Intermec
Technologies Corporation

6710 Mobile Bridge

USER'S GUIDE

P/N 961-028-098
Revision A
June 1998

" 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.

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

This publication printed on recycled paper.

Acknowledgments

Portions of this product contain software which is licensed from and is copyrighted by Epilogue Technology Corporation, 1988–1995, all rights reserved.

DECnet and *VT* are registered trademarks of Digital Equipment Corporation.

Ethernet is a trademark of Xerox Corporation.

Hewlett-Packard and *HP* are registered trademarks and *HP OpenView* is a trademark of Hewlett-Packard Company.

Microsoft is a registered trademark of Microsoft Corporation.

Netscape Navigator is a trademark of Netscape Communications Corporation.

Novell and *NetWare* are registered trademarks and *IPX* and *SPX* are trademarks of Novell, Inc.

PC AT is a registered trademark of International Business Machines Corporation.

PROCOMM and *PROCOMM PLUS* are registered trademarks of DataStorm Technologies, Inc.

FCC Computer Compliance

" **NOTICE**

This equipment meets Class B digital device limits per Part 15 of FCC Rules. These limits protect against interference in a residential 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. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning our equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the radio or television receiving antenna.
- Increase the separation between the computer equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the radio or television receiver is connected.
- Consult the dealer or an experienced radio or television technician for help.

Canadian Computer Compliance

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe B respecte toutes les exigences du Reglèment sur le material boilleur du Canada.

Telephone Installation Warning Notices

The following notices apply to equipment that may be connected to telephone lines or systems. For your personal safety, and to protect this equipment from potential electrical or physical damage, do NOT connect equipment to telephone lines or data communication equipment unless the following warnings have been read, understood, and complied with.

- Never install telephone wiring during a lightning storm.
- Never install telephone jacks in wet locations unless the jack is specifically designed for wet locations.
- Never touch uninsulated telephone wires or terminals unless the telephone line has been disconnected at the network interface.
- Use caution when installing or modifying telephone lines.
- Avoid using telephone (other than cordless type) during an electrical storm. There may be a remote risk of electric shock from lightning.
- Do not use the telephone to report a gas leak in the vicinity of the leak.

Installation du téléphone : avertissements

Les avertissements qui suivent s'appliquent à tout équipement qui peut être branché aux lignes ou systèmes téléphoniques. Pour votre sécurité personnelle et pour protéger l'équipement de tout dommage électrique ou physique potentiel, NE PAS brancher un ordinateur tablette électronique ou ses périphériques aux lignes téléphoniques ou équipements avant que les avertissements suivants aient été lus, compris et observés :

- Ne jamais installer de câblage téléphonique pendant un orage électrique.
- Ne jamais installer de prise téléphonique dans un endroit humide à moins que la prise ait été spécifiquement conçue pour être utilisée dans les endroits humides.
- Ne jamais toucher les fils de téléphone ou de l'équipement terminal non isolés à moins que la ligne téléphonique n'ait été débranchée de l'interface réseau.
- User de prudence lors de l'installation ou de la modification de lignes téléphoniques.
- Éviter d'utiliser un téléphone (autre qu'un appareil téléphonique sans fil) pendant un orage électrique. Il pourrait y avoir un faible risque d'électrocution par la foudre.
- Ne pas utiliser le téléphone afin de signaler une fuite de gaz à proximité de la fuite.

B CAUTION: **Intermec Technologies Corporation suggests you buy cables from us to connect with other devices. Our cables are safe, meet FCC rules, and suit our products. Other cables may not be tested. They may cause problems from electrostatic discharge or induced energy. Our warranties do not cover loss, injury, or damage from other cables.**

CONTENTS

SECTION 5

FIGURES

TABLES

Section 1

Introduction

Purpose of This Guide

This user's guide describes the installation, setup, and maintenance of the 6710 Mobile Bridge. This guide covers FLASH version 1.31 or greater and ROM version 1.14 or greater.

Organization

This Preface describes the intended audience for this guide, lists related publications, and tells how to contact the Customer Response Center. Other sections do the following:

- **Section 2, "Features and Functional Overview,"** describes the 6710 Mobile Bridge and how it operates on the RS-485–Ethernet LAN. It also describes hardware components.
- **Section 3, "Installation,"** helps you prepare your site before you install the 6710 Mobile Bridge, and shows how to connect the 6710 Mobile Bridge to the RS-485–Ethernet LAN.

- **Section 4, "Configuration,"** describes how to create a communications session with the 6710 Mobile Bridge, access its FLASH and ROM, and set it up through its configuration menus.
- **Section 5, "Software Download,"** describes file system methodology and the functional characteristics of the software download process.
- **Section 6, "Indicator Lights,"** describes the 6710 Mobile Bridge's indicator lights and contains troubleshooting tips.
- **Appendix A** lists mechanical, electrical, and environmental specifications for the 6710 Mobile Bridge.
- **Appendix B** shows port and cable pin-outs.
- **Appendix C** describes the open wireless LAN Management Information Base (MIB) product for the 6710 Mobile Bridge.

Intended Audiences

Following are the intended audiences for this user's guide:

- Network administrator who is familiar with various types and configurations of computer networks, how they work, and the terminology used when discussing them
- Hardware installer responsible for performing the physical installation of the 6710 Mobile Bridge and any related hardware that might be required

Related Publications

The following publications are available. They include information about hardware and software products related to or used with the 6710 Mobile Bridge and the network on which it operates.

Numbers in parentheses after the title indicate the part number (P/N) for the publication. Contact your Sales Representative for ordering information.

PEN*KEY® Computer User's Guides

User's guides for PEN*KEY computers describe how to set up, operate, and maintain the computers. Specific guides are:

PEN*KEY Model 6100 User's Guide **(P/N 961-028-085)**
PEN*KEY Model 6110 User's Guide **(P/N 961-028-102)**
PEN*KEY Model 6210 User's Guide **(P/N 961-028-090)**

PEN*KEY® Computer Multidock Installation Guides

Installation guides for PEN*KEY computer multidocks describe how to install the multidocks, and connect them to power and the network. Specific guides are:

6100 Series Docks Installation Instructions **(P/N 962-020-003)**

RS-485 LAN with 4000 Series, 6200, and 6210 Computers Site Preparation Guide **(P/N 961-028-097)**

Programmer's Reference Guides

PEN*KEY programmer guides contain information about power management, system and device support, and system messages for PEN*KEY computers. Specific programmer guides are:

PEN*KEY Model 6100 Computer Programmer's Reference Guide **(P/N 977-054-001)**

PEN*KEY Model 6200/6300 Computer Programmer's Reference Guide **(P/N 977-054-003)**

System Management Publication

Norand Open Wireless LAN with HP OpenView for Windows User's Guide **(P/N 961-051-009)**

This guide describes how to install and use the OpenView for Windows network management platform by Hewlett-Packard (HP).

Customer Support

The goal of Intermec Technologies Corporation is 100 percent customer satisfaction. If you would like more information about the 6710 Mobile Bridge or another system component, contact us through the Customer Response Center.

In the United States, call:	800-221-9236 *or* 319-369-3533
In Canada, call:	800-633-6149

Section 2

Features and Functional Overview

This section describes how the 6710 Mobile Bridge operates on the RS-485–Ethernet LAN. This section also describes hardware components.

Description

The 6710 Mobile Bridge (Figure 2-1) provides communications between a host on an Ethernet LAN and PEN*KEY® 6100, 6110, or 6210 Computers on an RS-485 LAN.

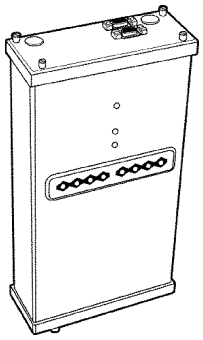


Figure 2-1
6710 Mobile Bridge

The 6710 Mobile Bridge functions as a 3-port translating bridge. Functionality within the device can be partitioned into two major functional blocks: *bridging functionality* and *management functionality*. Bridging functions pertain to the forwarding of data through the 6710 Mobile Bridge. Management functionality involves configuration, software upgrade, and network management.

Figure 2-2 is a simplified diagram showing the bridging and management functions within the 6710 Mobile Bridge.

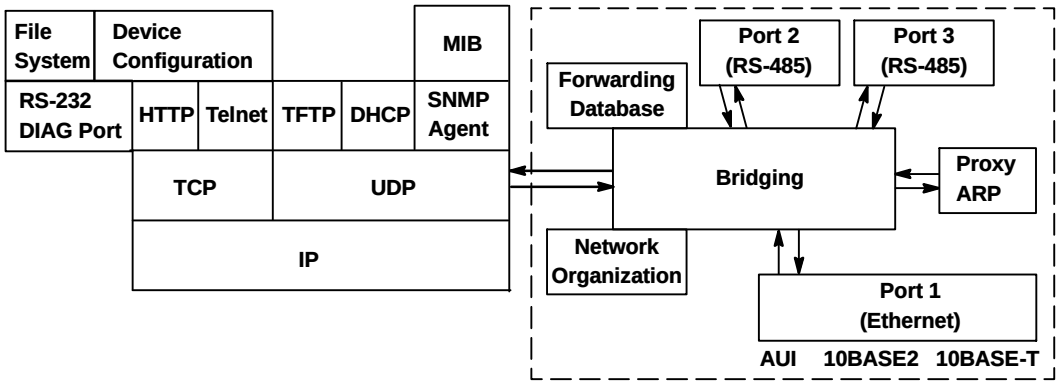


Figure 2-2
Bridging and Management Functions

Bridging Functionality

General Concepts

Bridges are common components in wired LANs. Bridges are devices that join two or more LAN segments. This provides the appearance of a single LAN segment to the protocols and applications that operate within the LAN.

Bridges operate at the Media Access Control (MAC) sublayer of the Data Link Layer (Layer 2) of the International Organization for Standardization (ISO) protocol model. Operating at the MAC layer allows bridges to operate transparently to commonly used network protocols such as TCP/IP, Novell SPX/IPX, NetBEUI, and DECnet.

In wired LANs, bridges do the following:

- Segment traffic for better efficiency and performance.
- Extend the reach of LANs when cable length or node limits have been reached.
- *Translate* between different LAN types.

A LAN environment normally consists of a collection of nodes or stations, each identified by a unique 48-bit physical address (also called an IEEE address or MAC address). Data is sent on the LAN as frames or packets that contain the *source address* of the station sending the frame, and the *destination address* of the recipient station.

A bridge has at least two *ports*, each connected to a different LAN segment. Bridges *learn* which source addresses are generating traffic on each of their ports. If the bridge receives a frame with a destination address corresponding to a source address it has seen on another port, it *forwards* the frame to the port.

If it receives a frame where the source and destination addresses are on the same port, it ignores *(drops)* the frame, since the destination node receives the original transmission. Generally, if a bridge receives a frame for an unknown destination address on any one port, it *floods* the frame on all other ports.

