------	Chan 4 Data Request
End of Data	-------->	
"?"	-------->	
	<--------	Chan 1 Upload Data
	.	
	.	
	.	

Appendix C

CRC Checking with Asynchronous Interface

Cyclical Redundancy Checking (CRC) is a method that verifies the integrity of data. It works by processing a block of data, bit by bit, with a known arithmetic algorithm designed to detect data errors. The NORAND® Controller has an option to enable this detection for data transferred between the host and the controller to ensure that data has not been corrupted. This 16-bit CRC checking and encoding is performed on data records only. If CRC checking is enabled in the initialization record, all data (type “0”) records have the following format:

- 4985 data records

```
<channel ID><record type “0”><256 bytes of data><2-byte CRC>
```

- Host data records

```
/0<256 bytes data><CRC byte><CRC byte>
```

`-S8129` enables CRC checking in the system mode parameter in the initialization record sent to the 4985 from the host. The system mode parameter is the `-S<system_mode>` parameter in the initialization record (see page 4-20).

The sample “C” subroutine, CRCCALC, calculates a 16-bit CRC (CRC-CCITT) on the data buffer. If `addcrc` is true, the calculated CRC is returned in unsigned character variable crcbytes. Otherwise, the calculated value is compared against the value passed in crcbytes and a non-zero error code is returned if the comparison fails. This routine performs the byte-wise calculation on the fly. A faster implementation would include generating a CRC lookup table and using it to look up the CRC rather than calculating it.

```
int crccalc (char* header, int hsize, unsigned char*
crcbytes, int addcrc)
```

```
unsigned int crccalc(unsigned char *buffer, unsigned int
size, unsigned int addcrc);
/*****************************************************
* CALCULATE CRC FOR BUFFER                            *
* inputs:                                             *
*    buffer - contains information to calculate crc   *
*    size -   size of buffer to process               *
*    addcrc - if nonzero, add 2 crc bytes to end of   *
*             block.                                  *
*             if zero, calculate and compare crc to   *
*             last 2 bytes.                           *
* returns:                                            *
*    0 = good crc                                     *
*    1 = bad crc                                      *
*****************************************************/
unsigned int crccalc(unsigned char *buffer, unsigned int
size, unsigned int addcrc)
{
  register int i;
  register int crc;
  char* ptr;
  int count;

  count=size;
  crc=0;
  ptr=buffer;

  while (--count >= 0)
  {
    crc=(int) *ptr++ << 8;
    for (i=0; i<8; i++)
    {
      crc=((crc & 0x8000_ ? 0x1021 : 0) ^ (crc << 1);
    }
  }

  if (addcrc)
  {
    buffer[size]=(0xff00 & crc) >> 8;
    buffer[size+1]=(0x00ff & crc);
  }
  else
    if (buffer[size] !=(unsigned)((0xff00 * crc)>>8) ||
        buffer[size+1] !=(unsigned)(0x00ff & crc))
        return(1);
    return(0);
}
{
    if (ptr) /*headr or buf may be NULL*/
    {
        while ( -- count >= 0)
        {
            crc ^=(int)*ptr++ << 8;
            for (i=0);i < 8; i++)
            {
                crc=((crc&0x8000)) ?0x1021:0)^(crc<<1);
            }
        }
    }
ptr=buf;
count=bsize;
}
if (addcrc)
{
    crcbytes[0]=(0xff00&crc)>>8;
    crcbytes[1]=(0x00ff&crc);
}
else if (crcbytes[0]!=(unsigned)((0xff00&crc)>>8) ||
                crcbytes[1]!=(unsigned)(0x00ff&crc))
    {
        return -1;
    }
return 0;
}
```

Appendix D

File Structure Records

" **NOTE:** *Section 4 has complete details on these file structures.*

Host-to-Controller Records

Special Commands

Table D-1
Type ? - Host Ready Command

Position	Bytes	Description
1	1	Question mark “?”

Table D-2
*Type * - Reinitialization Command*

Position	Bytes	Description
1	1	Asterisk “*”

Record Layouts

Table D-3
Type 0 - Download Data Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “0”

Table D-3
Type 0 - Download Data Record

Position	Bytes	Description
3	256	Download data
259	2	Optional CRC bytes (if enabled)
261+		Optional pad (up to 3 bytes)

Table D-4
Type 1 - End-of-Data Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "1"
3	256	Download data with required pad bytes if less than 256 bytes of data.
259+		Optional pad (up to 5 bytes)

Table D-5
Type 2 - Initialization Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "2"
3	256	Initialization parameters list. The parameter list terminates with a slash "/."
259+		Optional pad (up to 5 bytes)

Table D-6
Type 3 - Activate (Auto-Answer) Record

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "3"

