



Wireless Network Access Server
USER'S GUIDE

P/N: 961-051-006
Revision C
June 1999

" NOTICE

The information contained herein is proprietary and is provided solely for the purpose of allowing customers to operate and service Intermec manufactured equipment and is not to be released, reproduced, or used for any other purpose without written permission of Intermec.

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

ANTARES, INTERMEC, NORAND, NOR*WARE, PEN*KEY, ROUTEPOWER, TRAKKER, and TRAKKER ANTARES are registered trademarks and ENTERPRISE WIRELESS LAN, INCA, Mobile Framework, MobileLAN, TE 2000, UAP, and UNIVERSAL ACCESS POINT are trademarks of Intermec Technologies Corporation.

© 1994 Intermec Technologies Corporation. All rights reserved.

Acknowledgments

AS/400, *IBM*, *IBM PC*, *Micro Channel*, *PS/2*, and *OS/2* are registered trademarks and *AIX* and *Presentation Manager* are trademarks of International Business Machines Corporation.

DEC, *Digital*, *VAX*, *VMS*, *VT*, *VT131*, *VT220*, *VT330*, *VT340*, and *VT52* are registered trademarks of Digital Equipment Corporation.

Novell and *UNIX* are registered trademarks of Novell Inc.

CONTENTS



SECTION 1

Read This First

Scope	1-1
Purpose	1-1
Notation	1-2
Help Line	1-2
Warranty	1-3
Software Maintenance Agreement	1-3
Telephone Support	1-3
Bulletin Board	1-3
Version Releases	1-4
Purchase of Version Release	1-4
Documentation	1-4
Additional References	1-5

SECTION 2

Introduction

WNAS as Server	2-6
WNAS Features	2-7

SECTION 3

Installation

Overview	3-1
Installation on AIX Platform	3-4
Software Distribution Media	3-4
Installation Procedure	3-5
Installation	3-6
Software Distribution Media	3-6
Installation Procedure	3-6
Installation Example	3-7

SECTION 4

Configuration

Introduction	4-1
Process Flow	4-2
Multiple Instance	4-4
Set Up	4-5
Run Time Configuration	4-6
Ethernet Configuration	4-8
Host to RF Connection	4-8
Ethernet Device	4-9
Server ID	4-9
Debug Level	4-9
Server Host Name	4-10
RF LAN ID	4-10
RS-232 Configuration	4-10
Host to RF Connection	4-11
Configure RS-232	4-11
Server ID	4-11
Verbosity Level for Debug Mode	4-11
Send Reset at Start-up	4-12
RS-232 Setup	4-12
Application Definition	4-14
Reconnect	4-14
Comment	4-15
Command Line	4-15
Applications Setup Entries	4-15
Menu Definition	4-17
RF Unit Setup	4-20

SECTION 5

Operating Instructions

Introduction	5-1
Starting the WNAS	5-1
Script for Running the WNAS	5-1
Stopping the WNAS	5-2

Application Program Interface	5-3
Open Function	5-4
Write Function	5-4
Read Function	5-5
Close Function	5-6
Console	5-6

SECTION 6

Programming Techniques

Introduction	6-1
Overview	6-1
NORAND [®] Native Mode Syntax for Communicating with Radio Terminals	6-2
Native Mode Responses from the WNAS	6-3
Additional Sequences and Responses	6-4
Using the STDIO Redirection Interface	6-5
C Language Example	6-6

SECTION 7

C Language Source Code Locations	7-1
---	------------

FIGURES

Figure 2-1 Physical Link to Bases	2-2
Figure 2-2 Serial WNAS with RC3250 and RB3000 to UHF Bases	2-3
Figure 2-3 Serial WNAS with RCB4030s to SST Bases ..	2-4
Figure 2-4 WNAS to RC4030E to 6710 to Open Wireless LAN Synthesized UHF Bases	2-5
Figure 2-5 WNAS Ethernet to Access Points	2-5
Figure 4-1 Process Flow	4-3

TABLES

Table 1-1 Related Documentation	1-4
Table 2-1 Operating Systems Supported	2-1
Table 2-2 Part Numbers for Operating Systems	2-1
Table 3-1 Distribution Media for WNAS Software	3-2
Table 3-2 WNAS Files Installed from Distribution Media to norand/root Directory	3-2
Table 6-1 Codes in Radio Terminal Number Field	6-3
Table 6-2 Additional Features	6-5

INDEX

Section 1

Read This First

Scope

This *User's Guide* provides detailed information about the INTERMEC® Wireless Network Access Server (WNAS) software. It has these sections:

Section 2	Introduction
Section 3	Installation
Section 4	Configuration
Section 5	Operating Instructions
Section 6	Programming Techniques
Section 7	C Language Source Code Locations (.c files)

Purpose

This *User's Guide* helps experienced personnel connect their host computer to radio data equipment with the NORAND® label. It also shows the basics of communicating with a radio terminal.

- **NOTE:** *We continue to use the NORAND name in this guide rather than Intermec Technologies Corporation. It was retained to eliminate confusion since WNAS operates with wireless devices that still carry the NORAND label.*

Notation

Programming code examples and command line entries are shown in quotation marks within text to differentiate them from normal text. The Enter key on the keyboard is shown as <Enter>.

Help Line

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

Run Time Configuration:

1. Host to RF connection : R (R=RS232, E=Ethernet)
2. Configure RS232.
3. Server id : 0 (0 thru 7)
4. Verbosity level : 0 (0=Silent 1=normal, 2=debug, 3=debug file)
5. Send reset at startup? : 0 (0=no, 1=yes)

CTRL-> Enter option: 1 (E=exit or item number to change)

HELP-> Enter the Host to RF physical connection type.

The configuration menus for the “nui_cfg” program provide help to explain the expected information. After you select a menu option, additional help displays at the bottom of the screen. Notice the message that appears on the last line (begins “HELP->”).

In the preceding example, option 1 was selected and the help line displayed a message about what is expected for the choice. Help lines are available for most menu options. In addition, if you input a value outside the accepted range, the help line reports the accepted range.

Warranty

Intermec warrants, during the first ninety (90) days after shipment, that software will perform in accordance with the user manual or written specification. We do not warrant that it will meet your needs. If software does not perform as warranted, we will, without charge, attempt to correct the software, the user manual or the written specification. Consult the warranty attachment to your purchase agreement or call us for additional details

Software Maintenance Agreement

You may purchase a software maintenance agreement when you purchase the software. The agreement automatically renews annually. A list of basic services follows.

Telephone Support

Staff you designate have toll free telephone access to Intermec support personnel between 8 AM and 5 PM Central time, Monday through Friday excluding Intermec holidays. The service number is 800-755-5505 or 425-356-1799. You are asked to identify the product and the type of support you need before the call is routed to the proper location.

Bulletin Board

You receive toll call access to Intermec's Bulletin Board Service (BBS) to log problem reports or questions. You can also receive version releases for your software under the terms of your agreement with Intermec. Intermec staff respond to BBS requests on the same schedule as telephone support. Access to the service is available 24 hours a day.

Version Releases

Version releases are the normal enhancements and corrections we make to our product. You are entitled to these as part of your maintenance agreement. To receive these, contact your Intermec representative.

Purchase of Version Release

If you do not have a maintenance agreement, you may be able to purchase the version release. Contact your Intermec sales representative.

Documentation

The following documents are referenced throughout this User's Guide. You need the appropriate programmer's guide.

Table 1-1
Related Documentation

Title	Part Number	Notes
Native Terminal Emulation Asynchronous Programmer's Reference Guide	977-047-038	INTERMEC document
VT220/ANSI Terminal Emulation Programmer's Reference Guide	977-047-037	INTERMEC document
5250 Terminal Emulation Programmer's Reference Guide	977-047-039	INTERMEC document
3270 Terminal Emulation Programmer's Reference Guide	977-047-040	INTERMEC document
Radio Frequency Applications Using the Native Command Set Programmer's Guide*	977-052-002	INTERMEC document

Table 1-1 (continued)
Related Documentation

Title	Part Number	Notes
RC3240/RC3250 Network Controller User's Guide	961-047-040	INTERMEC document
6710 Access Point User's Guide	961-047-081	INTERMEC document
21XX Universal Access Point User's Manual	067150-002	INTERMEC document
RC4030E Gateway User's Guide	961-047-087	INTERMEC document
RCB4030 Spread Spectrum Base and Base/Controller User's Guide	961-047-075	INTERMEC document
UNIX Operating System Programmer's Manual	See Table 2-1	IBM, SUN, or HP document

*This guide was last updated in 1993. It describes RT1000, RT1100, RT1201, RT1210, RT2210, RT3210, RT3310, RT3410, and RT5910 terminals. More recent 1700 terminals and any product having a four-digit number beginning with 6 are not discussed.