Bridging Layer

The 6710 Mobile Bridge functions as a bridge with up to three ports:

- An Ethernet port
- One or two RS-485 ports

The 6710 Mobile Bridge is a *translating bridge* because it forwards frames between Ethernet and RS-485 wired media that have unique physical and MAC protocol implementations. The 6710 Mobile Bridge implements the basic learning and forwarding functions of a simple wired LAN bridge.

Significant functions supported at the bridging layer include network organization and programmable flooding levels.

Network Organization

6710 Mobile Bridges automatically configure into a self-organized network using a spanning tree topology. They automatically reconfigure the network to maintain reliable operation as devices are added or removed, or in the event of some types of wired LAN failure. The spanning tree provides efficient, loop-free forwarding of frames through the network.

The spanning tree is initiated by the *super root*, a 6710 Mobile Bridge that coordinates the network and distributes common system parameters to other 6710 Mobile Bridges and RS-485 stations. The super root is elected from a group of 6710 Mobile Bridges designated at the time of installation. The election process also occurs in the event of a super root failure, preventing a single point of failure.

Forwarding

The 6710 Mobile Bridge maintains a forwarding database of all physical station addresses known to it, and the correct port for each address. This database makes efficient forwarding decisions in the bridging software. The database is updated through monitoring addresses on each port.

Flooding Configurations

Standard LAN bridges flood frames on all ports when the destination address is unknown. Additionally, many network protocols use *multicast* addressing for connection and status communications. A multicast frame is a special type of frame destined for more than one physical address. Standard bridges always flood multicast frames.

All RS-485 media supported in the 6710 Mobile Bridge operate at slower media speeds than Ethernet. Indiscriminate flooding from a busy Ethernet backbone to an RS-485 medium can consume a substantial portion of the available bandwidth. This reduces system performance even though flooded frames are frequently not intended for stations on a given RS-485 segment.

To allow performance tuning, the 6710 Mobile Bridge provides separate flooding control options for both unicast (single physical address) and multicast frames.

Certain ODI and NDIS drivers supplied with INTERMEC® RS-485 PEN*KEY computers use the OWL_ATTACH protocol to provide reliable attach mechanisms. These mechanisms guarantee that the RS-485 stations are always in the 6710 Mobile Bridge's forwarding database. Unicast flooding is never required for these stations.

Proxy ARP Server

The Proxy ARP Server is an advanced flooding control capability for RS-485 stations using IP. An ARP (Address Resolution Protocol) is a type of multicast message used to determine the physical (MAC) address of a station using a specific IP address. When Proxy ARP is enabled, the IP addresses of RS-485 stations using IP are included in the forwarding database. If the destination IP address matches an entry in the forwarding database, the ARP is sent to the physical unicast address matching that IP address.

To allow customization of this capability to optimize performance, the server operates in one of these modes:

- No flooding
- Delayed flooding
- Normal flooding

Proxy ARP Server is discussed in more detail in Section 4, "Configuration."

Bridge Ports

The 6710 Mobile Bridge has a physical Ethernet port and two RS-485 PC card slots.

Ethernet Port

The Ethernet port can be configured to support 10BASE-T twisted pair, 10BASE2 thinnet, or an AUI connection. The AUI connection can support 10BASE5 thicknet or 10BASEF fiber optic connections with the appropriate media adapters.

The physical connections are on the bottom panel of the 6710 Mobile Bridge. The desired Ethernet medium is selectable through the device configuration menus. More information about connecting the 6710 Mobile Bridge to Ethernet media is in Section 3, "Installation." Instructions for setting the medium through the configuration menus are in Section 4, "Configuration."

Ethernet Port Filters

The Ethernet port can be configured to support a variety of custom input filters. 6710 Mobile Bridges are commonly installed on LANs to carry traffic between Ethernet and RS-485 devices. Setting filters prevents unnecessary traffic from the Ethernet LAN from being forwarded onto the RS-485 LAN. This is important because common RS-485 technologies operate at data rates below Ethernet speeds.

Filters are set to *pass* frames likely used by RS-485 stations, and *drop* frames of a type which are known to not be used by stations requiring RS-485 connectivity. Filtering occurs in the Ethernet driver software that controls low level operation of the Ethernet ports, minimizing involvement of other functions when unnecessary frames are received. The default 6710 Mobile Bridge configuration sets no filters. Filter setup is discussed in more detail in Section 4, "Configuration."

Filtering vs. Flooding

Filtering and flooding control (described on page 2-5) are complimentary but have different functions. Filters allow frames to be eliminated based upon content of the frame, usually the network protocol header fields within the frame. For example, filters can be set to eliminate some or all IP traffic or Novell IPX traffic.

Filtering occurs regardless of whether the destination address is in the forwarding database. Using filters can improve the performance of the 6710 Mobile Bridge and prevent undesired frames from being forwarded to RS-485 stations connected to it.

Flooding decisions are made after frames have been received on a port and filtered. Flooding settings determine how the 6710 Mobile Bridge forwards frames to destination addresses not in the forwarding database.

RS-485 Ports

The two PC card slots are intended for RS-485 network interface cards (NICs) and are designated as NIC1 and NIC2. Internally, they are configured as Port 3 and Port 2, respectively. Configuration of individual RS-485 ports is discussed in Section 4, “Configuration.”

Configuration and Management

Configuration

The 6710 Mobile Bridge can be configured through a local RS-232 connection, or remotely through a TCP/IP connection. The 6710 Mobile Bridge includes a command monitor and menu driven configuration with online help.

The command monitor and file system configuration are contained in permanent read-only memory (ROM) within the 6710 Mobile Bridge, and can be accessed through the RS-232 diagnostics port even if software is not loaded in the device.

Most 6710 Mobile Bridge functionality is provided by the software stored within the file system. Configuration parameters are stored in nonvolatile EEPROM memory, and are maintained in the event of power loss.

Diagnostics and Configuration Port

An RS-232 configuration port is provided for direct access to the 6710 Mobile Bridge's command monitor and configuration menus. Access through the diagnostics port is password-protected for security.

The port uses a standard PC AT style cable, and operates at speeds up to 57.6 Kbps. Configuration using this port is described in Section 4, "Configuration."

Remote Access

Remote access is available over TCP/IP connections using Telnet or Hypertext Transfer Protocol (HTTP) for configuration management, and Simple Network Management Protocol (SNMP) for network management.

TCP/IP

The 6710 Mobile Bridge supports remote access through a Request for Comments (RFC) compliant TCP/IP stack. Before initial usage, the stack must be configured with an IP address and an optional default router through the RS-232 diagnostics port. Alternatively, the 6710 Mobile Bridge may be configured with a Dynamic Host Configuration Protocol (DHCP) server name. The 6710 Mobile Bridge then obtains its IP address, default router, and subnet mask from a DHCP server.

DHCP Client

The 6710 Mobile Bridge contains a DHCP client, allowing it to receive an IP address over the network. The DHCP client supports temporary and permanent leases. It also accepts permanent leases from a Bootstrap Protocol (Bootp) server. For further detail on DHCP operation, see Section 4, "Configuration."

Telnet

Telnet may be used to access the 6710 Mobile Bridge's configuration menus. The command interface is identical to the command interface through the diagnostics port. More information about access through Telnet is in Section 4.

HTTP

The 6710 Mobile Bridge supports configuration using HTTP from a workstation equipped with a Web browser. Microsoft Internet Explorer or Netscape Navigator is recommended. More information about access through a Web browser is in Section 4.

Electronic Software Distribution

The 6710 Mobile Bridge supports electronic software distribution, which allows software upgrades after installation. The 6710 Mobile Bridge provides a dual bank file system with one active bank and one inactive bank. It operates from the active bank, allowing software upgrades to be stored in the inactive bank. This enables upgrades to be loaded while the 6710 Mobile Bridge is operating.

The upgraded software can be used immediately after downloading by swapping the active and inactive banks and rebooting. The 6710 Mobile Bridge can also be programmed to load the new software at a later time, such as after all 6710 Mobile Bridges have been upgraded or during a time of little system activity.

TFTP Client and Server

Software downloads are accomplished using the Trivial File Transfer Protocol (TFTP), another member of the IP suite. Each 6710 Mobile Bridge contains a TFTP client and server.

The TFTP client allows the 6710 Mobile Bridge to obtain software updates from a TFTP server. The server can be a 6710 Mobile Bridge configured with the TFTP server enabled, or another network workstation with TFTP server capability.

Scripting

The 6710 Mobile Bridge supports a scripting capability that automates most of the software download process. The script process can be configured in the 6710 Mobile Bridge through Telnet or SNMP. The script file is loaded into the 6710 Mobile Bridge through TFTP, similar to the software.

Network Management

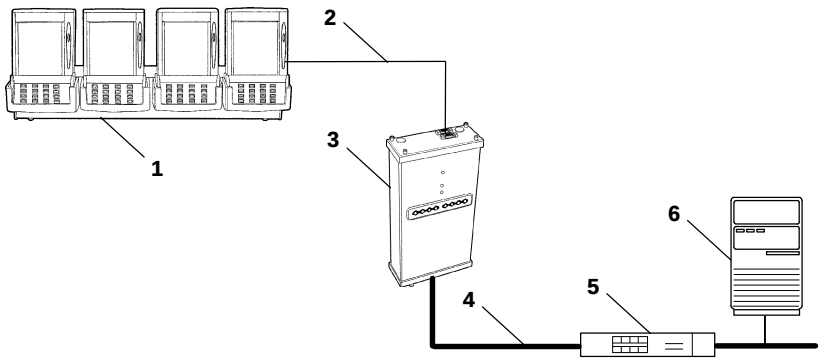
The 6710 Mobile Bridge is instrumented for network management, with variables defined in the Management Information Base (MIB). The MIB is SNMP V1 compliant.

Management information can be accessed through the SNMP agent. The MIB may be ordered separately and compiled for any SNMP network management platform.

For more information about network management, see Appendix C, "MIB." Also consult the *Norand Open Wireless LAN with HP OpenView for Windows User's Guide* (P/N 961-051-009).

Sample Network Configuration

Figure 2-3 shows a sample network configuration where the components of a bridged system provide communication between a 10BASE-T–based host and PEN*KEY 6100 Computers on an RS-485 LAN.



1. **PEN*KEY 6100 Computers (with TCP/IP stack and INTERMEC NDIS or ODI driver) in multidock**
2. **RS-485 LAN cable**
3. **6710 Mobile Bridge**
4. **TCP/IP over 10BASE-T Ethernet**
5. **10BASE-T Ethernet hub**
6. **Host**

Figure 2-3
Sample Network Configuration

PEN*KEY 6110 Computers and PEN*KEY 6210 Computers are also supported. Note that PEN*KEY computer multidocks can be daisy-chained.

Hardware Components

Figure 2-4 shows 6710 Mobile Bridge components, described on the following pages. Not shown is the mounting bracket, which attaches the 6710 Mobile Bridge to a wall.