Table D-6

Type 3 - Activate (Auto-Answer) Record

Position	Bytes	Description
3	1	ASCII blank “ ”
4	255	Optional activation parameters list. This list terminates with a slash “/” and the remaining bytes filled with blanks.
259+		Optional pad up to 5 bytes

Table D-7

Type 4 - Activate (Auto-Dial) Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “4”
3	1	ASCII blank “ ”
4	255	Optional activation parameters list. This list terminates with a slash “/”. When used, this field should be filled with blanks.
259+		Optional pad up to 5 bytes

Table D-8

Type 5 - Deactivate for One Minute Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “5”
3	256	Required pad bytes
259+		Optional pad up to 5 bytes

Table D-9
Type 6 - Host Directive Record

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “6”
3	1	Directive type: “1” - “4” and “6”: “1” = File upload directive “2” = File create or load directive “3” = File delete directive “4” = File download directive “6” = File processing abort
4	256	Directive information. The information required for this field depends on the directive type used.
259+		Optional pad up to 5 bytes

Table D-10
Type 7 - Reboot Controller Directive

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “7”
3	1	Reboot state: “0” = Do not clear files from RAM disk “1” = Clear files from RAM disk
4	256	Filler bytes

Directive Types

Table D-11
File Upload Directive “1”

Position	Bytes	Description
1	1	Beginning delimiter: “/”
2	1	Command code: “5”

Table D-11
File Upload Directive "1"

Position	Bytes	Description
3	1	Directive Type: "1"
4	8	DOS compatible filename, left-justified
12	3	DOS compatible filename extension, left-justified
15	245	Filler bytes (ASCII space)

Table D-12
File Create or Load Directive "2"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "2"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	6	Date field (YYMMDD)
21	4	24-hour time (HHMM)
25	2	User-defined filler
27	1	User-defined file type "0" or blank indicates an HHC boot file.
28	1	Filler *(should be zero filled 0x30)*
29	1	File status: "0" = Good "1" = Undefined "2" = Deleted "3" = System
30	6	File size (in ASCII digits) *required*
36		Filler bytes

Table D-13
File Delete Directive "3"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "3"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	245	Filler bytes

Table D-14
File Download Directive "4"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "4"
4	8	DOS compatible filename *(required)*
12	3	DOS compatible file extension *(required)*
15	245	Filler bytes

Table D-15
File Processing Abort "6"

Position	Bytes	Description
1	1	Beginning delimiter: "/"
2	1	Command code: "6"
3	1	Directive type: "6"
4	256	Filler bytes

Controller-to-Host Record Layouts

Table D-16
Type 0 - Upload Data Record

Position	Bytes	Description
1	1	Device channel ID
2	1	HHC channel ID: “1”, “4”, or “7”
3	1	Record type: “0”
4	256	Upload data
260	2	CRC bytes if CRC checking is enabled
262	3	Optional padding bytes

Table D-17
Type 1 - End of Session Status Record

Position	Bytes	Description
1	1	Device channel ID: “0”
2	1	Channel ID: “1”, “4”, or “7”
3	1	Record type: “1”
4	1	End of session status code: “0” = Good status. “1” = Abort. Port connection lost. “4” = Disconnect. The session disconnected, probably due to a datalink protocol error. “8” = HHC logic error. The HHC detected a logic error in the file transfer process.
5	4	HHC application status information
9	1	Local port identifier
10	249	Reserved for diagnostic information

Table D-18
Type 2 - Data Request Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “2”

Table D-19
Type 3 - Inactive Status Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “3”

Table D-20
Type 4 - Active Status Record

Position	Bytes	Description
1	1	Device channel ID: “1”
2	1	HHC channel ID: “1”
3	1	Record type = “4”

Table D-21
Type 5 - Activate Request Record

Position	Bytes	Description
1	1	Device channel: “0”
2	1	HHC channel ID: “1”
3	1	Record type: “5”

Table D-21
Type 5 - Activate Request Record

Position	Bytes	Description
4	1	Status code for the previous activation of the port: “0” = Good “1” = Port disconnect due to protocol error “2” = No answer “3” = Modem error “4” = Line protocol error “5” = Bad activation parameters “6” = Error activating port “7” = Auto-answer port timeout “8” = HHC error “9” = Busy “ : ” = Modem reported no dial tone “ ; ” = Modem reported no carrier “ < ” = Call collision. Modem reported ring.
5	4	Reserved for diagnostic information
9	252	Required padding

Table D-22
Type 6 - Special Request Record

Position	Bytes	Description
1	1	Device channel ID: “0”
2	1	Channel ID: “1”
3	1	Record type: “6”
4	1	Special request subtype: “1” = Obtains file processing from the host. The host responds with a file directive or an end-of-data record.