Additional References

The Application Development Kit has tools to write host applications for the radio terminals.

Application Development Kit User's Guide P/N 961-051-002

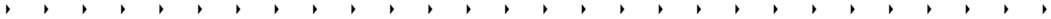
Application Development Kit Programmer's Reference Manual Volume A P/N 977-051-004

Application Development Kit Programmer's Reference Manual Volume B P/N 977-051-005

Notes

Section 2

Introduction



The WNAS is a software component of a radio data network which uses combinations of user-supplied hardware and hardware with the NORAND or INTERMEC label. As shown you supply the host computer. The WNAS software is installed, configured, run, and programmed as necessary. WNAS software is compatible with several common computer operating systems (see Table 2-1). You purchase the software for your operating system.

Table 2-1
Operating Systems Supported

Operating System	Company Name	Version
IBM AIX (RS6000)	International Business Machines	Release 3.2 to 4.X
HP – UX (Series 800)	Hewlett Packard	Release 10.02 to 11.3
SunOS	Sun	Version 5.3 to 5.5.1
Solaris	Sun	Version 2.3 to 2.5.1

Table 2-2
Part Numbers for Operating Systems

Operating System	Part Number
IBM AIX 3.2.5	215-352-001
IBM AIX 4.x.x	215-587-001

Table 2-2 (continued)
Part Numbers for Operating Systems

Operating System	Part Number
HP – UX (Series 800)	215-523-001 (HPUX 10.02–10.1)
HP – UX (Series 800)	215-933-001 (HPUX 10.2–11.3)
SunOS	215-468-001 (SunOS 5.3–5.5.1)
Solaris	215-468-001 (Solaris 2.3–2.5.1)

If you need the product for a different operating system, contact your Intermecc Technologies Corporation representative.

Sometimes, a dedicated RS-232 communication COM port on the host computer handles the data transmissions between the host computer and the RF network controller connected to one or more RF radio controller or bases. These devices communicate with the radio terminals over the radio network. Multi-tasking operation of the application program by the host computer lets it communicate with one or more of the radio terminals.

The following figures show some typical installations using a COM port on the host.

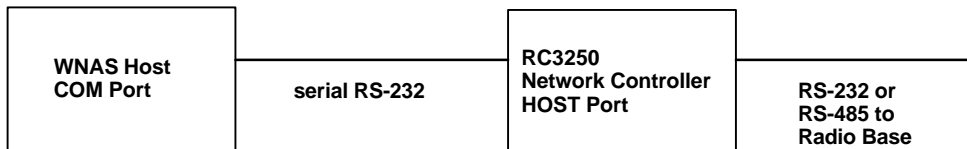


Figure 2-1
Physical Link to Base

In Figure 2-1 WNAS runs on a host that talks to the radio network through a serial connection to the HOST port on an RC3250 Network Controller. The controller may connect to radio bases by RS-232 or RS-485 serial interfaces.

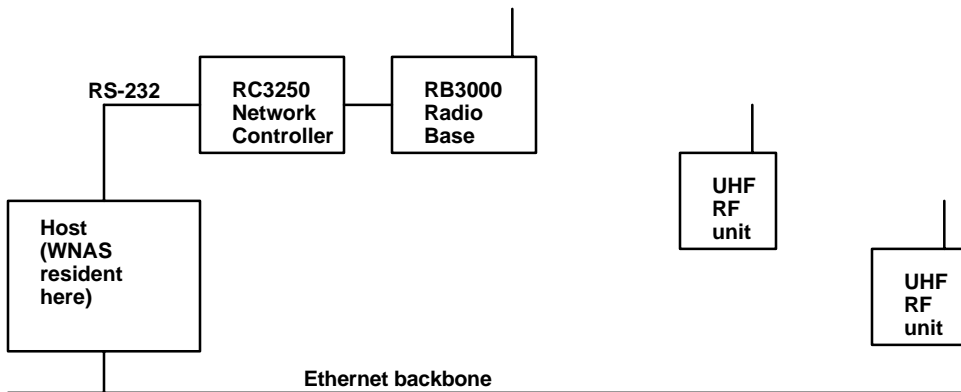


Figure 2-2
Serial WNAS with RC3250 and RB3000

In Figure 2-2 WNAS uses a serial connection to an RC3250 Network Controller. The controller sends and receives its radio messages via the RB3000 Radio Base. The base communicates with radio terminals with UHF radios.

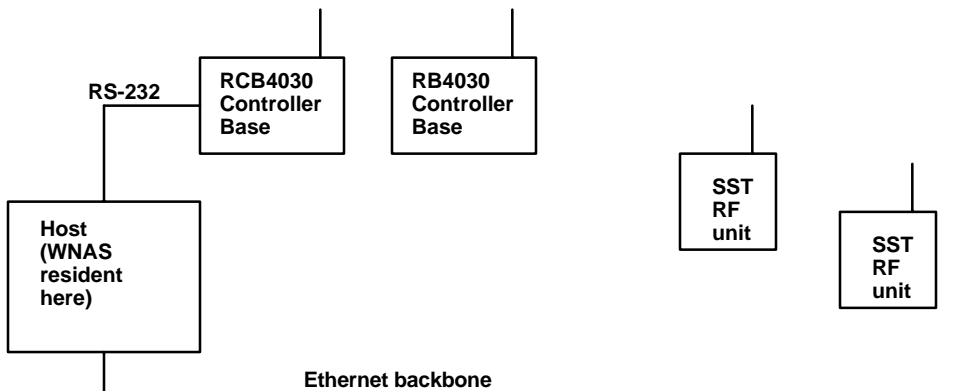


Figure 2-3
Serial WNAS with RCB4030 and RB4030

In Figure 2-3 WNAS uses a serial connection to an RCB4030 Radio Controller/Base. The RCB4030 Base

communicates with the RB4030 via radio. The controllers send and receive radio messages with the SST radio frequency units.

WNAS can also act as a controller and talk directly to an Ethernet base (6710 Access Point with 2.4 GHz radio, RB4030 Radio Base, or 2100 Universal Access Point).

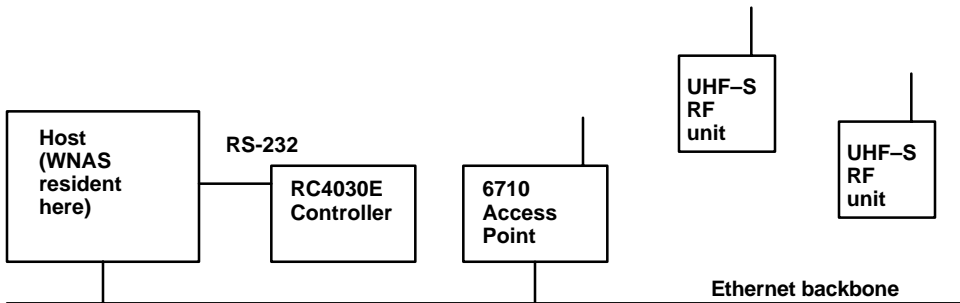


Figure 2-4
WNAS with RC4030E and 6710 on Open Wireless LAN

In Figure 2-4 WNAS uses a serial connection to an RC4030E Radio Controller which can talk via ethernet to a 6710 Access Point. Radio links the synthesized UHF radio frequency units. The controller in the previous figure could be removed, since the controller functions are integrated in the Access Point.

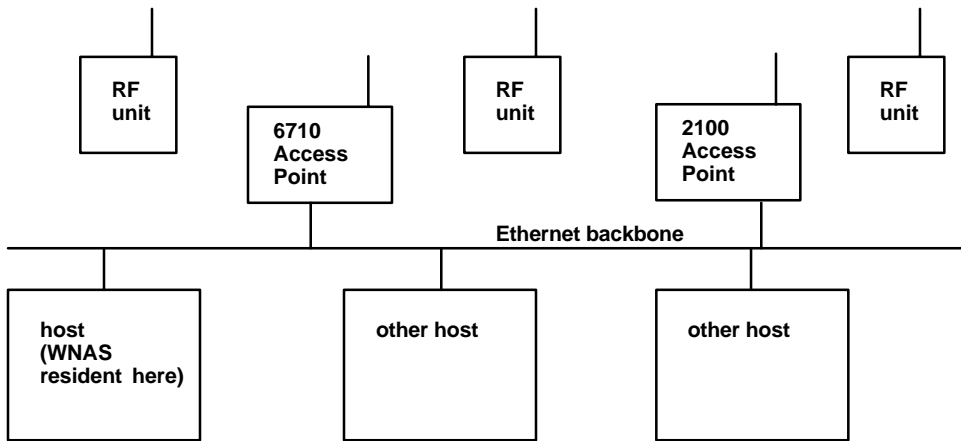


Figure 2-5
WNAS Ethernet to Access Points

For an Ethernet WNAS one or more radio bases reside on the same LAN with a host running WNAS. In Figure 2-5 6710 Access Points attached to the Ethernet backbone communicate to RF units with open wireless LAN 2.4 GHz or 900 MHz radios.