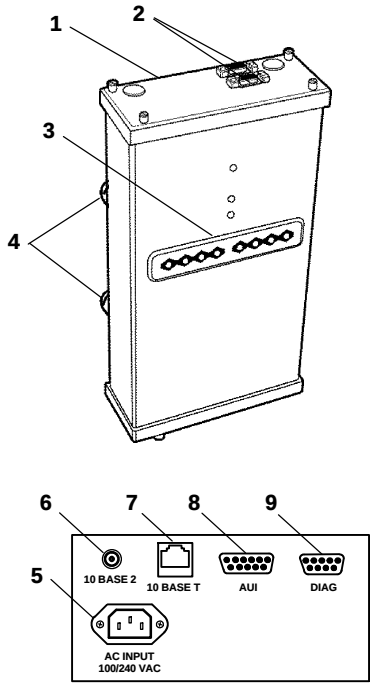


Figure 2-4
Hardware Components

1. **Protective cover.** The cover protects two RS-485 PC card slots.
2. **Connectors** (9-pin, D-sub female) for RS-485 PC cards.

3. **Indicator lights.** Four pairs of indicator lights (LEDs) on the front panel show the 6710 Mobile Bridge's status. During the power-up sequence, the lights show the results of the power-up self diagnostics and provide information about the operating status.

   After the power-up sequence, the lights show the current operating status and indicate if a problem exists. The lights are described in detail in Section 6, "Indicator Lights."

4. **Rubber feet.** Four nonskid rubber feet provide a stable base for the 6710 Mobile Bridge when you place it on a desktop or other horizontal surface. When the mounting bracket is installed for a 6710 Mobile Bridge mounted vertically, the rubber feet provide a small amount of tension to the bracket to help hold it in place.
5. **AC INPUT.** The AC INPUT connector is a standard IEC type, three-prong AC input connector. The power cord attaches to this connector. The internal power supply is an autosensing international power supply. It accepts a source voltage between 85 and 264 V ac, with a frequency between 47 and 63 Hz.
6. **10 BASE 2.** The 10 BASE 2 port is a standard BNC port through which the 6710 Mobile Bridge connects to 10BASE2 Ethernet (thinnet).
7. **10 BASE T.** The 10 BASE T port is a standard RJ45 port through which the 6710 Mobile Bridge connects to 10BASE-T (UTP) Ethernet.
8. **AUI.** The AUI port is a 15-pin, D-subminiature (D-sub) port. The 6710 Mobile Bridge connects to an AUI network adaptor through this port, for connection to 10BASE5 Ethernet (thicknet). Pin definitions are in Appendix B, "Port and Cable Pin-Outs."

" **NOTE:** *Instructions for connecting the 6710 Mobile Bridge to 10BASE2, 10BASE5, and 10BASE-T are in Section 3, "Installation."*

9. **DIAG.** The DIAG port is a 9-pin D-sub communication port that communicates at RS-232 levels. Use this port to configure the 6710 Mobile Bridge, download new software, and retrieve statistics. Pin definitions are in Appendix B.

Accessories

Power Cord

The power cord connects the 6710 Mobile Bridge to the wall outlet. The following chart lists power cord part numbers.

Country	Part Number
Australia	321-472-001
Denmark	321-501-001
Europe	321-473-001
Italy	321-471-001
Germany	321-515-001
United Kingdom	321-474-001
United States	321-054-001

Industrial Locking Mounting Bracket

The Industrial Locking Mounting Bracket (P/N 203-386-001) "locks" the 6710 Mobile Bridge into the bracket. This bracket is recommended for installations where vibration, shaking, or other movement can dislodge the 6710 Mobile Bridge from its mount.

Section 3

Installation

This section describes how to do the following:

- Check the 6710 Mobile Bridge's default configuration.
- Prepare for the installation.
- Collect the networking equipment you need.
- Select the best location for the 6710 Mobile Bridge.
- Connect the 6710 Mobile Bridge to PEN*KEY® multidocks.

Checking the Default Configuration

The 6710 Mobile Bridge is shipped with default settings for system software parameters, which are listed in Section 4, "Configuration." You may need to change some default settings to achieve a more efficient configuration for your site.

For information about reconfiguring the 6710 Mobile Bridge, see Section 4. The device should be properly configured before it is connected to the RS-485–Ethernet LAN.

Preparing for the Installation

Unpack the 6710 Mobile Bridge and inspect it for damage or missing parts. Save all the paperwork you received. If the 6710 Mobile Bridge appears to be damaged, contact the Customer Response Center for instructions on returning the device for replacement.

The shipment contains the 6710 Mobile Bridge with FLASH and the following items:

- Mounting bracket
- AC power cord
- Warranty card

Collecting the Equipment

Before you install the 6710 Mobile Bridge onto the network, collect the equipment you will need.

RS-485 LAN Components

The 6710 Mobile Bridge connects to the following PEN*KEY computer multidocks through a nonpowered RS-485 LAN cable:

- PEN*KEY 6100 Series Multidock (for PEN*KEY 6100 and 6110 Computers).
- 4960 Multidock (for PEN*KEY 6210 Computers). The multidock requires an external power supply and Y-cable P/N 226-151-001.

Table 3-1 lists recommended lengths for the nonpowered RS-485 LAN cable (P/N 226-103-0XX, where “0XX” is the cable length).

Table 3-1
Nonpowered RS-485 LAN Cable Lengths

Cable Length	P/N
1.5 feet	226-103-002
6 feet	226-103-004
10 feet	226-103-005
20 feet	226-103-006
50 feet	226-103-009
100 feet	226-103-014
200 feet	226-103-019
500 feet	226-103-028
1000 feet	226-103-038

Ethernet LAN Components

The 6710 Mobile Bridge directly connects to 10BASE2, 10BASE-T, or 10BASE5 Ethernet medium. Consult a cabling reference for maximum run lengths and node limits for Ethernet wiring.

10BASE2 Components

10BASE2 components include a T-connector, a cable terminator, and the proper lengths of 10BASE2 coax cable. The **10BASE2 T-connector** (Figure 3-1) attaches to the 6710 Mobile Bridge's 10BASE2 port, and connects the device to the middle or end of 10BASE2 cable.

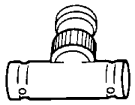


Figure 3-1
T-Connector

A **cable terminator** (Figure 3-2) attaches to the T-connector. It is required for a device connected to the end of 10BASE2 cable. The terminator properly terminates the network cable to maintain proper impedance. Proper termination is necessary for reliable Ethernet operation.

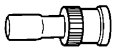


Figure 3-2
Cable Terminator

10BASE-T Component

10BASE-T **coax cable** normally connects the 6710 Mobile Bridge to an Ethernet hub. The cable has an RJ45 plug on each end (Figure 3-3).

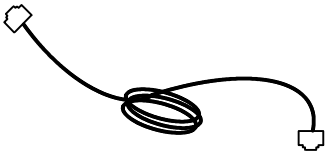


Figure 3-3
Cable With RJ45 Plugs

10BASE5 Components

10BASE2 components include the proper lengths of 10BASE5 coax cable, an AUI drop cable (less than or equal to 50 feet/15 meters long), and a transceiver. Two types of transceivers are the intrusive N-Series transceiver and the nonintrusive vampire tap.

The N-Series transceiver (Figure 3-4) is a T-shaped connector with a 15-pin AUI port and two type N connectors. This transceiver is intrusive because network service is disrupted while the coaxial cable is cut and a threaded N-series connector placed on each end of the cable.

A 10BASEF (fiber optic) adapter may be attached directly to the AUI connector.

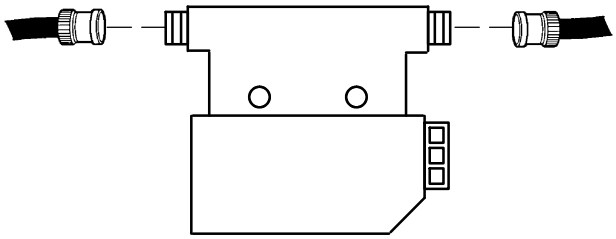


Figure 3-4
N-Series Transceiver

The vampire tap is an insulation-piercing clamp device that clamps onto the coaxial cable (Figure 3-5). The vampire tap pierces the coaxial cable's insulation and makes contact with the shield and inner conductor without cutting the cable.

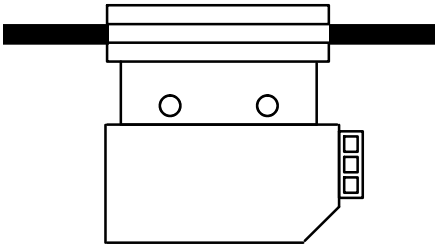


Figure 3-5
Vampire Tap

Communication Equipment

You can access the 6710 Mobile Bridge's system software configuration menus locally through the DIAG port, or remotely through a Telnet session or Web browser.

Local DIAG Port Access

For local access, you need the following:

- Third-party communications software terminal emulation package with Y-modem capability (such as PROCOMM PLUS by DataStorm Technologies, Inc.). Install the program according to its user guide.
- PC (personal computer) station, which should meet the requirements outlined in the user guide for the terminal emulation program.
- Cable to connect the PC to the 6710 Mobile Bridge's DIAG port. The following chart lists cables.

For this PC Port	Use Cable Part Number
9-pin	226-106-001 (null modem cable)
25-pin	321-355-001

Telnet

You need the following to access the configuration menus through a Telnet session:

- PC or workstation with an installed and configured network interface card and a Telnet application. You can also use a host capable of acting as a Telnet client.
- Telnet VT emulator (TNVT) installed on the PC.
- IP address for the 6710 Mobile Bridge. For more information about IP addresses, see Section 4.

Web Browser

The 6710 Mobile Bridge's configuration menus are designed for HTML Level 2.0 or higher. You need the following to access the configuration menus through a Web browser:

- Graphical browser application
- Internet or local network connection
- IP address for the 6710 Mobile Bridge

Network Management Platform

To manage the system through a network management platform, you need the platform (such as OpenView for Windows by Hewlett-Packard) installed on a network management station using SNMP. The station must meet the requirements outlined in the platform's user guide.

Selecting the Best Location

Factors related to the intended use of the system dictate installation practices. The following general practices should be followed in any installation:

- Locate the 6710 Mobile Bridge in the same area as other networking equipment, such as routers and hubs.
- Position the 6710 Mobile Bridge so its indicator lights are visible. The lights are useful for troubleshooting the installation.
- Install Ethernet LAN cabling within node limit and cable length limitations. Consult a cabling reference for maximum run lengths and node limits for Ethernet wiring.

- Ensure a power outlet is within 6 feet of the 6710 Mobile Bridge. An uninterruptable power supply is recommended when the ac power system is not reliable.
- Ensure that the RS-485 LAN cabling, Ethernet LAN cabling, and ac power cord can reach the 6710 Mobile Bridge after you install it. Leave sufficient room around the 6710 Mobile Bridge so you can easily attach and remove cables.

Mounting the 6710 Mobile Bridge

You can mount the 6710 Mobile Bridge horizontally on a tabletop, or vertically on a wall or post.