Table D-23
Type 7 - Directive Status Record

Position	Bytes	Description
1	1	Device channel ID: “0”
2	1	Channel ID: “1”
3	1	Record type: “7”
4	1	Status type: “1” - “6”: “1” = File upload “2” = File create or load “3” = File delete “4” = File download “5” = Reserved “6” = File processing short
5	1	Directive status: “0” = Good status “A” = File open error “B” = File read error “C” = File write error “D” = Maximum files exceeded on a create operation “E” = Maximum open files exceeded on a file open operation “F” = File not found “G” = Bad file handle “H” = File in use. A create operation cannot overwrite or delete a file while in use. This occurs if a file boots an HHC when the host attempts to recreate the file. “I” = File not open “J” = File allocation error “K” = End of file “L” = File directory error “M” =Request aborted “O” = File system locked
6	3	Reserved for diagnostic information

Initialization Record

Table D-24
System Mode Parameters

Value	Meaning	Description
0	Mode disabled.	All mode parameters set off.
1	Force EBCDIC host.	Controller determines if host is EBCDIC or ASCII if this field is not added in.
4	Transparent mode on all ports.	Applies to EBCDIC hosts. When added in, ASCII or EBCDIC data translation is disabled on all ports. Data translation can be enabled or disabled for individual ports.
8	Boot 4000 Series HHCs in a LAN connected to the controller.	Enabled when a LAN port is installed and a list of boot files exists on the controller's system disk.
16	Debug mode.	Reserved for Norand diagnostic use.
32	2-byte channel option.	When added in, all records sent to the host are preceded by a 2-byte channel identifier. The 2-byte channel option is required if any of the controllers in the network are remote.
64	Forwards ID records from remote controller to the host.	Must be *on* if the 2-byte channel option is enabled.
128	Controller file processing.	When added in, the host is prompted with file requests at the beginning of a host-controller communications session.
256	Forwards port activation requests from remote controllers to the host.	Enable only if necessary for the host computer to send activation parameters to ports on remote controllers. Normally, the host does not need to see activation requests from remote controllers, unless the remote controller is directly cabled to a host controller directly connected to the host computer.

Table D-24
System Mode Parameters

Value	Meaning	Description
512	Send Local controller ID on power up to host.	When added in, the local controller begins a communications session with the host by sending an ID record. This host then responds with an initialization record. Should be disabled, as it prohibits the host from using initialization record parameters to tailor the host-controller interface. When enabled, the EBCDIC host option must be set for EBCDIC host computers.
1024	Force auto-answer for remote controllers.	When enabled, a remote controller always answers the phone, even when no HHCs are ready to communicate. If not enabled, a remote controller will not answer the phone unless an HHC is ready.
2048	The secondary controller is direct-connected to the primary.	Applies to remote controllers. If enabled, a controller connected to an HHC port on a host controller always maintains the link to the host controller.
4096	Deactivate Request ON *(ASYNC only)*	If *on*, the controller periodically sends a deactivate request to the host for each enabled auto-answer port not currently active, such as a port waiting for a phone call. The host can then time window port configurations. Not useful with the 4985.
8192	Cyclic Redundancy Check (CRC) ON. *(ASYNC only)*	If *on*, the controller appends two CRC characters to the end of the upload data (type 0) records. Host appends two characters to the end of download data blocks (type 0). See Appendix C.

Table D-25
Host Parameters

Parameter	Default	Description
-A[mode]	0	1 to 5 digit numeric field setting mode for host port, such as -A0.
-C[parity]	0	1 digit numeric field setting parity and number of data bits. Valid values are: "0" = No parity/8 data bits(-C0) "1" = Odd parity/7 data bits (-C1) "2" = Even parity/7 data bits (-C2)
-D[stopbits]	1	1 digit numeric field setting stop bits. Valid values are: "1" = One stop bit (-D1) "2" = Two stop bits (-D2) "3" = Three stop bits (-D3)
-E[delimiter]	13	1- to 2-digit numeric field specifying collating sequence number of character marking end of all records sent from controller to host. A delimiter of "0" specifies no end-of-record delimiter.
-F[block size]	0	Numeric field ranging from 258 to 1029; specifies the minimum controller-to-host record block size. Does not include the end-of-record delimiter character. Records shorter than the block size are padded to the block size and the delimiter character is appended. Forces the controller to send fixed-length blocks to the host.
-G[pad char]	32	1- to 3-digit numeric field specifying the characters that pad short blocks when a minimum block size is specified.
-H[ready char]	63	1- to 3-digit field specifying the host "ready" directive.

Table D-25
Host Parameters

Parameter	Default	Description
-I[read timeout]	30	2- or 3-digit numeric field specifying the maximum seconds the controller waits for a response from host. If no response received within set time, the controller will reset and await an initialization record from the host. Valid values are 10 to 120.
-J[host-delay]	0	1- or 2-digit numeric field specifying delay in hundredth of seconds before controller sends next message to host after receiving a ready character from host. Valid values are 0 to 50.
-K[char-gap]	10	1- or 2-digit numeric field specifying maximum intercharacter gap time in whole seconds allowed in a block of data sent from host. Valid values are 9 to 40.
-L[cts-gap]	5	1- to 3-digit numeric field specifying maximum time in tenths of seconds the controller waits for CTS after RTS. Valid values are 0 to 100.