- **NOTE:** *At least one radio base (transceiver) on an Ethernet LAN must have a nonzero root priority.*

WNAS as Server

The WNAS software uses the client-server concept, wherein the host computer is the server, and the remote radio terminals are clients. This allows software developers to write applications for the radio data network which are independent of the host operating system, as long as it is a supported operating system.

The WNAS handles all messages between the application program on the host computer and the RF network

controller. Since all NORAND[®] controllers are supported by the WNAS, you do not need to know controller commands or variations among NORAND controllers.

There is one exception. You need to know controller CMT commands to implement the direct VT220 interface method.

Software applications developed using the WNAS environment are written as though there was only one radio terminal because the multi-tasking capability of the host computer controls specific radio terminals on the RF Radio Network.

The applications program interface (API) techniques which are compatible with the WNAS software are:

- Program Direct Interface
- Program Normal Interface (STDIO Redirection)
- VT220 Interface
- Telnet for 3270, 5250 and VT220 Emulation

Program Direct Interface is written in the C programming language and supports all programming languages that provide the ability to link to a C object file. With this interface, a set of functions may be called from an application program to enable communications with the WNAS. Use this interface with NORAND Native.

Program Normal Interface (STDIO Redirection) has the most convenient way to access the WNAS from an application program. This interface is for use with NORAND Native. It redirects the flow of messages from standard in, standard out, and standard error to the WNAS. You can use standard language commands for displaying and accepting information. This interface supports any programming language or fourth-generation (4GL) language that can communicate with a display screen and a computer keyboard. The language must also allow redirection of “stdin,” “stdout,” and “stderr” programming instructions.

The **VT220 Interface** method provides an interface to existing applications that support VT220 terminal emulation. This method initiates a telnet session for each terminal.

WNAS Features

- Interrupt driven software conserves system resources.
- Efficient use of the host computer CPU for heavily loaded environments.
- Easy to configure.
- Runs in foreground or background mode.
- Supports full data logging.
- Debug mode lets you view dialog on the COM port or Ethernet port between the Wireless Network Access Server and NORAND equipment.
- Supports all NORAND RF network controllers. The direct VT220 interface method supports only a subset comprising the RC3250 Network Controller, RCB4030 Radio Controller/Base, and 6710 Access Point.
- Talk directly to RCB4030 Radio/Base Controller, RB4030 Radio Base, or 6710 or INTERMEC® 2100 Access Point.

Notes

Section 3

Installation



Overview

This section provides step-by-step instructions for copying the Wireless Network Access Server (WNAS) software files from the distribution media onto your hard disk drive. Depending on your host operating system, the INTERMEC[®] distribution media consists of a 1.4 MB, 3.5-inch disk; or a magnetic tape cartridge.

Table 3-1 on page 3-2, summarizes the various media combinations for the compatible operating systems and Table 3-2 through Table 3-4 list all the files on the distribution media that are copied to your hard disk during the installation procedure. The files are organized according to their location in the installation directory, the “source” subdirectory, and the applications or “apps” subdirectory. A brief description of each file is included. The files are automatically placed in their correct subdirectories during the installation process. You simply create a name for the installation directory. In Table 3-2 through Table 3-4, we assume that the installation directory is “norand”. See *Section 7 C Language Source Code Locations* for the location of examples.

Table 3-1
Distribution Media for WNAS Software

Operating System	Distribution Media	Media Format
UNIX	1.4 MB disk or 4 mm tape cartridge (HP)	tar image

Table 3-2
WNAS Files Installed from Distribution Media to norand/root Directory

File Name	Description
nui_serv	WNAS executable
nui_comm	WNAS communication handler (RS-232)
nui_mgr	WNAS communications handler (Ethernet)
nui_irun	WNAS VT220 interface program
nui_pps	WNAS protocol stack handler
nui_stop	WNAS stop program
nui_tran	STDIO translation executable
nuif_adb,nuif_cfg, nuif_mdb,nuif_ude, nuif_idb	Configuration files for WNAS
nui_eth	WNAS Ethernet handler
ethlog	Ethernet log file
starteth.sh	Command script to run the Ethernet interface (must be run as 'root')
stopeth.sh	Script to stop Ethernet driver (must be 'root')

Table 3-2 (continued)
WNAS Files Installed from Distribution Media to norand/root Directory

File Name	Description
nui.sh	Command script to run WNAS
nuiSTOP.sh	Script to stop WNAS

Table 3-3
WNAS Files Installed from Distribution Media to norand/source Subdirectory

File Name	Description	Notes
Makefile	Script to compile DIRECT, STDIO and VT220 examples	ASCII
nui_api.c	DIRECT interface example	
nui_api.o	DIRECT interface example	
nui_ctst.c	VT220 sample source file	
nui_func.o	Program provides DIRECT interface as object file	
nui_func.h	API header file	
nui_lib.h	Header file for nui_lib.o	
nui_lib.o	WNAS library object	
nui_ptst.c	STDIO interface example	ASCII
nui_ptst	Executable STDIO interface	
nui_test.c	Direct interface example program	ASCII
nui_test	API executable example	

Table 3-3 (continued)
WNAS Files Installed from Distribution Media to
 norand/source **Subdirectory**

File Name	Description	Notes
nui_test.o	API object file example	
nui_time.c	VT220 sample source file	

Table 3-4
WNAS Files Installed from Distribution Media to norand/apps
Subdirectory

File Name	Description
nui_test	Example of DIRECT interface executable
nui_ptst	Example of STDIO interface executable
nui_ctst	VT220 sample executable
nui_time	VT220 sample executable

A subdirectory called “norand/pipes” contains data pipes used by the WNAS.

Installation on AIX Platform

These instructions describe installation for AIX version 4.X. Check for a text file of these instructions on the release disk. It may have been updated since this guide was printed.

Software Distribution Media

Media is a 1.4 MB, 3.5-inch disk, or download a self-extracting image file from our BBS.

Installation Procedure

► **NOTE:**

This procedure describes installation to a directory called /usr/norand. Your directory does not have to have this name.

1. Log into the AIX 4.X.X machine and create a directory for the software. As “root,” issue the following command making a directory named “/usr/norand”.
`mkdir /usr/norand`
2. Change directory to the newly created directory where the software will reside.
`cd /usr/norand`
3. Copy this software to the AIX machine.

DOS Disk: If copying the files from a DOS disk, follow these steps.

- a. Issue this command. It copies the file on the disk to a file called “nui605.tar.z” on the RS6000.
`dosread NUI605.z nui605.tar.z`
- b. Uncompress the file using this command:
`uncompress nui605.tar`
or
`compress -d nui605.tar`
- c. Extract the files from the “tar” archive using:
`tar -vxf nui605.tar`

UNIX Tape Archive Disk: If copying from a UNIX “tar” disk, issue the extraction command. Substitute the name for your disk drive in place of “/dev/fd0” in the example if using a different drive.

```
tar -vxf /dev/fd0
```

4. Edit the file “/etc/pse.conf” and uncomment the line with “/dev/dlpi/en”.

EXAMPLE:

Type “vi /etc/pse.conf” and <Enter>. Type the string “//dev/dlpi/en” to search for it within the file.

If a “#” appears at the beginning of the line, delete it.
Cursor movement: Use <h> to back up a character, <l> to move forward a character, <j> to move down a line, <k> to move up a line.

When positioned on the character to delete press <x>.

5. Save the file. Type <Z> <Z>. Must be capital Zs.
6. Run “strload”. Simply type “strload” at the operating system prompt and press <Enter>.
7. Configure our software. Execute “nui_cfg” by typing the name from within the directory where the software is installed.

Use the “nui_cfg” application to verify settings match those needed by the access points and radio terminals.

Installation on Other Platforms

Software Distribution Media

A 1.4 MB, 3.5-inch disk for all supported UNIX platforms or a 4 mm tape cartridge (HP-UX only).

Installation Procedure

To install the WNAS on a UNIX platform, proceed as follows:

1. Logon so that you have access to the correct floppy drive or tape device and can create a subdirectory.
2. Insert the 3.5-inch disk into the floppy drive or insert the tape into the tape drive.
3. Create a directory for the WNAS software and make that directory current.
4. Issue the UNIX “tar” command to copy the WNAS files from the 3.5-inch disk or tape to the current directory on your hard drive.

Installation Example

In the following installation screen dialog, the “\$” represents your system prompt.

Make sure that the WNAS distribution disk is in floppy drive (fd0); otherwise change the “tar” command to access the correct floppy drive.