Horizontal (Tabletop) Mount

1. Remove the mounting bracket from the bottom of the 6710 Mobile Bridge. The bracket is not needed for a tabletop installation.
2. Set the 6710 Mobile Bridge in position. The device rests securely on four rubber feet that keep it from slipping out of place.
3. Make all RS-485 connections. See “Connecting to a Single Multidock” on page 3-11, or “Connecting to Two Multidocks” on page 3-12.
4. Make all Ethernet connections. See “Connecting to Ethernet” on page 3-13.
5. Make all power connections. See “Applying Power” on page 3-19.
6. Watch the indicator lights to verify that the 6710 Mobile Bridge is working properly. For help, see Section 6, “Indicator Lights.”

Vertical Mount

See Figure 3-6 and the procedure following it.

" **NOTE:** *If mounting the 6710 Mobile Bridge on a hollow wall, secure the mounting plate to a 3/4" (thick) plywood base by four 1" x 1/4" nuts, bolts, and washers. Anchor the plywood base to two separate wall studs by four 2" x 1/4" diameter lag screws (two lag screws in each stud).*

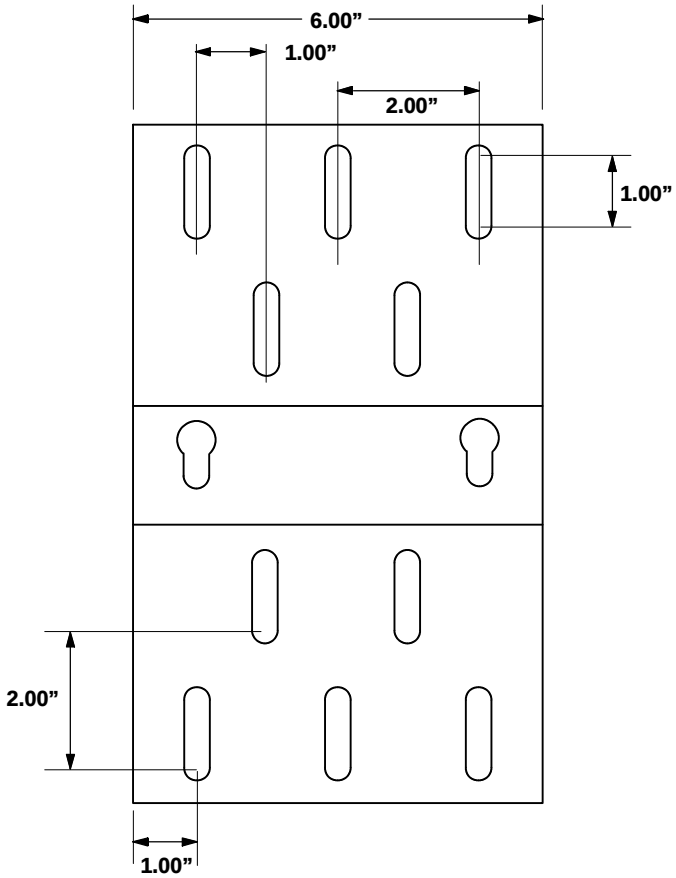


Figure 3-6
Mounting Bracket

1. Inspect where the 6710 Mobile Bridge will be mounted and determine what hardware is needed. Different surfaces such as drywall, wood, and concrete block require different mounting hardware. For this reason, a universal mounting bracket is included with the 6710 Mobile Bridge.
2. Remove the mounting plate from the bottom of the 6710 Mobile Bridge.
3. Using the mounting plate as a template, mark where the anchors that secure the mounting plate to the surface should be located.
4. Attach the 6710 Mobile Bridge mounting plate to the wall with 2" x 1/4" diameter lag screws or bolts, depending upon the surface. The mounting plate must be secured to the surface by at least four anchors, one on each corner.
5. Reattach the 6710 Mobile Bridge to the mounting plate.
6. Connect the 6710 Mobile Bridge to the PEN*KEY multidock. To connect to a single multidock, see page 3-11. To connect to two multidocks, see page 3-12.
7. Make all Ethernet connections. See "Connecting to Ethernet" on page 3-13.
8. Make all power connections. See "Applying Power" on page 3-19.
9. Watch the indicator lights to verify that the 6710 Mobile Bridge is working properly. For help, see Section 6, "Indicator Lights."

" **NOTE:** *An optional locking kit is available. For more information about the kit, see Section 2, "Features and Functional Overview."*

Connecting to a Single Multidock

To connect the 6710 Mobile Bridge to a PEN*KEY computer multidock, see Figure 3-7.

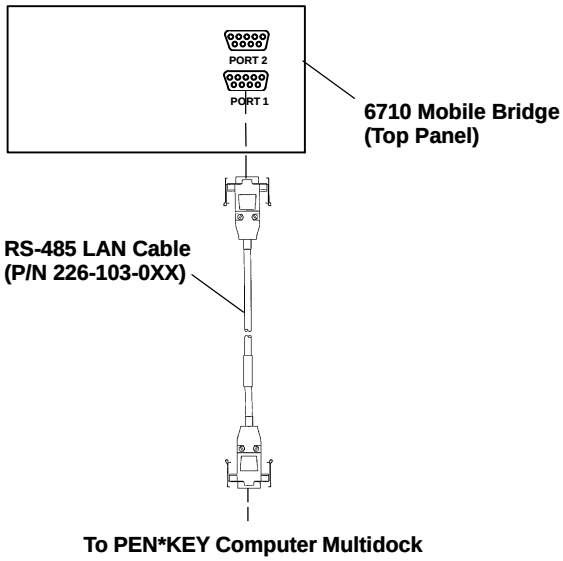


Figure 3-7
Connecting to a PEN*KEY Computer Multidock

For PEN*KEY multidock installation instructions, refer to the following publications:

- To connect to the PEN*KEY 6100 Series Multidock (for PEN*KEY 6100 and 6110 Computers), refer to the *6100 Series Dock Installation Instructions* (P/N 962-020-003).
- To connect to the 4960 Multidock (for PEN*KEY 6210 Computers), refer to the *RS-485 LAN with 4000 Series, 6200, and 6210 Computers Site Preparation Guide* (P/N 961-028-097).

Connecting to Two Multidocks

To connect the 6710 Mobile Bridge to two PEN*KEY computer multidocks, see Figure 3-8.

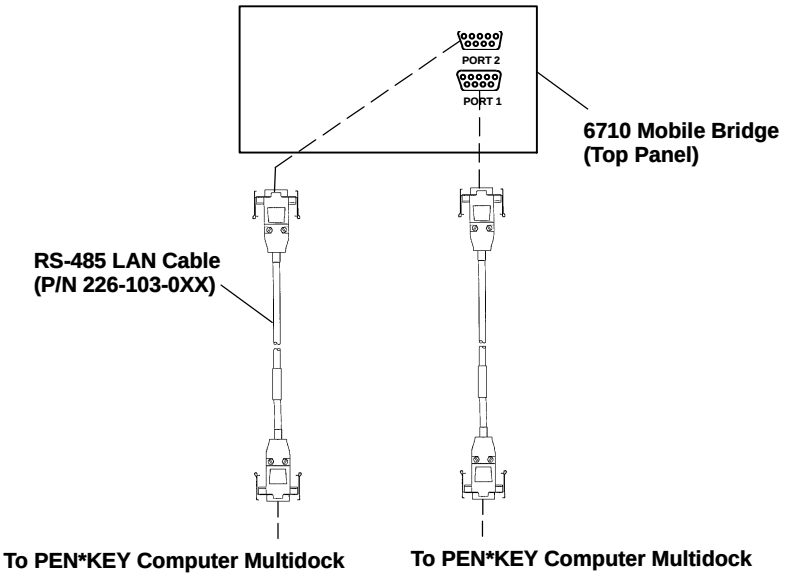


Figure 3-8
Connecting to Two PEN*KEY Computer Multidocks

For PEN*KEY multidock installation instructions, refer to the following publications:

- To connect to the PEN*KEY 6100 Series Multidock (for PEN*KEY 6100 and 6110 Computers), refer to the *6100 Series Dock Installation Instructions* (P/N 962-020-003).
- To connect to the 4960 Multidock (for PEN*KEY 6210 Computers), refer to the *RS-485 LAN with 4000 Series, 6200, and 6210 Computers Site Preparation Guide* (P/N 961-028-097).

Connecting to Ethernet

The following pages show how to connect the 6710 Mobile Bridge to 10BASE2, 10BASE5, and 10BASE-T Ethernet.

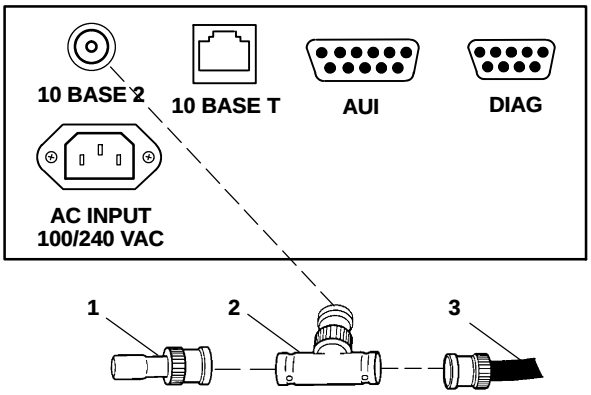
10BASE2 Ethernet

The 6710 Mobile Bridge connects to the end or middle of the 10BASE2 cable segment.

" **NOTE:** *Cable lengths between network devices on the 10BASE2 Ethernet LAN must meet ANSI/IEEE standards.*

End of Segment

See Figure 3-7 and the procedure following it.



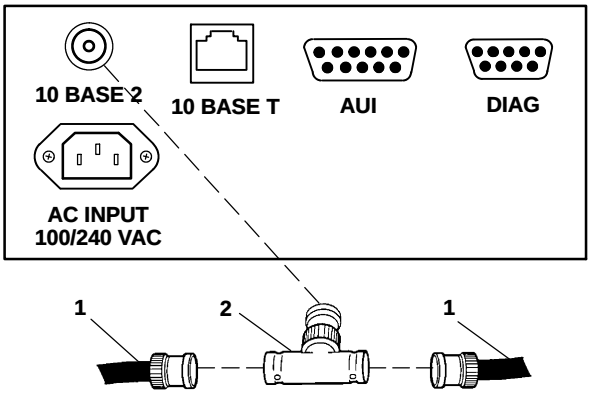
1. **Cable terminator (50 ohm)**
2. **T-connector**
3. **10BASE2 cable**

Figure 3-9
End of 10BASE2 Segment

1. Plug the T-connector (2) into the 10 BASE 2 port.
2. Plug one end of the Ethernet cable (3) into an open end of the T-connector. Align the notches in the cable end with the posts on the T-connector, push the cable in, and twist one-quarter turn.
3. Plug the cable terminator (1) into the other end of the T-connector.

Middle of Segment

See Figure 3-10 and the procedure following it.