Activation Parameters

Table D-26
Type 4 - LAN Parameters

Parameter	Default	Description
-C[channels]	3 channels	Specifies number of HHCs that communicate concurrently. Valid values are 1, 2, or 3 (-C3).
-T[timeout]	60 seconds	Specifies a “no activity” timeout period in seconds. If timeout period expires, port deactivates. Valid values are from 20 to 120 (-T60).

Glossary

ADCCP

Advanced Data Communications Control Procedures. The U.S. Federal Standard communications protocol. The American National Standards Institute version of a bit-oriented synchronous data link protocol.

APU

Auxiliary Power Unit.

ASYNC

Asynchronous communications. A method of transmitting data using an external clocking source (the transmitted characters are preceded by a start bit and followed by a stop bit).

BISYNC

BInary SYNChronous communications. A method of transmitting data in which the transmission of a character is marked by a drop or rise in the signal. An IBM defined, byte-controlled communications protocol, using control characters and synchronized transmission of binary coded data.

cfg str

Configuration string which refers to a logical data channel. A port may contain one or more channels. Data for any given HHC is contiguous on a channel. The port on a controller may contain up to three channels.

Channel

Refers to a logical data channel. A port may contain one or more channels. Data for any given HHC is contiguous on a channel. Each of the remote ports on a controller may contain up to three channels apiece.

CRC

Cyclic Redundancy Check. An error detection scheme. A number is derived from the data that will be transmitted. By recalculating the CRC at the remote end and comparing it to the value originally transmitted, the receiving node can detect errors.

CTS

Clear To Send. A time delay inserted after a data HHC equipment requests to send data to allow the modem to turn carrier on and establish equalization and synchronization. Also known as RTS-CTS delay and turnaround delay.

HHC

NORAND® Hand-Held Computer.

HHC or Remote Port

An ADCCP, TTY, or NPCP port which can connect to HHC or remote controllers. Data channels on HHC ports are multiplexed onto the host port.

Host

A customer's host computer.

Host Controller

A 4980 Controller which is functioning as a host to a secondary controller.

Host Port

A controller always has one host port which can connect to a host or a 4980 Network Controller.

LAN

Local Area Network. A LAN connects computers and peripherals in a limited area, such as a building or office. RS-485 interface for NO-RAND HHC LAN.

Local Controller

A controller which is directly connected to a host computer. A local controller uses either secondary ADCCP or ASYNC on its host port.

Lockbox

A 4-slot communications rack, used for NORAND TTY/ADCCP communications with 101/121/141 HHCs.

Logical Port

The host port on the controller is logically numbered 0. Port 0 is always the host port.

MQL

Multi-Quad Lockbox. Multi-quad lockboxes are used for ADCCP communications with 121/141 HHCs.

MSD

Norand Mobile Systems Division.

NPCP

Norand Portable Communications Protocol. NPCP provides session, network, and datalink services for NORAND HHCs in the NORAND LAN environment

Physical Port

Equates to port connector. The controller has four internal physical ports (“A”, “B”, “C”, and “D”). It also has six connectors (“A”, “B”, “C”, “D”, “LAN1”, and “LAN2”). Ports “B”, “C”, and “D” are permanently attached to the 15-pin connectors labeled “B”, “C”, and “D” respectively on the back of the controller. Port “A” is attached to either the 15-pin RS-232 connector labeled “A” or to both of the two 9-pin RS-485 connectors labeled “LAN1” and “LAN2”. Port “A” is attached to the RS-485 connectors when it is configured as a LAN port, otherwise it is attached to the RS-232 connector labeled “A.”

Port

The physical hardware communication port.

Primary ADCCP

Connects a remote port to secondary ADCCP HHCs, including 121/141 HHCs and remote connectors.

RTS

Request To Send. A modem control signal on a standard RS-232-C connector that puts the modem in originate mode to start sending data.

Remote Controller

The controller which is logically farthest away from the host computer when two controllers are connected. A remote controller uses secondary ADCCP on its host port.

Secondary ADCCP

Protocol on host port of a remote 4980 or the 4985 Controller and communicates to a Primary ADCCP remote port on another controller.

TCOM or Telecom

Telecommunications.

Terminal

Circuit terminating device such as a HHC.

TTY

NORAND 2-way TTY asynchronous data link protocol.

INDEX

NOTE:

This index covers all topics. Numbers in italics are figures. Numbers in bold are tables. Index entries in UPPERCASE letters are files.