SCREEN DIALOG	MEANING
login id: mylogin<Enter>	Prompt to enter your login ID.
password: mypassword<Enter>	Prompt to enter your password.
\$ mkdir /usr/norand<Enter>	Command to make a directory called “norand”.
\$ cd /usr/norand<Enter>	Command to change to the “norand” directory.
\$ tar -xvf /dev/fd0/<Enter>	Command to extract files from floppy drive 0 and place them in the current directory (norand).
\$	New UNIX prompt.

When the extract is finished, check the current directory for the existence of all the files listed in Table 3-2 through Table 3-4. A “tar” extract example for a tape device is as follows:

```
tar -xvf /dev/rmt0
```

Notes

Section 4

Configuration



Introduction

This section provides examples of screen displays from the “nui_cfg” program. These screens show the various system configurations you need for the following:

- Runtime Setup
- Application Definition
- Menu Setup
- RF Unit Setup

You need to configure all these options to successfully communicate with a terminal.

Runtime setup defines the primary communication connection between the application and the wireless network. This includes the host name, server ID, and RF LAN ID.

Application definition links a number to the application’s command line for each application. It also controls which instance of the application talks to which terminals and whether terminals can reconnect if the application is still active after a connection is restored. You use the application number in menu definition to present the application to terminals.

Use menu definition to construct menus for the terminals. You can define up to 98 menus. For each menu, you supply

menu text and applications to correspond with up to 10 menu choices per terminal display.

RF Unit Setup configures each active unit. Unit information includes: its type (native or other); the menu number WNAS supplies the terminal; the IP host name, and port the unit wishes to connect to; and user ID and password (if desired or needed for logon purposes). Two additional prompts provide an easy way to copy one unit's set up to many other units.

Process Flow

Before configuring the software, it helps to know the relationship between your applications and processes that communicate with NORAND[®] radio bases, access points, controllers, or multiplexers.

See Figure 4-1 on page 4-3 for the process flow. In the figure, program names are bold. Numbers in parentheses show the number of processes. Double arrows marked "IPC" are InterProcess Communication. Commonly, you use only one of the three methods to interact with the user application.

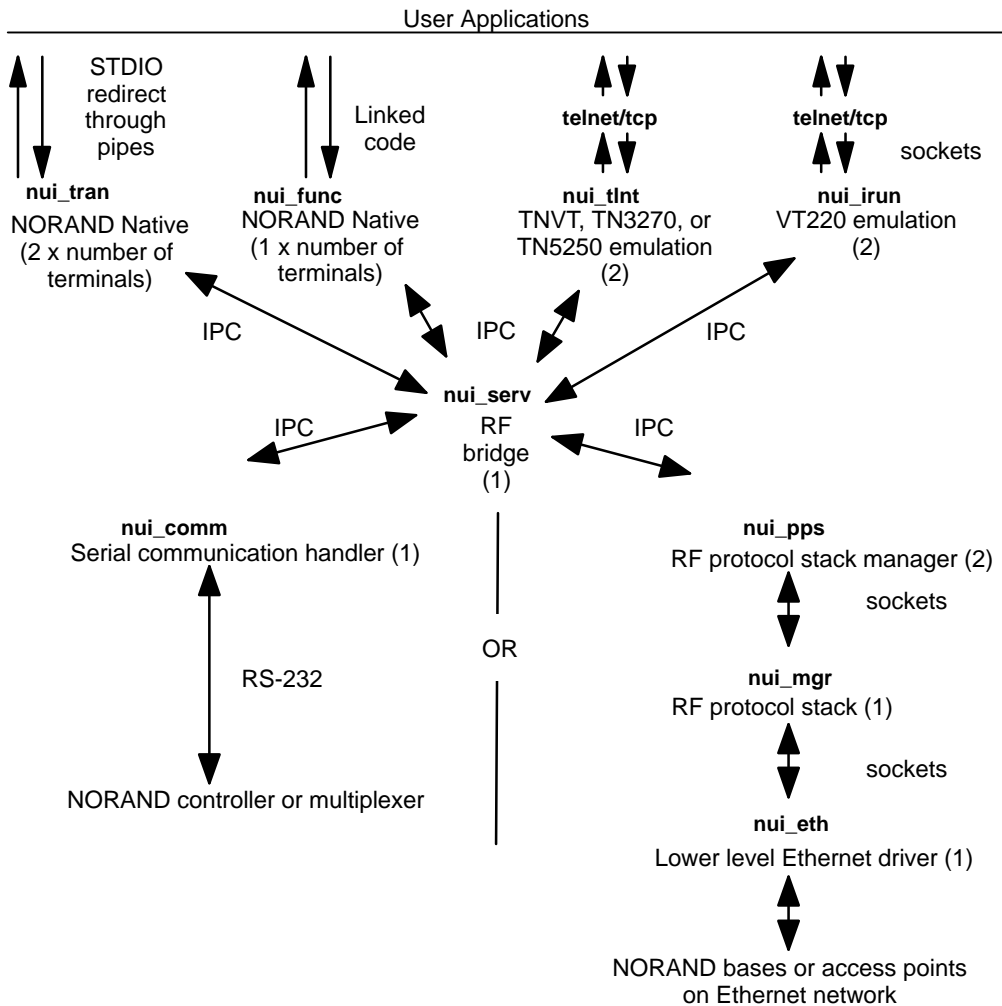


Figure 4-1
Process Flow

Multiple Instance

WNAS provides multiple instance support. With multiple instance, up to eight copies of WNAS software can run concurrently on a single host. This means a single host can theoretically handle $8 \times 127 = 1016$ terminals subject to processing limitations in the host. Current generation work stations have a practical limitation of 300–400 terminals.

Each instance of WNAS must have the same host name, for example HOST and an instance number. WNAS uses Server ID and RF LAN ID to identify the instance. The host name field in the terminal menus must specify both the host name and the instance number. For example, host named HOST and instance one would be identified, HOST 1. Use a single space to separate host name and instance name.

Assign instances in the range 0–7 on the Run Time Configuration menu, option 3, Server ID and RF LAN ID, option 6. The digit in both options must be the same. Use the same location for assigning host names (See page 4-8.).

► **NOTE:** *Server ID and RF LAN ID must be identical in each instance you create.*

EXAMPLE: WNAS with two instances. Both instances are assigned the host name “HOST”. The first instance is assigned Server ID and RF LAN ID = 0 in “nui_cfg”. The second instance is assigned Server ID and RF LAN ID = 1.

Four terminals are configured, two for each instance.

```
Host Name HOST 0   Terminal Number 3
Host Name HOST 0   Terminal Number 9
Host Name HOST 1   Terminal Number 12
Host Name HOST 1   Terminal Number 6
```

When running multiple instances using the Ethernet interface and the same Ethernet device, start only one

instance of Ethernet shell (starteth.sh). All instances will start their own user interface shell (nui.sh). Before starting the interface shell programs, modify them so that their RBPIPES and NMPIPES variables point to the same directories as defined in the start ethernet shell (starteth.sh) that you executed.

Set Up

Use the following steps to set up the WNAS.

1. With the host computer displaying the Wireless Network Access Server (WNAS) directory system prompt, type the following:

```
nui_cfg <Enter>
```
2. The top level menu displays Runtime Setup, Application Definition, Menu Setup, and RF Unit Setup selections.
3. Select the option you want and press the <Enter> key.
4. The corresponding menu displays as shown in the examples that follow.
5. When you finish using a menu, press the <E> key then the <Return> key to exit and return to the top-level menu. Generally, the software prompts you to confirm that you want to save changes.

Select another menu or exit from the main menu.

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

1. Runtime Setup
2. Application Definition
3. Menu Setup
4. RF Unit Setup

CTRL-> Enter option: _ (E=exit or menu item number)

From the main menu select '1' to configure run time options such as RS-232 parameters. Select '2' to define applications to be run from the hand-held terminal. Choose option '3' to configure the menu that you want a terminal to display or an application that opens when it powers up. Option '4' is terminal specific configuration, for example, which menu to use for a specific terminal number.

The following pages describe each of these options in detail.

Run Time Configuration

Choose Runtime Setup from the configuration menu to begin run-time configuration.

Runtime setup defines the primary communication connection between the WNAS and the wireless network. It may be an Ethernet or RS-232 connection. The information you supply varies slightly depending on the connection method.

RS-232 connectivity is used when WNAS talks to a NORAND or INTERMEC® Controller via a serial port on your UNIX machine. For RS-232, you must supply the port device name, callsign only for UHF radios, baud rate,

parity, data bits, then activate all the wireless station numbers you expect to use. WNAS supports up to 8 instances. Each instance requires a separate serial port and NORAND Controller. Then you set the verbosity for debug information. Finally, you define whether WNAS sends a reset command to the controller on startup.

Port is the name of the serial port on which the controller is connected. The other settings (i.e. baud rate) must match your controller. You must disable “getty” on the serial port you are using. The callsign provides information UHF controllers must periodically broadcast. The list of enabled terminals sets parameters for polling the wireless network.