1. 10BASE2 cable
2. T-connector

Figure 3-10
Middle of 10BASE2 Segment

1. Plug the T-connector (2) into the 10 BASE 2 port.
2. Plug one end of the Ethernet coaxial cable (1) into an open end of the T-connector. Align the notches in the cable end with the posts on the T-connector, push the cable in, and twist about one-quarter turn.
3. Plug the end of another Ethernet coaxial cable segment into the other open end of the T-connector.

10BASE5 Ethernet

The 6710 Mobile Bridge connects to 10BASE5 through an N-Series transceiver or vampire tap.

" **NOTE:** *Cable lengths between network devices on the 10BASE5 Ethernet LAN must meet ANSI/IEEE standards.*

N-Series Transceiver

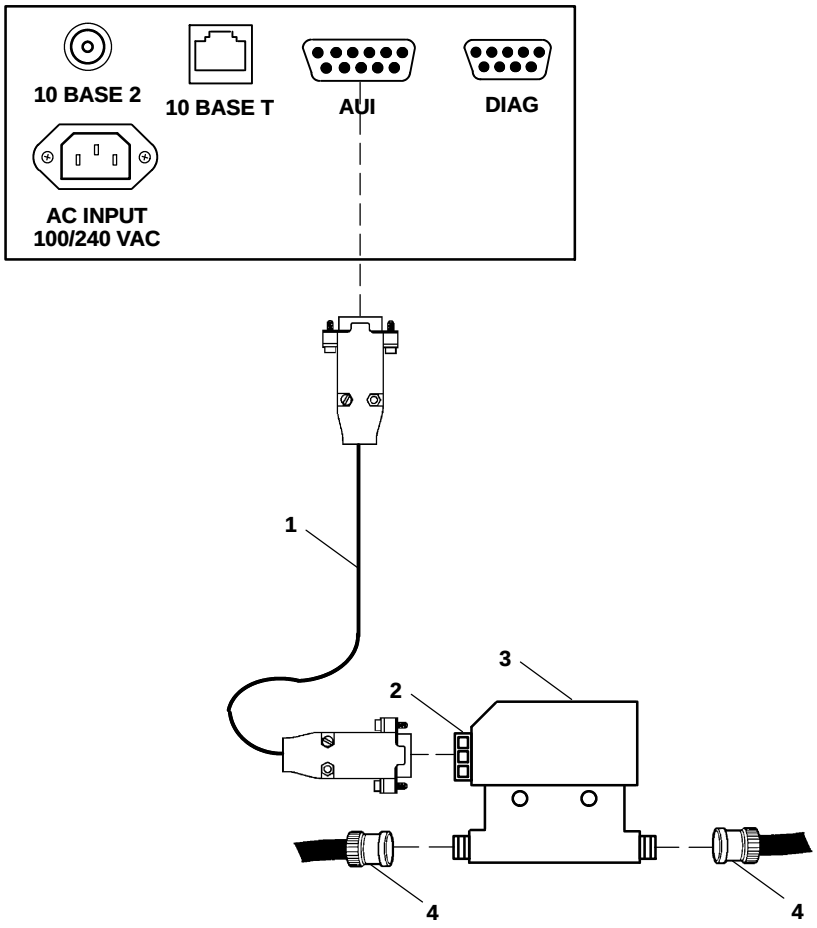
See Figure 3-11 and the following procedure.

1. Attach one end of the drop cable (1) to the AUI port.
2. Route the drop cable to the 10BASE5 cable (4) and determine a suitable spot to cut the cable and attach the transceiver (3).
3. Attach the transceiver to the 10BASE5 cable, then connect the other end of the drop cable to the AUI port (2) on the transceiver.

Vampire Tap

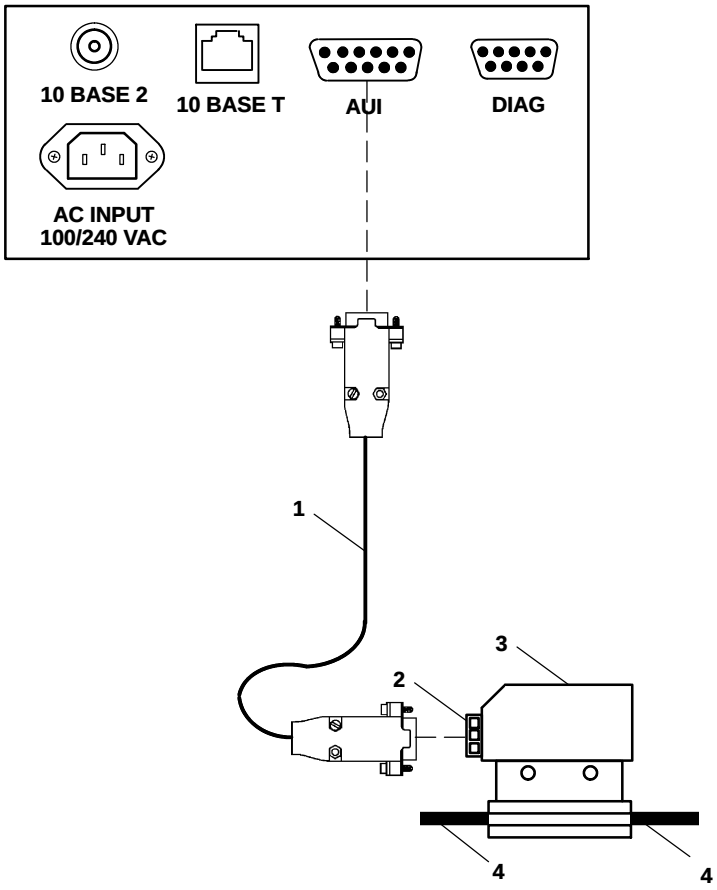
See Figure 3-12 and the following procedure.

1. Attach one end of the drop cable (1) to the AUI port.
2. Route the drop cable to the 10BASE5 cable and determine a suitable spot on the cable to attach the vampire tap (3).
3. Attach the vampire tap to the 10BASE5 cable, then connect the other end of the drop cable to the AUI port (2) on the tap.



1. **Drop cable**
2 **15-pin AUI port**
3. **N-Series transceiver**
4. **10BASE5 coax**

Figure 3-11
N-Series Transceiver



1. **Drop cable**
2. **15-pin AUI port**
3. **Vampire tap**
4. **10BASE5 coax**

Figure 3-12
Vampire Tap

10BASE-T Ethernet

See Figure 3-13 and the procedure following it.

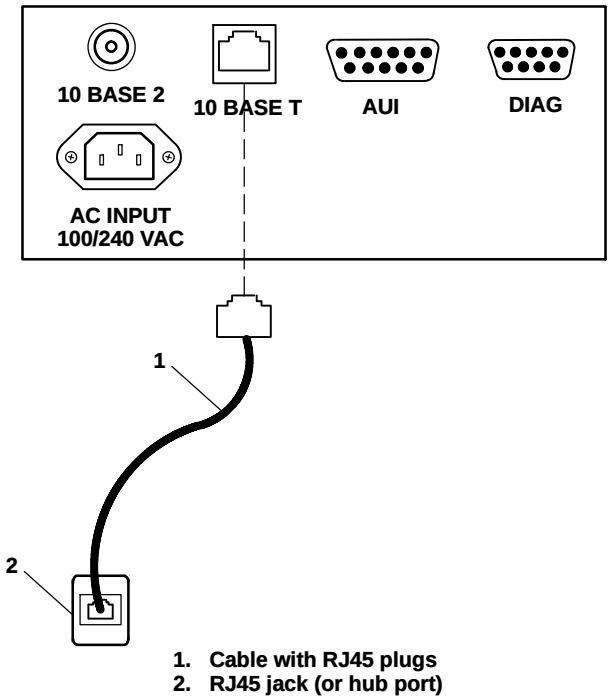


Figure 3-13
10BASE-T

1. Plug the cable with RJ45 jacks (1) into the 10 BASE T port.
2. Plug the other end of the cable into RJ45 jack or hub port (2).

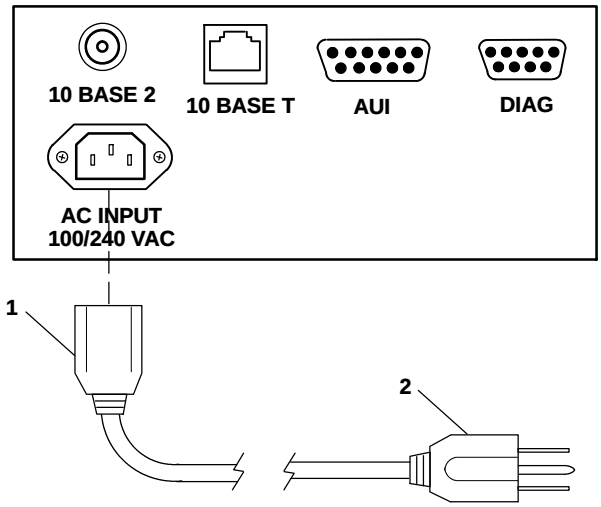
Applying Power

B CAUTION: **NEVER remove the cover of the 6710 Mobile Bridge with power applied. ALWAYS make the 6710 Mobile Bridge connection before making the connection at the source ("load to source"). Damage can occur with the cover removed.**

" NOTE: *Connect the 6710 Mobile Bridge to an uninterruptable power source — a power source that cannot be inadvertently turned off or otherwise disconnected.*

Power is applied to the 6710 Mobile Bridge through the grounded AC INPUT connector. To apply power, see Figure 3-14 and the following procedure.

1. Plug the receptacle end of the power cord (1) into the AC INPUT connector.
2. Insert the three-prong plug on the other end of the power cord (2) into a grounded power outlet.
3. For descriptions of the indicator lights, see Section 6, "Indicator Lights."



1. **Receptacle on power cord**
2. **Three-prong plug**

Figure 3-14
AC Power Input Connection

Section 4

Configuration

This section describes how to:

- Create a local DIAG port, Telnet, or Web browser session with the 6710 Mobile Bridge.
- Access the 6710 Mobile Bridge's FLASH and ROM.
- Set up the 6710 Mobile Bridge through its configuration menus.

You can configure the 6710 Mobile Bridge locally through its DIAG port, or remotely through Telnet or a Web browser. The following chart shows the sessions you can use to do other tasks.

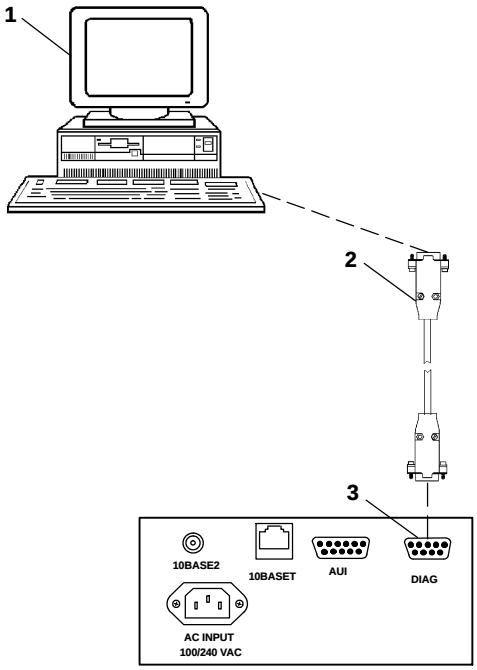
Task	DIAG Port	Telnet	Browser
Change configuration passwords	√	√	√
Modify the configuration	√	√	√
Upgrade FLASH	√	√	
Check the FLASH version	√	√	
Access ROM	√		
Check the ROM version	√		
Use online help	√	√	√

Only one type of session can be running at a time. For example, if someone starts a Telnet session while someone else is configuring the 6710 Mobile Bridge through its DIAG port, the configuration through the DIAG port will terminate.

Creating a Local DIAG Port Session

You establish a local DIAG port session with the 6710 Mobile Bridge through a VT100 terminal emulation program. Most general purpose communications software (such as PROCOMM PLUS) supports this emulation.

To create a session, see Figure 4-1 and the procedure following it. You should carefully review the procedure first to become familiar with the process.



1. **PC with terminal emulation program**
2. **Cable: P/N 321-355-001 for a 25-pin PC COM port**
   or
   Cable: P/N 226-106-001 for a 9-pin PC COM port (standard null modem cable)
3. **6710 Mobile Bridge DIAG port**

Figure 4-1
Local Session

1. Ensure the terminal emulation program is installed on the PC.
2. With both the PC and 6710 Mobile Bridge powered OFF, connect the communication cable to the appropriate PC COM port.

3. Connect the other end of the communication cable to the DIAG port on the 6710 Mobile Bridge. Turn the PC on.
4. After the PC boots, start the terminal emulation program.
5. Set the terminal emulation program's options according to what you want to do: Access the configuration menus, or access the ROM command monitor.

Accessing the Configuration Menus

1. Set the terminal emulation parameters in your communications software. If you are configuring this 6710 Mobile Bridge for the first time, set the parameters to the device's default settings:

   9600, **8N1, full duplex**

   If you have already changed the default settings, set the parameters to those you set in FLASH mode through the configuration menus.
2. Plug the 6710 Mobile Bridge into the outlet. These messages appear:

QXS6700K <version> <date>
<Press any key within 5 seconds to enter the ROM monitor>
Executing file MSD29.BIN from segment <segment number>
Quickly press a key to perform configuration before startup
Starting system

3. To access the configuration menus, wait until you see the message "Quickly press a key to perform configuration before startup." Press any key to access the configuration menus.

4. See "Configuring the 6710 Mobile Bridge" on page 4-10.

Accessing the ROM Command Monitor

1. Set the terminal emulation parameters in your communications software. If you are configuring this 6710 Mobile Bridge for the first time, set the parameters to the default settings for ROM mode:

   9600, 8N1, full duplex

   If you have already changed the default settings, set the parameters to those you set in ROM mode through the ROM command monitor.

2. Plug the 6710 Mobile Bridge into the outlet. These messages appear:

QXS6700K <version> <date>
<Press any key within 5 seconds to enter the ROM monitor>
Executing file MSD29.BIN from segment <segment number>
Quickly press a key to perform configuration before startup
Starting system

3. Press any key within 5 seconds of the first ROM message.

   Note that if the 6710 Mobile Bridge is in Power-Up Quiet mode (versus Power-Up Normal mode, the default setting), the ROM messages do not display. More information about Power-Up Quiet (PQ) mode and Power-Up Normal (PN) mode starts on page 5-27 in Section 5, "Software Download."

4. For information about the ROM command monitor, see page 5-22 in Section 5, "Software Download."

Creating a Telnet Session

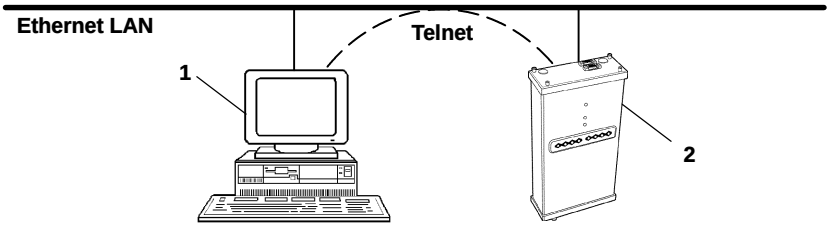
Before you can configure the 6710 Mobile Bridge through Telnet, you must connect it to the Ethernet cable. (See Section 3, "Installation," for help.) You must also perform initial configuration through the DIAG port to:

- Set an IP address or DHCP server name. You should also configure a subnet mask and IP router address.
- Set the Ethernet cable type.

" NOTE: *The 6710 Mobile Bridge includes an autodetect feature that senses the Ethernet medium if traffic is present. If no traffic is present on the cable, the system software defaults to 10BASE-T. For most installations, it is recommended that you explicitly set the Ethernet type.*

The 6710 Mobile Bridge must go through its boot sequence before you can create a Telnet session. If you reboot the device while in a session, the session terminates. You can create a new session after the device reboots.

To create a Telnet session, see Figure 4-2 and the procedure following it.



1. **PC or workstation with Telnet VT emulator (TNVT)**
2. **6710 Mobile Bridge**

Figure 4-2
Telnet Session

1. Ensure the 6710 Mobile Bridge is connected to the Ethernet cable, and has an assigned IP address and cable type.
2. Ensure the PC or workstation has an installed and configured Ethernet interface card.
3. Ensure the Telnet VT emulator is installed on the PC or workstation.
4. Open a new Telnet session on the PC or workstation.
5. Enter the 6710 Mobile Bridge's IP address in the host name or IP address field.
6. See "Configuring the 6710 Mobile Bridge" on page 4-10.

Default and Site Settings

The 6710 Mobile Bridge is factory configured with the default settings listed in the following charts. You may need to change the defaults to match the way your system is set up. You can record your site's settings in the table for reference.

TCP/IP Options

Option	Default	Site Setting
IP Address	0.0.0.0	
IP Subnet Mask	255.255.255.0	
IP Router	0.0.0.0	
IP Frame Type	DIX	
DHCP	Enabled, if IP Address is zero	
DHCP Server Name	Norand DHCP Server	
Auto ARP Minutes	5	

Security Options

Option	Default	Site Setting
Password	CR52401	
Service Password	Enabled	

Bridge Options

Option	Default	Comments
Serial Number	*(Read-only)*	
Lan ID	0	
[Root]		
Root Priority	1	
[Global Flooding]		
Inbound		
Multicast	Primary	
Unicast	Disabled	
Outbound to Secondaries		
Multicast	Disabled	
Unicast	Disabled	
[Ports]		
Ethernet port:		
Name	omde	
MAC Address	*(Unique number)*	
Status	Enabled	
Hello Period	2 seconds	
[Ethernet]		
OWL Frame Type	DIX	
Cable Type	Auto Detect	
[Static Addresses]	00:00:00:00:00:00	
[Normal RX Filter]		
[Frame Types]		
Action	Pass	
Scope	Unlisted	

Option	Default	Comments
[SubTypes 1]		
Action	Pass	
SubType	*(Various)*	
Scope	*(Various)*	
[SubTypes 2]		
Action	Pass	
SubType	DIX–IP–TCP–Port	
Scope	00 00	
[Advanced RX Filter]		
[Expressions]		
ExprSeq	0	
Offset	0	
Op	EQ	
Value Id	0	
Action	And	
[Values]		
Value	0	
RS-485 port:		
Name	omd485a *or* omd485b	
MAC Address	*(Ethernet MAC address)*	
Status	Enabled	
Hello Period	2 seconds	
[B_B485]		
Baud Rate	115200	
ARP Server Mode	No Flooding	

Notes:

Configuring the 6710 Mobile Bridge

When you create a local DIAG port or remote Telnet session with the 6710 Mobile Bridge, the configuration program's password screen appears:

```
Configuration of 6710 Mobile Bridge
Copyright (c) 1995-1998 Intermec Technologies Corporation. All
rights reserved.
Portions copyright Epilogue Technology Corporation 1988-1995.
All rights reserved

IP:       0.0.0.0
Serial:   (Unique 10-digit number.)

Password:
```

" **NOTE:** *A different screen appears when you create a session through a Web browser. See page 4-46 for information about Web browser sessions.*

The password screen shows the current settings for the IP address and serial number. It also shows the prompt for the top-level password. Enter the password (case insensitive) to display the Main Menu. The default password is CR52401.

Main Menu

After you enter the top-level password, the Main Menu appears:

```
Loading configuration from EEPROM

Command  Description
File     File system menu
View     View/modify the configuration
Clear    Set the configuration to default values
Read     Read the configuration from EEPROM
Write    Write the configuration to EEPROM
Reboot   Restart using last written configuration
Exit     Disconnect
?        Display this help

>
```

The menu lists the commands you can use to do various tasks, described on the following pages. The screen also displays the command prompt (>). At the prompt, type the name of the command you want to perform and press [Enter]. (Commands are case insensitive.) The Main Menu redisplays when you enter an invalid command.

The following chart describes how to use the commands.

Use	**To**
File	List file system commands and descriptions. Section 5, "Software Download," describes the commands and file system methodology.
View	View or modify configuration program settings. See "Using the View Command" on page 4-12.
Clear	Reset the 6710 Mobile Bridge's configuration to the factory-set default settings, which start on page 4-7.
Read	Load the most recent configuration from EEPROM. The configuration that was written to EEPROM **since the 6710 Mobile Bridge was last rebooted** becomes the new configuration.

Use	To
Read *(Continued)*	The 6710 Mobile Bridge's configuration is stored in EEPROM. You reprogram the EEPROM whenever you change the configuration, write (save) the new configuration to EEPROM, and reboot the 6710 Mobile Bridge.
Write	Write (save) a new configuration to EEPROM. This command overwrites the previous configuration. **You must write the new configuration to EEPROM and reboot the 6710 Mobile Bridge for any changes to take effect.**
Reboot	Reboot the 6710 Mobile Bridge. You must reboot it for any changes you made to the configuration to take effect.
Exit	Quit the configuration program. If you exit a new configuration without writing it to EEPROM, any changes you made are **not** saved.
?	Display online help for a command, option, or setting.

Using the View Command

To view or modify configuration program settings, type View at the command prompt. The Main Options Menu appears:

```
[Tcpip]
[Bridge]
[Security]
```

The following chart describes how to use the options.

Use	To
[Tcpip] *Page 4-14*	Set options necessary for communications with this 6710 Mobile Bridge. The options apply to all TCP/IP ports. Telnet, SNMP, and HTTP communications are supported.
[Bridge] *Page 4-21*	Control the bridging of messages among the RS-485 and Ethernet ports for this 6710 Mobile Bridge. Settings to control interaction with other 6710 Mobile Bridges are also under the [Bridge] option.
[Security] *Page 4-45*	Set the configuration program's top-level password and service password.

The screens in this section show the options' default settings. Some settings (such as the serial number) are unique to each 6710 Mobile Bridge. Other settings are automatically set and you cannot change them. This section identifies the settings you cannot change as "read-only."

The following chart shows how to navigate the View command's menus and edit data.