For an Ethernet connection, you record similar information. You supply the Ethernet device name, identify the instances for multiple instance support, set the debug level, supply the server’s host name, and identify which of 8 RF LAN IDs to use. At least one base or access point must be a root candidate.

The Ethernet device tells WNAS which device to use when communicating with the network. Server ID combines with the server host name and RF LAN ID number to uniquely identify which instance and LAN this configuration applies to.

If using the older 902 GHz SST radio technology with 4030E Bases, set LAN ID and instance to match the LAN ID of your bases. LAN ID is not used in conjunction with other radio technologies and should be set to match instance.

Ethernet Configuration

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

Run Time Configuration:

1. Host to RF connection : E (R=RS232, E=Ethernet)
2. Ethernet device : /dev/dlpi/en0
3. Server id : 0 (0 thru 7)
4. Debug level : 0 (0=normal, 1=monitor, 2=debug, 3=debug file)
5. Server Host name ... : HOST
6. RF lan id : 0 (0 thru 7)

CTRL-> Enter option: _ (E=exit or item number to change)

HELP->

Run Time Configuration shows a system using the WNAS to talk directly to Ethernet radio bases. The Ethernet device on this host is “/dev/dlpi/en0”. This configuration uses Server ID 0 and Server Host name “HOST.” For networks with older SST radios, the LAN ID setting must match the Ethernet bases and the radio terminals. The Host name must be set in SST and open wireless LAN systems. It must also match in the radio terminals. Ethernet is the default RF connection. You must also supply port device, debug, and LAN ID information. Server ID applies only if you run multiple instances of the application. Server ID and LAN ID must match.

► **NOTE:**

The Host name here does not refer to your UNIX computer's host name. It is, instead, the name of the RF host for the radio terminals.

Host to RF Connection

Run time options designate either an Ethernet setup or an RS-232 setup for the host to RF connection. Choose “E” for option “1” to configure for Ethernet.

Ethernet Device

Supply the Ethernet device name that is the port used by the application program to communicate with the RF network. If you do not know the device name, consult the “read_me” file that came with this software. It often explains how to determine the device name.

Server ID

Use option 3 if you are running more than one instance per host. Each instance must be assigned a number from 0–7. You then address radio terminals by a combination of host name and server ID number. Each server can address up to 127 terminals. Multiple instance support is available on the operating systems modeled on UNIX.

If using a server ID greater than 0, your host name on the terminals must be set to show both the host name and the server ID. Server ID and RF LAN ID must be the same. For example host name of HOST and server ID of 3 would appear on the terminal as “HOST 3”.

If your WNAS version does not support multiple instances, leave server ID set to 0.

Debug Level

Option 4 configures debug mode. “Normal” runs the WNAS without displaying any information. “Monitor” displays some messages to the console while the WNAS is running. “Debug” displays data coming from the terminals. “Debug file” writes the same information shown on the console to a file called “nuiserv.dbg”. Debug mode can be useful during development, to get timings and to view data being transmitted and received.

► **NOTE:**

Running in “debug” mode can slow the WNAS and use a lot of disk space if the data is routed to a file for long periods of time. “Debug file” displays the debug data to the console and also writes the data to a file called “nuiserv.dbg”.

Server Host Name

Supply a host name for this server. In most cases “HOST” is sufficient.

RF LAN ID

You may run up to eight copies of this software per machine without purchasing additional licenses. To run multiple copies, simply copy all software in the server directory and subdirectories to another directory, reconfigure using a different server ID and matching LAN ID and change any other options needed (i.e., different host to RF port). Then specify which RF LAN ID that each copy controls.

Use the same RF host for all instances. Ensure the Server ID and RF LAN ID are the same. Be sure to configure terminals’ host names correctly, for example “HOST 0” for instance 0, “HOST 1” for instance 1, etc.

RS-232 Configuration

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

Run Time Configuration:

1. Host to RF connection : R (R=RS232, E=Ethernet)
2. Configure RS232.
3. Server id : 0 (0 thru 7)
4. Verbosity level : 0 (0=silent 1=normal, 2=debug, 3=debug file)
5. Send reset at startup? : 0 (0=no, 1=yes)

CTRL-> Enter option: _ (E=exit or item number to change)
HELP->

Host to RF Connection

Run time options designate either an Ethernet setup or an RS-232 setup for the host to RF connection. The preceding shows an RS-232 setup screen. Choose option 1 and type “r” to configure for RS-232.

Configure RS-232

Choose option 2 to configure RS-232 settings to communicate with your serial controller. This selection opens another menu that is described following the current choices under the topic “Configure Communication Port for Serial RS-232” beginning on page 4-12.

Server ID

Use option 3 if you are running more than one server per host machine. You may run up to eight copies of this software per machine without purchasing additional licenses. To run multiple copies, simply copy all software in the server directory and subdirectories to another directory, reconfigure using a different server ID and change any other options needed (i.e., different RS-232 port).

Verbosity Level for Debug Mode

Option 4 configures debug mode. You tell the WNAS how verbose to be with debug information. “Silent” runs the WNAS without displaying any information. “Normal” displays some controller or access point messages to the console while the WNAS is running. “Debug” displays data coming from the terminals. This mode can be useful during development, to get timings and to view data being transmitted and received. “Debug file” displays the debug data to the console and also writes the data to a file called “nui_serv.dbg”.

- **NOTE:** *Running in “debug” mode can slow the WNAS down and use a lot of disk space if the data is routed to a file for long periods of time.*

tells what type of pseudoterminals this software uses (for example, PTC on an IBM RS6000).

Set the baud rate, parity, and data bits to match the NORAND controller or multiplexer you are using.

► **NOTE:**

To use the VT220 interface, set the WNAS and the RF Network Controller communications to run with No Parity and eight data bits. Also, you must make sure multiple buffering and chaining are enabled in the RF Network Controller.

You may type in your callsign (legacy UHF only) using option 5. This causes the controller to broadcast your callsign at least once every 15 minutes.

Options 6 and 7 enable or disable terminal IDs that the WNAS communicates with. The preceding figure shows terminals 0 through 5 are enabled. In this example if you turn on terminal number 10, it will not communicate with the WNAS.

Simply select option 6 to enable terminals or option 7 to disable them. Then type the first and last terminals numbers in the range you want to change. Your changes are reflected in the lines that follow option 7. A “Y” indicates an enabled terminal, while an “N” represents a disabled terminal.

Having terminals enabled is essential. Correctly disabling unused numbers improves the wireless network’s polling efficiency.

Application Definition

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

Application Definition

1. App number . . . : 3
2. Multiterminal . . : 0 (1=Yes, 0=No)
3. Reconnect : 0 (1=Yes, 0=No)
4. Comment : NATIVE program using API
5. Command line . . : ./apps/nui_test

CTRL-> Enter option: _ (E=exit, N=next, P=Previous or item number to change)
HELP->

Use the Application Definition menu (option '2' from the main menu) to configure the different applications that your terminals use. The preceding example shows that application 3 is not a multiterminal application, does not allow reconnect, and is an application called 'nui_test' located in the './apps' directory. A multiterminal application is an application that can handle all terminals. Common applications are written to communicate with only one terminal.

Reconnect

- **NOTE:** *Reconnect works only for VT220 or Native logon. It does not work for 3270 or 5250 emulation nor telnet VT220 emulation.*

Reconnect should be turned on if you want the terminal to reattach to its previous session. For example, if a terminal powers off then back on, it will reattach to the same process and resume from where it was. If reconnect is turned off,

the previous session will be terminated when a terminal powers back on and a new session is started. Reconnect affect the “nui_irun” program only.

► **NOTE:** *If using reconnect, the terminal's screen is not automatically refreshed.*

Comment

Use the comment line for text that may be helpful to you. These comments are not used by the WNAS.

Command Line

The command line contains the program to be run by the WNAS. “nui_tran” is the STDIO redirector. A filename must follow it on the command line.

This example shows the demonstration program “nui_ptst” (executable STDIO interface “nui_ptst”) in the apps directory. A command line example using the STDIO redirector is:

```
/nui_tran ./apps/nui_ptst
```

The “nui_ptst” program in the “apps” directory is used here. It is an example of an executable “DIRECT” interface program (nui_ptst). NORAND[®] native syntax uses “nui_tran” but not VT220 emulation.

Applications Setup Entries

Application Setup may run two programs on its command line that behave differently. An example is provided for each.

nui_cfg — Wireless Network Access Server Configuration — Modvl 6.03

Application Definition

1. App number . . . : 7
2. Multiterminal . . . : 1 (1=Yes, 0=No)
3. Reconnect : 0 (1=Yes, 0=No)
4. Comment : faster/more efficient VT220
5. Command line . . . : ./nui_irun

CTRL→ Enter option: _ (E=exit, N=next, P=Previous or item number to change)
HELP→

Reconnect works for some connections but not others. It works for VT220 and Native connections. It does not work for any form of telnet connection including 3270, 5250, and VT220.

A command line option “**auto**” that you can add after the program name works in one case and not others. Adding the option automatically runs the program on the command line. This option works only for direct VT220. It is irrelevant for Native, since it does not use log in. It does not work for telnet to 3270, 5250, or VT220 emulation.

Process “nui_irun” runs two total processes regardless of the number of RF units in use. Command line “./nui_irun auto” automatically attempts to connect to the RF unit’s defined IP host. Type application number 6 on the application definition screen for an example of automatic connection.

► **NOTE:** *Each unit also has a process containing the application that is running (for example, telnet).*

nui_cfg — Wireless Network Access Server Configuration — Modvl 6.03

Application Definition

1. App number . . . : 14
2. Multiterminal . . . : 1 (1=Yes, 0=No)
3. Reconnect : 0 (1=Yes, 0=No)
4. Comment : TN3270 or TN5250
5. Command line . . : ./nui_tln

CTRL→ Enter option: _ (E=exit, N=next, P=Previous or item number to change)
HELP→

To connect for telnet to 3270 or 5250 you need a different program. Use “nui_tln” if connecting to a host via telnet 3270 and 5250. You must also ensure the radio terminals are set for “telnet.”

- **NOTE:** *This program does not automatically login or reconnect. Telnet sessions require a user ID and password from the user to complete the connection.*

Menu Definition

By default menu 32 is set up for NORAND Native and menu 63 is set up for VT220 emulation. Terminal numbers 2 and 3 are set up for NORAND Native and all others are set up for VT220 emulation.

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

Menu Definition

1. Menu number : 32 2 Autoexec? . : 0 (0=no, 1=yes)

3. Menu item 1 : 02 4. Menu item 2 .. : 03

5. Menu item 3 : 04 6. Menu item 4 .. : 05

7. Menu item 5 : 8. Menu item 6 .. :

9. Menu item 7 : 10. Menu item 8 .. :

11. Menu text:

1 – STDIO TEST 2 – DIRECT TEST 3 – MULTIAPP 4 – MULTI/RECON

CTRL-> Enter option: _ (E=exit, N=next, P=Previous or item number to change)

HELP->

The Menu Definition screen (option 3 from the main menu) lets you customize menus on a per-terminal basis.

Terminal 0 could have a different menu and different characteristics than terminal 1 and so forth. To create a definition, specify a menu number. This corresponds with a menu number from the RF Unit Setup menu. Determine if you would like this terminal to automatically start a certain application (Autoexec) without seeing the menu text. If you select Autoexec, the user never sees this menu. But, the terminal automatically runs the application number listed in Menu item 1.

Thus the previous example displays as follows on a terminal running NORAND Native emulation:

1 – STDIO TEST 2 – DIRECT TEST 3 – MULTIAPP 4 – MULTI/RECON
--

If your menu does not display correctly, check the Host View size on the radio terminal.

Here is a VT220 example.

```
nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03
```

```
Menu Definition
```

```
1. Menu number : 63      2. Autoexec? .. : 0 (0=no, 1=yes)
```

```
3. Menu item 1  : 06 .      . 4. Menu item 2 .. :07
```

```
5. Menu item 3  :      .      . 6. Menu item 4 .. :
```

```
7. Menu item 5  :      .      . 8. Menu item 6 .. :
```

```
9. Menu item 7  :      .      . 10. Menu item 8 .. :
```

```
11. Menu text:
```

```
^1B[0;0H^1B[2J^0D^0A1 = irun auto^0D^0A2 = nui_irun^0D^0A
```

```
CTRL-> Enter option: _ (E=exit, N=next, P=Previous or item number to change)
```

```
HELP->
```

The menu number is 63. The menu text shows how you can embed VT220 escape sequences in text going to the terminal.

This particular example contains a move to position 0,0 (^1B[0;0H), a clear (^1B[2J), the text '1 = irun auto', and then a carriage-return-line-feed (^0D^0A). The '^' symbol allows you to send nondisplayable sequences to the radio terminal. The two characters after the '^' symbol are the

hex value of the nondisplayable character you want transmitted. So the preceding text displays as follows:

```
1 = irun auto
2 = nui_irun
```

VT220 Emulation interprets text from Menu Text (option 11) for Menu Number 63 as shown at the terminal.

RF Unit Setup

nui_cfg — Wireless Network Access Server Configuration — Modlvl 6.03

RF Unit Setup:

1. Unit number . . . : 2
2. Unit type : 0 (0=native, 1=other)
3. Comment : Native terminal 2
4. Menu Number . . : 32
5. IP Host name . . : localhost
6. Port : 0023
7. User ID :
8. Password :
9. Copy setup from unit ???.
10. Duplicate this setup to unit ??? thru unit ???.

CTRL-> Enter option: _ (E=exit, N=next, P=Previous or item number to change)
HELP->

The RF Unit Setup screen configures each terminal.

This example shows that terminal number 2 is a Native terminal (a radio terminal running NORAND native), and uses menu number 32 (see the Menu Setup). To copy this configuration to other terminal numbers, use option 10.

There may be instances where you do not want all terminals to have the same applications (i.e., terminals 1 through 10 may run inventory, whereas terminals 11 through 20 may run loads). Or you may have NORAND Native emulation terminals running along with NORAND VT220 emulation terminals. Therefore you need two separate menus displayed to the different emulations.

Leave the port number as is unless your host uses a different port for Ethernet telnet. Type the port number. Use 23 for the default port on the remote host.

The user ID and password is for automatic login. It works with direct VT220 **only**. Leave blank if not used.

Password can be 40 characters long. Acceptable characters are numbers, uppercase and lowercase letters, underscores, dashes, spaces, periods, and forward or backslashes.

The text file “nuif_idb” contains the match string that WNAS uses for automatic login.

Use “IP Host name” with VT220 terminal emulation running “nui_iron” and telnet 3270 and telnet 5250 to specify the IP address or name of the host to connect to. It initiates a telnet session to the specified host if selected. Also use “User ID” with VT220 terminal emulation running “nui_iron”. It initiates a “telnet” session to specified host (option 5) as specified user (option 7).

Notes

Section 5

Operating Instructions



Introduction

This section provides instructions for operating the Wireless Network Access Server (WNAS) and its associated hardware in the radio terminal network. It includes a description of the script for running the WNAS and procedures to stop the it.

Starting the WNAS

If you are using the Ethernet interface with the server (connecting directly to RB4030E Bases with no controller), execute the program “starteth.sh” as the root user. Then, as “root” or any user with proper permissions execute “nui.sh”.

- **NOTE:** *If you are using the Ethernet interface, you must start “starteth.sh” then “nui.sh” to start the server, since it uses the environment variable RBPIPES which is set in the “nui.sh” shell script.*

Script for Running the WNAS

The file “nui.sh” should exist in the WNAS directory. Use it to start the WNAS.

```
## SHELL SCRIPT TO RUN WIRELESS NETWORK
## ACCESS SERVER
## IF WE GET AN EXIT CODE OF 2 INDICATING A
## CONTROLLER REBOOT, WE WILL CONTINUE TO
## CYCLE.

RBPIPES=./pipes/
NMPIPES=./pipes/
export RBPIPES
export NMPIPES
ec="2"
while [ "$ec" = "2" ]
do
    rm nb*
    ./nui_serv
    ec=$?
done
```

The preceding script uses the Bourne shell (sh(1)) or the Korn shell (ksh(1)). If your system does not have these shells available, you must modify this script.

In this example, environment variables called RBPIPES and NMPIPES are set, then the error code is set to two using the

```
ec="2"
```

command. A “while” loop is then entered that continues to run “nui_serv” until the error code is not equal to two.

Stopping the WNAS

Use the “nuistop.sh” program in the WNAS directory to stop the WNAS. “Nuistop.sh” requires no parameters. It sends a message telling the WNAS to quit. The WNAS does the following:

1. Notifies all running applications, that the system is going down.

2. Sends a “System Down” message to all radio terminals.
3. Waits for applications to stop. If an application does not stop, the WNAS terminates the application after 15 seconds.
4. Cleans up and releases any operating system resources that the WNAS was using.
5. Stops.

► **NOTE:** *You can also signal the WNAS by sending an interrupt 15. Use this as a last resort, since it will not free all system resources in use, for example the message queue.*

A command to interrupt 15 would look like this:

```
'kill -15 server_pid'
```

If the WNAS terminates abnormally, make sure that all active applications are terminated. See your operating system manual for details on stopping processes.

When running the Ethernet interface, kill “nui_eth” as the root user by running “stopeth.sh”. The other WNAS process should terminate with “nuistop.sh”.