Press	To
[↑] or [–]	Scroll up through items in a list.
[↓], [+], [=], or [Tab]	Scroll down through items in a list.
[→], [Enter], or [Spacebar]	Display an option's settings or prompt after you highlight the option. Also use these keys to select the desired setting.
[←], [Esc], or [Backspace]	Exit a menu or prompt.
[Esc]	Cancel editing.
[Enter]	Complete editing.

TCP/IP Options

Use [Tcpip] to set options necessary for communications with this 6710 Mobile Bridge, such as IP addresses. Addresses are required for remote setup or SNMP network management. Options are:

```
IP Address          0.0.0.0
IP Subnet Mask      255.255.255.0
IP Router           0.0.0.0
IP Frame Type       <DIX>
DHCP                <Enabled, if IP address is zero>
DHCP Server Name    "Norand DHCP Server"
Auto ARP Minutes    5
```

IP Address

IP Address is the unique address locally assigned to this 6710 Mobile Bridge. The prompt is:

```
Range is:
4 nums 0..255
```

The default is 0.0.0.0, which disables the ability to use TCP/IP. Following are suggestions for setting the address:

- If you are installing this 6710 Mobile Bridge on an existing Ethernet segment, you should allocate the IP address from the same pool as the existing computers on the segment.
- If you are installing this 6710 Mobile Bridge on a new Ethernet segment that is not going to connect to the Internet, try using this Class B address:

  172.16.h.h

The host number is "h.h." This Class B network address is reserved by the numbering authority for a company's internal use. If the Class B address appears on the Internet, routers drop the data.

Note the following:

- If the IP address is 0.0.0.0 *and* DHCP is enabled, this IP address is obtained through DHCP.
- If the IP address is 0.0.0.0 *and* DHCP is disabled, TCP/IP access to this 6710 Mobile Bridge is disabled.

A discussion of DHCP starts on page 4-18.

IP Subnet Mask

IP subnets partition traffic and are connected by routers. The subnet mask indicates how many bits of the IP address represent a network number and how many indicate a host number. The prompt is:

```
Range is:
4 nums 0..255
```

The default is 255.255.255.0. Following are suggestions for setting the subnet mask:

- If you are installing this 6710 Mobile Bridge on an existing Ethernet segment, the subnet mask should match the other computers on the segment.
- If you are using the 172.16.h.h address suggested for IP Address, you may want to use a subnet mask of 255.255.248.0. This mask provides the network 172.16 with 30 subnets of 2046 computers each.

The IP address breakdown is:

- 16 bits of network address.
- 5 bits of subnet address. Do not use all 0's or all 1's.
- 11 bits of host address. Do not use all 0's or all 1's.

The following chart lists IP addresses when the submask is 255.255.248.0.

Subnet	First Address	Last Address
1	172.16.8.1	172.16.15.254
2	172.16.16.1	172.16.23.254
3	172.16.24.1	172.16.31.254
.		
.		
.		
30	172.16.240.1	172.16.247.254

If you are using DHCP to obtain an IP subnet mask for this 6710 Mobile Bridge, the subnet mask obtained from DHCP overrides the setting for the IP Subnet Mask option.

IP Router

" **NOTE:** *The IP address of the router is required only if this 6710 Mobile Bridge will communicate with devices on the other side of the router.*

IP Router identifies the default router used to forward data frames to addresses on another subnet. The prompt is:

```
Range is:
4 nums  0..255
```

The default is 0.0.0.0, which disables the ability to exchange TCP/IP traffic with another subnet or network.

A router that connects subnet 1 to subnet 2 might have the address 172.16.8.1 on subnet 1 and 172.16.16.1 on subnet 2. A host with IP address 172.16.16.5 would specify an IP router address of 172.16.16.1 to reach host 172.16.8.10.

IP routers are usually configured so a computer only needs to know one router's address. This is true even if several routers on the segment connect to several other segments.

If you are using DHCP to obtain an IP router address, and the DHCP server specifies a default IP router, the DHCP server specification overrides the setting for IP Router.

IP Frame Type

IP Frame Type sets the type of frame containing IP traffic:

DIX 802.3

Setting	Description
DIX *(default)*	Sets Ethernet type to DIX (Ethernet 2.0) for IP frames.
802.3	Sets Ethernet type to 802.3 with a SNAP header for IP frames. Select 802.3 if other network computers use SNAP encapsulation for IP frames.

DHCP

DHCP provides a way for this 6710 Mobile Bridge (the client) to obtain IP addresses from a DHCP server on the network. Settings are:

```
Enabled
Enabled, if IP address is zero
Disabled
```

Setting	Description
Enabled	Always uses DHCP to obtain the IP address, subnet mask, and (optional) default IP router when the 6710 Mobile Bridge reboots.
Enabled, if IP address is zero *(default)*	Retrieves IP addresses for the 6710 Mobile Bridge, subnet mask, and default router, and the lease expiration time, from the DHCP server. The 6710 Mobile Bridge ignores other configuration options. Note that if you enable DHCP, you must set IP Address to 0.0.0.0.
Disabled	Disables DHCP. You must manually set the IP addresses before the TCP/IP stack is enabled.

The 6710 Mobile Bridge responds only to address offers from DHCP or Bootp servers. In either case the server is specified in the DHCP server name field.

DHCP Server Name

The prompt for the DHCP server name is:

```
Range is:
31 chars
```

The 6710 Mobile Bridge responds only to the named server. The default server name is "Norand DHCP Server." This name prevents the 6710 Mobile Bridge from inadvertently obtaining an IP configuration from existing servers on the network.

If the DHCP server name is configured with a null string (""), the 6710 Mobile Bridge responds to offers from any server.

The class identifier string for the 6710 Mobile Bridge is "Norand Access Point." Servers use this string to identify the 6710 Mobile Bridge.

Bootp Operation

The 6710 Mobile Bridge can also accept addresses from a Bootp server identified in the DHCP server name field. An address offer from a Bootp server is treated as if it were an infinite lease from a DHCP server.

Networks With DHCP and Bootp Servers

If the DHCP server name is configured as "", the 6710 Mobile Bridge responds to either DHCP or Bootp servers. The 6710 Mobile Bridge gives preference to DHCP offers. If a Bootp reply arrives at the 6710 Mobile Bridge before any DHCP offers are received, it waits an additional 4 seconds for a DHCP offer before responding. If a DHCP offer is received within the 4-second period, the Bootp reply is ignored and the DHCP offer is accepted.

Handshaking

When the 6710 Mobile Bridge responds to a DHCP or Bootp server, it broadcasts a single ARP request to the address offered. If no ARP response is received within 3 seconds, the 6710 Mobile Bridge assumes the IP address is unique and completes the negotiation for that address. If an ARP reply is received before the timeout, the 6710 Mobile Bridge assumes the address is a duplicate and declines the offer.

Infinite Leases

A DHCP server may be configured to grant an infinite lease to the 6710 Mobile Bridge. A Bootp grant is always treated as an infinite lease. The 6710 Mobile Bridge stores the IP address, subnet mask, and default router in the EEPROM configuration register and disables DHCP. These settings are maintained if the 6710 Mobile Bridge is powered off or rebooted through the ROM command monitor. To restore DHCP client operation, reconfigure the IP address to 0.0.0.0.

" **NOTE:** *DHCP is documented in RFCs 1533, 1534, and 1541. Bootp is documented in RFC 951.*

Auto ARP Minutes

The 6710 Mobile Bridge periodically sends an unsolicited ARP response so routers can update their routing tables. The response enables a network management platform to learn about the 6710 Mobile Bridge on the network by querying routers.

Auto ARP Minutes is the number of minutes between periodic ARP requests. The prompt is:

```
Range is:
0..120
```

The default is 5 minutes. A setting of 0 disables Auto ARP Minutes.

If the default router's address is 0, the ARP request is sent to the IP address of this 6710 Mobile Bridge. Without the Auto ARP Minutes option, a 6710 Mobile Bridge might not use its IP address for extended periods of time and expire from the router's ARP table.

Auto ARP Minutes enhances the discovery of the network architecture by network management tools, such as OpenView by Hewlett-Packard. The network management tool queries IP router ARP tables to locate the active IP addresses for the subnet IP addresses for 6710 Mobile Bridges should not be allowed to expire. The network management program would then need to ping all potential addresses on a subnet to locate active IP addresses, or require the user to enter a list.

Bridge Options

Use [Bridge] to configure options that define the bridging topology of the RS-485–Ethernet LAN. Options are:

Serial Number	*"(Unique 10-digit number.)"*
Lan ID	0
[Root]	
[Ports]	
ARP Server Mode	<No Flooding>

Serial Number

Serial Number is a read-only setting that displays this 6710 Mobile Bridge's unique 10-digit serial number, which identifies the device on the network.

Lan ID

The LAN ID (also called *domain*) is a number that logically isolates adjacent but independent open wireless LANs.

The prompt is:

```
Range is:
0..254
```

You should never need to change the default of 0. All 6710 Mobile Bridges on the same Ethernet LAN segment should have the same LAN ID.

[Root]

[Root] options apply to 6710 Mobile Bridges configured to operate as the super root. They should be set to the same settings in all 6710 Mobile Bridges with a nonzero root priority configured. Options are:

```
Root Priority
[Global Flooding]
```

Root Priority

Root Priority determines which 6710 Mobile Bridges are candidates to become the super root node on the distribution LAN (also called *primary LAN*). The prompt is:

```
Range is:
0..7
```

The default root priority is 1.

Super Root Candidates

6710 Mobile Bridges assigned a root priority between 1 and 7 are candidates to become the super root. 6710 Mobile Bridges assigned a root priority of 0 are prohibited from becoming the super root.

Super Root Selection

The 6710 Mobile Bridge with the highest assigned root priority becomes the super root whenever it is powered on and active. If the current super root goes offline, the remaining candidates negotiate to determine which one becomes the new super root. This normally takes about 1 minute.

The super root is always the 6710 Mobile Bridge with the highest root priority (other than 0). If two or more 6710 Mobile Bridges have the same root priority, the one with the highest Ethernet address becomes the super root.

Super Root Redundancy

For *redundancy*, two or three 6710 Mobile Bridges should have a nonzero root priority. All other 6710 Mobile Bridges should have a root priority of 0. (Redundancy is the ability of another 6710 Mobile Bridge to take over if the super root goes offline.)

You should do the following:

- Configure one 6710 Mobile Bridge as a primary super root (with the highest root priority).
- Configure one or two 6710 Mobile Bridges as “fallback” super roots (with lower priority).
- Configure remaining 6710 Mobile Bridges with a root priority of 0.

[Global Flooding]

" **NOTE:** *Use the same [Global Flooding] settings in all super root candidates.*

Use [Global Flooding] to set system-wide flooding options. The settings are sent throughout the network when and if this 6710 Mobile Bridge becomes the super root. Options are:

	Multicast	Unicast
Inbound	<Primary>	<Disabled>
Outbound to Secondaries	<Disabled>	<Disabled>

A 6710 Mobile Bridge normally forwards frames only to destination addresses it has learned and stored in the forwarding database. Frames are forwarded only on the port that provides the shortest path to the destination address. The 6710 Mobile Bridge can be configured to flood frames on one or more ports when the destination address is unknown.