Application Program Interface

The WNAS has a “C” language application program interface (API) with the demonstration program “nui_test.c”. The functions are:

```
int nui_api_open(int mode)
int nui_api_write(int trm,unsigned char *buf,int *cnt)
int nui_api_read(int tim,int *tnum,unsigned char *buf,int *cnt)
int nui_api_close()
int nui_api_con(char *conmsg)
```

We also provide the sources to “nui_api” and “nui_func” (lower level functions called by “nui_api”).

Open Function

The open function “nui_api_open” opens the API for use.

```
int nui_api_open(int mode);
```

Include “nui_func.h”.

The variable “mode” sets the mode in which to open the API. Mode can be:

- NUI_NATIVE which is the normal method
- NUI_BINARY which does not parse to detect if a valid native command exists

Both modes mask nondisplayable values in the following manner: <value> is passed through as ^<hex value>. The “^” tells the radio controller to pass through the next 2 values as hex and not to interpret them. For example, if sending the character with the value 1 (Ctrl-A), the WNAS translates this to ^01. The radio controller may interpret certain nondisplayable characters as valid commands for it (the controller), and interpret it as an SOH (Start Of Header) that it would not pass through to the terminal. See the read and write commands for more differences between NUI_BINARY and NUI_NATIVE.

Returns NUI_BAD if the command failed, else 0.

Write Function

The write function “nui_api_write” writes a string to a radio terminal.

```
int nui_api_write(int trm,unsigned char *buf,int *cnt);
```

Include “nui_func.h”.

The variable “trm” is the specific terminal number to write to if using more than one terminal. If “-1” open writes to the terminal running this application. Valid terminal numbers are 0-126.

The “buf” variable is a null-terminated string to be sent to the terminal. This follows the “WtD” command described in

the *Native Terminal Emulation Asynchronous Programmer's Reference Guide* without the “Wt”. For example, if you wanted to send the command “W000D/Hello”, instead send the string “D/Hello”.

Other commands this works with are:

<u>Command</u>	<u>Set string to</u>
reset	“g”
StD	“eD/<command>”
DtV	“v”

If using the mode NUI_BINARY, only the WtD command works. Only data should be sent, the “D/” is added already. The variable “cnt” is the size of message in bytes to be sent. Each character is a byte.

Returns NUI_BAD if failed, else NUI_GOOD.

Read Function

The read function, “nui_api_read”, reads data from a radio terminal.

```
int nui_api_read(int tim,int *tnum,unsigned char *buf,int *cnt);
Include “nui_func.h”.
```

The variable “tim” is the time in seconds for the read to wait for a response from the terminal. This variable is not currently supported—timeout value is 15 seconds. Valid values would be 0 to 1800.

The variable “tnum” shows the terminal number from which the data was read. Valid values are 0-126, NUI_TIMEOUT if no data was returned within given time, NUI_DEAD if no data was returned within long period of time (30 minutes), or NUI_STOP if WNAS software requested that program be terminated (issued an “nui_stop”). NUI_TIMEOUT and NUI_DEAD can be ignored and another “nui_api_read” called.

The “buf” variable is a buffer to read the incoming string into. If mode is NUI_BINARY, only the incoming data is in

the string. If the mode is NUI_NATIVE, the string starts with an “S” for scanned data or a “K” for keyed data.

The variable “cnt” is the number of bytes read.

Returns NUI_BAD if failed, NUI_BOOT if power up message from the terminal, else NUI_GOOD.

Close Function

The “nui_api_close” function closes the API.

```
int nui_api_close();
```

Include “nui_func.h”.

Returns NUI_BAD if failed, else NUI_GOOD.

Console

The “nui_api_con” function writes a message to the console, but not the radio terminal.

```
int nui_api_con(char *conmsg);
```

Include “nui_func.h”.

The variable “conmsg” contains the message to be written to the console.

Returns NUI_BAD if failed, else NUI_GOOD.

Section 6

Programming Techniques



Introduction

This section provides an overview of the programming methods which host applications can use to communicate with the WNAS; a description of how to use Native Mode Syntax to communicate with the Radio Terminals; a list of Native Mode Responses from the WNAS; and a description of how to use the STDIO redirection interface for each of the platforms.

Overview

Host applications use one of three available methods to communicate with the WNAS as follows:

- STDIO redirection interface
- DIRECT interface
- VT220 telnet method

Examples of all methods are stored in the “source” subdirectory in the WNAS directory (see Table 3-2 through Table 3-4) described in Section 3 on pages 3-2 through 3-4.

If you are using the NATIVE data stream, you should normally use the STDIO interface, but for customized coding, you may use DIRECT.

STDIO and DIRECT methods communicate with the radio terminals using the NORAND Native Mode syntax.

The VT220 method runs existing host applications written for VT220 Terminal types running under UNIX. For the best visibility, you may want to modify your host applications to use only a small part of the screen; otherwise, it may be necessary to scroll to see the full 25-row by 80-column screen.

NORAND[®] Native Mode Syntax for Communicating with Radio Terminals

You may use any “WtD” command specified in the Programmer’s Guide. We strongly recommend that you familiarize yourself with these commands before starting to write any code.

The following Write display command is a sample startup menu.

```
DBCDNL1H2P15/01 – STDIO TEST 02 – DIRECT TEST
```

Notice that this message does not have the “W” at the beginning, nor does it contain the terminal number. The communication driver in the WNAS adds the “W” and the terminal number. If the “W” is included here, the message is rejected.

The remainder of the message, as shown in the preceding example, follows Native syntax rules.

Responses received from the terminals follow the responses listed in the Programmer’s Reference Guide. (See the interface method for more information about the format of incoming messages.)

Syntax checking is not done on messages transmitted through the WNAS to the terminal. Check the response to

an asynchronous command for error notifications from the controller.

Code does not need to be designed to handle multiple terminals if you are running with `multiterminal` set to 0 (Application Setup Menu). As each terminal powers up, the operator is prompted, as part of the start-up message, to enter the number of the application to run. The WNAS looks up the application number specified in the applications section of the configurer (`nui_cfg`) and executes it. If two operators select the same application number, a copy of the application runs for each terminal.

Native Mode Responses from the WNAS

The WNAS sends the following codes in the Radio Terminal number field returned to the application if no data was returned in a read.

Table 6-1
Codes in Radio Terminal Number Field

Code or Message	Meaning to the Application Program
800	No messages are available for the application. The application can perform housekeeping tasks or go back to waiting for a message (same as “NUI_TIMEOUT” that follows).
998	The terminal has not passed a message for a predetermined length of time. The application can either ignore the message or act on it. For example, this code could be returned if the terminal is left unattended with the application running (same as “NUI_DEAD” that follows).

Table 6-1 (continued)
Codes in Radio Terminal Number Field

Code or Message	Meaning to the Application Program
999	The WNAS wants it to terminate. If the application does not terminate within 15 seconds, it is terminated automatically. The application receives this message when the WNAS shuts down or is reset, or if the WNAS detects an error (same as “NUI_STOP” that follows).
NUI_OK	The WNAS completion was normal (equal to -3).
NUI_STOP	The WNAS wants to terminate (same as 999).
NUI_DEAD	The associated RF terminal has not been passed a message for a predetermined length of time (same as 998).
NUI_TIMEOUT	No data available from the application to be read from the WNAS (same as 800).
NUI_BAD	A fatal error occurred which prevents continued execution (equal to -1).
NUI_BOOT	The associated RF terminal is powered on and can be used to reload data into the terminal or it can be ignored (equal to -4).

Program interfaces return the terminal number and the controller or terminal response. If one of these codes is not set, the field contains the number of the terminal.

Additional Sequences and Responses

The following table shows code sequences which can be sent in Native Mode and a description of the normal responses from the Radio Terminal.

Table 6-2
Additional Features

Feature	Description
“e” or “s” = St	Sends the “St” command. NOTE: You need to send from the “D” on when issuing this. For example, if you were transmitting to terminal number 2, sending “eD” would send “s002D”. The next response would be “CO”.
“v” = DtV	Sending a “v” requests the version number of the terminal. That is, sending a “v” issues a DtV command to the terminal. The next message from that terminal should be its version number.
“g” = Gt	Sending a “g” resets the terminal. The next message from that terminal should be a power-up message.

Using the STDIO Redirection Interface

Program is `nui_tran` for standard I/O redirection.

Located in the `WNAS` directory, this file starts applications that use the STDIO redirection interface.

The example file, “`nui_ptst.c`”, is included in the “`server/source`” subdirectory as an application that uses STDIO redirection.

Use this program to communicate with terminals without extensive low-level coding.

This method allows an application, written in any language which is native to the host computer and allows redirection of “`stdin`” and “`stdout`,” to communicate with a NORAND Native Mode Radio Terminal using the Native Mode Syntax.

Special libraries or external functions do not need to be added to the code. The transport layer between the application and the terminal is transparent.

Most languages can communicate with the keyboard or CRT screen. The interface captures the CRT screen output and sends it to the RF terminal screen. Responses from the terminal are routed through the application's keyboard entry facility.

The requirements for using this interface are as follows:

1. The application program must display NORAND Native Mode messages on the host computer screen.
2. The application must accept data from the host computer keyboard.
3. The application (or routine) must not reset the screen nor the keyboard device handlers.

Most programming languages conform to these rules.

C Language Example

The following example shows a program that continually sends the phrase "Hello world" each time anyone presses a key at the terminal and then it waits for any response other than "X" from the terminal before looping. "X" stops the program.

```
char work_buf[80];
main_func() {
    gets(work_buf); /* get power up */
                    /* message */
    hello_func();
    return;
}
void hello_func() {
    while(work_buf[4] != "X")
    {
        puts("DBCL1H1P15/HELLO WORLD."
            "  PRESS A KEY\n");
        gets(work_buf);
    }
    return;
}
```

Notes

Section 7

C Language Source Code Locations



C language code examples are provided in the Wireless Network Access Server “source” directory for each type of I/O supported by the WNAS software. These include:

- › API examples (nui_test)
- › STDIO example (nui_ptst)
- › VT220 examples (nui_ctst & nui_time)

Notes

Index

Numbers in bold are figures.
Numbers in italics are tables.

NUMBERS

- 800 message, radio terminal number field, *6-3*
- 998 message, radio terminal number field, *6-3*
- 999 message, radio terminal number field, *6-4*

A

- AIX, installation procedure, 3-5
- API, 5-3
 - C language example location, 7-1
 - close function, 5-6
 - console, 5-6
 - open function, 5-4
 - read function, 5-4, 5-5
- Application definition, 4-14
- Application Development Kit Programmer's Reference Manual, 1-5
- Application Development Kit User's Guide, 1-5
- Applications setup entries, 4-15
- Autoexec, 4-18
- Automatic execution. *See* Autoexec

B

- Baud rate, 4-13
- Bourne shell, 5-2

C

- C language source code locations, 7-1
- Callsign, 4-13
- Codes in radio terminal number field, *6-3*
- Command line, 4-15
- Comment line, 4-15
- Communication port, for serial RS-232, 4-12
- Configuration, 4-1
- Configure communication port, RS-232, 4-12
- Configure RS-232, 4-11

D

- Data bits, 4-13
- Debug level, 4-9
- DIRECT interface, 4-15, 6-1
- Distribution media for WNAS software, 3-2
- Documentation, related, 1-4
- DtV, *6-5*

E

- Embed escape sequences, 4-19
- Escape sequences, embed, 4-19
- Ethernet device, 4-9
- Ethernet WNAS
 - with 2100 Access Point, **2-5**
 - with 6710 Access Point, **2-5**
 - with RC4030E, **2-4**
- ethlog, 3-2

F

- Features, 2-7
- Figures
 - Ethernet on Open Wireless LAN, **2-4**
 - Ethernet to 2100, **2-5**
 - Ethernet to 6710, **2-5**
 - physical link to base, **2-2**
 - serial WNAS, **2-3**
 - WNAS with RB4030, **2-3**
- Files
 - installed to norand/apps directory, *3-4*
 - installed to norand/source directory, *3-3*
 - installed to root directory, *3-2*

G

- Gt, *6-5*

H

- Help line, 1-2
- Host, methods for talking to the server, 6-1
- Host configuration
 - Ethernet, 4-8
 - RS-232, 4-10
- Host name, 4-8
- Host to RF connection
 - Ethernet connection, 4-8
 - RS-232, 4-11
- HP-UX, *2-1*

I

- IBM AIX, *2-1*
- Installation, 3-1
 - AIX 4 platform, 3-4
 - other platforms, 3-6
 - overview, 3-1
 - software media, 3-2

Installation example, UNIX, 3-7
Installation procedure
 AIX, 3-5
 UNIX, 3-6
Instance, 4-4
Interprocess communication, **4-3**
Introduction, 1-1, 2-1
IP host name, VT220 emulation,
 4-21
IPC. *See* Interprocess
 communication

K

Korn shell, 5-2

M

Makefile, 3-3
Media. *See* Software
 distribution media
Menu definition, 4-17
 VT220 emulation example,
 4-19
Multiple instance, 4-4

N

Native mode
 responses from the server, 6-3
 sequences and responses, 6-4
 syntax for communicating
 with radio terminals, 6-2
Native Programmer's Reference
 Guide, 1-4
norand/pipes directory, 3-4
nui.sh, 3-3
nui_api.c, 3-3
nui_api.o, 3-3
NUI_BAD message, radio
 terminal number field, 6-4
NUI_BOOT message, radio
 terminal number field, 6-4
nui_comm, 3-2, **4-3**
nui_cppts, 3-2
nui_ctst, 3-4, 7-1

nui_ctst.c, 3-3
NUI_DEAD message, radio
 terminal number field, 6-4
nui_eth, 3-2, **4-3**
nui_func, **4-3**
nui_func.h, 3-3
nui_func.o, 3-3
nui_irun, 3-2, **4-3**, 4-16, 4-21
nui_lib.h, 3-3
nui_lib.o, 3-3
nui_mgr, **4-3**
NUI_OK message, radio
 terminal number field, 6-4
nui_pps, 3-2, **4-3**
nui_ptst, 3-3, 3-4, 4-15, 7-1
nui_ptst.c, 3-3
nui_serv, 3-2, **4-3**
nui_stop, 3-2
NUI_STOP message, radio
 terminal number field, 6-4
nui_test, 3-3, 3-4, 4-15, 7-1
nui_test.c, 3-3
nui_test.o, 3-4
nui_time, 3-4, 7-1
nui_time.c, 3-4
NUI_TIMEOUT message, radio
 terminal number field, 6-4
nui_tlnt, **4-3**, 4-17
nui_tran, 3-2, **4-3**, 4-15
nui_tran for STDIO redirection,
 6-5
nuif_adb, 3-2
nuif_cfg, 3-2
nuif_idb, 3-2
nuif_mdb, 3-2
nuif_udb, 3-2
nuistop.sh, 3-3
Number field, radio terminal
 codes, 6-3

O

Operating instructions, 5-1

Operating systems supported
 part number, 2-1
 version, 2-1
Overview, installation, 3-1

P

Parity, 4-13
Process flow, **4-3**
Program
 direct interface, 2-6
 normal interface, 2-6
Programming techniques, 6-1
Purpose, document, 1-1

R

Radio Frequency Applications
 Using the Native
 Command Set
 Programmer's Guide, 1-4
Radio terminal number field
 codes, 6-3
RBPIPES, 5-2
RC3240/RC3250 Network
 Controller User's Guide,
 1-5
RC4030E Gateway User's
 Guide, 1-5
RCB4030 Spread Spectrum
 Base and Base/Controller
 User's Guide, 1-5
Reconnect, 4-16
Reconnect terminal session, 4-14
RF LAN ID, 4-4, 4-10
RF unit setup, 4-20
rlogin, 4-21
RS-232 port configuration, 4-12
Run time configuration, 4-6

S

Scope, document, 1-1
Script, for running the WNAS,
 5-1
Send reset at start-up, 4-12
Serial connection, **2-2**

- Serial port configuration, 4-12
 - Serial WNAS
 - with RB3000, **2-3**
 - with RB4030, **2-3**
 - Server ID, 4-4
 - Ethernet, 4-8
 - Ethernet connection, 4-9
 - RS-232, 4-11
 - Server, WNAS as, 2-5
 - Software distribution media, 3-2
 - AIX 4, 3-4
 - UNIX, 3-6
 - Solaris, version supported, 2-1
 - starteth.sh, 3-2
 - Starting the WNAS, 5-1
 - stdin, 2-6
 - STDIO
 - C language example, 6-6
 - C language example location, 7-1
 - redirection, 2-6
 - redirection interface, 6-1
 - using, 6-5
 - redirector, 4-15
 - stdout, 2-6
 - stopeth.sh, 3-2
 - Stopping the WNAS, 5-2
 - SunOS, version supported, 2-1
- T**
- Telnet/TCP, **4-3**
- U**
- UNIX, kill example, 5-3
 - User ID, VT220 emulation, 4-21
 - Using the STDIO redirection interface, 6-5
- V**
- Verbosity level for debug mode, 4-11
- VT220**
- C language example location, 7-1
 - embedded escape sequence, 4-19
 - emulation example, menu definition, 4-19
 - interface, 2-7
 - telnet method, 6-1
- VT220/ANSI Terminal Emulation Programmer's Reference Guide, 1-4**
- W**
- WNAS
 - as server, 2-5
 - features, 2-7
 - stopping, 5-2
 - Write to display commands, 6-2
 - WtD commands, 6-2

Notes