Global flooding options allow for different flooding configurations to optimize performance. Settings in the super root are distributed to all other 6710 Mobile Bridges.

" **NOTE:** *A Flooding Level Checklist starts on page 4-26.*

Frames are defined as follows:

Inbound: A frame flooded towards the distribution LAN (the Ethernet LAN segment containing the super root).

Outbound: A frame flooded away from the distribution LAN. A special case of outbound is *outbound to secondary (RS-485) LANs.*

Inbound

Flooding may be configured separately for unicast (single physical address) and multicast (group address) frame types. Many network protocols use multicast messages for establishing and maintaining connections, and use unicast messages for data exchange. Inbound options are:

```
Multicast      <Primary>
Unicast        <Disabled>
```

Multicast and Unicast options have the following settings:

```
Enabled
Primary
Disabled
```

Setting	Description
Enabled	6710 Mobile Bridge floods to all ports, similar to a conventional bridge.
Primary *(Multicast default)*	Frames are flooded inbound only. This setting is useful in many RS-485–Ethernet LAN installations where the super root, servers, or gateways for RS-485 stations are on the same Ethernet segment.
Disabled *(Unicast default)*	Frames are not flooded. Use this setting only if the Outbound to Secondaries option is also set to Disabled.

Outbound to Secondaries

Outbound to Secondaries floods frames with unknown destinations to secondary (RS-485) LANs. Settings are:

```
Enabled
Disabled
```

Setting	Description
Enabled	All designated 6710 Mobile Bridges flood to secondary (RS-485) LANs. This setting allows the super root to control flooding for all 6710 Mobile Bridges serving as designated devices for secondary (RS-485) LANs.
Disabled *(Multicast and Unicast default)*	Flooding is disabled in all designated 6710 Mobile Bridges. This setting allows the super root to control flooding for all 6710 Mobile Bridges serving as designated devices for secondary (RS-485) LANs. This setting should be used only if Inbound flooding is Disabled.

Flooding Level Checklist

You can use the following list of questions to determine the required flooding levels for the Inbound and Outbound to Secondaries options. The list is structured so that you can skip later questions as soon as you determine the appropriate flood level settings.

If your answer is "I do not know," go to the next question. If you cannot determine the appropriate flooding levels, use the higher (multicast) flooding levels.

" **NOTE:** *If relatively high flooding levels are required, you may need to set frame filters to reduce unnecessary traffic on the RS-485 LAN segments. In general, the need for filters increases with the amount of traffic on the distribution LAN and the flooding levels. Information about filtering starts on page 4-35.*

1. Do all stations on the RS-485 LAN segment routinely transmit a frame at least once every 4 minutes?

Answer	Settings
Yes	Inbound/Unicast/Disabled Outbound to Secondaries/Unicast/Disabled

2. Do any stations on the RS-485 LAN segment need to receive multicast or broadcast messages?

Answer	Settings
No	Inbound/Multicast/Enabled Outbound to Secondaries/Multicast/Disabled

Note: TCP/IP stations must receive broadcast ARP frames.

Comments:

The destination of a multicast frame is never known. The Disabled setting should be used for any network where RS-485 stations do not need to receive multicast frames. The Disabled setting can be used for secondary (RS-485) LANs that only need to receive ARP frames by enabling the ARP Server Mode to one of its flooding options (page 4-28).

3. Do stations on the RS-485 LAN segment communicate with other nodes on a different RS-485 LAN segment?

Answer	Settings
Yes	Inbound/Unicast/Enabled Inbound/Multicast/Enabled
No	Inbound/Unicast/Primary Inbound/Multicast/Primary Outbound to Secondaries/Unicast/Disabled Outbound to Secondaries/Multicast/Disabled

Comments:

The Enabled settings facilitate peer-to-peer applications, where stations on one RS-485 LAN segment communicate with stations on a different RS-485 LAN segment.

In general, the Primary setting is designed for client applications where stations on the RS-485 LAN segment communicate with server nodes on the distribution LAN.

4. Does the RS-485 LAN segment contains stations that do not periodically generate traffic, or do not include the OWL_ATTACH protocol as part of their RS-485 driver?

Answer	Settings
Yes	Inbound/Unicast/Primary Inbound/Unicast/Enabled

Comments:

You may need to enable unicast flooding if RS-485 stations do not periodically generate traffic. Occasional traffic is needed to maintain information in the forwarding database.

You can also use the Outbound to Secondaries/Unicast/Enabled setting to force unicast flooding to RS-485 stations.

ARP Server Mode

ARP Server Mode can convert multicast ARP requests to unicast ARP requests for RS-485 stations in the forwarding database. ARP Server Mode can significantly improve RS-485 network performance in busy IP networks.

Settings are:

```
Disabled
No Flooding
Delay Flooding
Normal Flooding
```

When ARP Server Mode is enabled, the IP addresses of the RS-485 stations are included in the forwarding database. The ARP server learns the IP addresses of RS-485 stations by monitoring IP traffic to those stations. Additionally, some stations may have the capability of explicitly registering IP addresses with the ARP server. This is done with the OWL_ATTACH protocol included in certain RS-485 drivers supplied by Intermec.

Setting	Description
Disabled	No special action is taken when an ARP is received. Multicast ARP requests are subject to the frame filters and the setting for MultiFloodLevel. Disabled is the preferred option when a system has no IP RS-485 traffic or has stations that do not register IP addresses.
No Flooding *(default)*	The ARP server converts ARPs from multicast destination addresses to the unicast address of the destination station. No Flooding is the most efficient configuration, since multicast ARPs are never forwarded. Use of this setting requires stations to register IP addresses with the 6710 Mobile Bridge.

Setting	Description
No Flooding *(Continued)*	No Flooding is the preferred option when RS-485 stations should respond to ARPs, and are capable of registering their IP addresses with the 6710 Mobile Bridge. This registration can be accomplished either through regular periodic traffic from the RS-485 station or use of the OWL_ATTACH protocol included in some RS-485 drivers supplied by Intermec.
Delay Flooding	The ARP server converts ARPs from multicast to the unicast address of the destination station. If the destination address is unknown, the initial ARP request is not forwarded. If the requesting device retries the ARP request, second and subsequent ARP requests are forwarded. Delay Flooding is the preferred option when RS-485 stations should respond to ARPs, but are not capable of registering their IP addresses with the 6710 Mobile Bridge.
Normal Flooding	The ARP server converts ARPs from multicast destination addresses to the unicast address of the destination station. If the destination address is unknown, the ARP request is flooded according to the multicast flood level settings. Normal Flooding is useful when RS-485 stations need to respond to ARP requests, but are not capable of registering IP addresses with the 6710 Mobile Bridge. Normal Flooding sends more unnecessary ARPs over RS-485 links then delay flooding. Normal Flooding does not introduce occasional delays in ARP responses as Delay Flooding does.

[Ports]

Use [Ports] to define options for the 6710 Mobile Bridge's Ethernet port and each RS-485 port.

```
   Name          MAC Address      Status   Hello Period
1  "omde"     (Unique address)  <Enabled> <2 Seconds>
2  "omd485a"  (Ethernet port's  <Enabled> <2 Seconds>
              MAC address)
3  "omd485b"  (Ethernet port's  <Enabled> <2 Seconds>
              MAC address)
```

" NOTE: *If no RS-485 PC card is installed internally in slot A or B, the Name is "" and Status is Disabled.*

The following chart defines options in the Name column.

Option	Description
omde	Ethernet port
omd485a *or* omd485b	RS-485 port

Select a port to display its options:

```
Name              (Depends on the port.)
MAC Address       00:00:00:00:00:00
Status            <Enabled>
Hello Period      <2 Seconds>
[Ethernet]
[B_B485]
```

Name, MAC Address, Status, and Hello Period appear for all ports. The remaining options appear as follows:

[Ethernet]	Appears if you selected "omde." Options start on page 4-33.
[B_B485]	Appears if you selected "omd485a" or "omd485b." Options start on page 4-45.

Name

The read-only Name setting displays the driver name for the type of device occupying this communication port. For example, “omde” is the driver name for the Ethernet port. The name is for internal system use.

MAC Address

MAC Address is a read-only option that displays the network address of the Ethernet port or RS-485 port. The 6710 Mobile Bridge automatically identifies the addresses of devices installed in or attached to its communication ports. The MAC address for each RS-485 port is the MAC address of the Ethernet port.

Status

The Status option sets the condition of the Ethernet port or RS-485 port. Settings are:

Enabled Disabled

Setting	Description
Enabled *(default)*	Port is available for use.
Disabled	Port is not available for use.

Hello Period

The hello period determines how frequently the 6710 Mobile Bridge broadcasts hello messages on the network. On Ethernet links, hello messages are used to maintain the spanning tree.

Settings for Hello Period are:

```
1 Second
2 Seconds
3 Seconds
```

You should never need to change the default of 2.

Ethernet Options

Use [Ethernet] to set Ethernet port options:

```
OWL Frame Type          <DIX>
Cable Type              <Auto Detect>
[Static Addresses]
[Normal RX Filter]
[Advanced RX Filter]
```

OWL Frame Type

OWL Frame Type is the Ethernet type for communication among 6710 Mobile Bridges. Settings are:

```
DIX
SNAP
```

Setting	Description
DIX *(default)*	Adds DIX (Ethernet 2.0) header to open wireless LAN frames. DIX is the default because it is the most commonly used Ethernet frame type.
SNAP	Adds an 802.3 SNAP header to open wireless LAN frames. In some cases, open wireless LAN frames must be encapsulated in SNAP frames.

Cable Type

Cable Type specifies the type of Ethernet medium to which the 6710 Mobile Bridge is connected. It is recommended that you explicitly set the cable type. Settings are:

```
10BaseT
10Base2
AUI
Auto Detect
```

Setting	Description
10BaseT	Selects the RJ11 connector (sets type to 10BASE-T, twisted pair). The cable type defaults to 10BASE-T if no traffic is heard on any Ethernet port (10BASE2, 10BASE5, or 10BASE-T) during a 10-second time window when the 6710 Mobile Bridge starts up.
10Base2	Selects the BNC connector (sets type to 10BASE2, thinnet).
AUI	Selects the AUI 15-pin D-sub connector (sets type to 10BASE5 thicknet, and other types).
Auto Detect *(default)*	Automatically selects cable type by listening for traffic on Ethernet ports during initialization. For this to work, the 6710 Mobile Bridge must be connected to the Ethernet medium during system start-up, and other devices on the Ethernet medium must be transmitting at least one frame every 10 seconds.

[Static Addresses]

Use [Static Addresses] to define a list of 20 or fewer permanent unicast 802 MAC addresses connected to this Ethernet port. The Static Address Table displays the addresses:

```
1  00.00.00.00.00.00
2  00.00.00.00.00.00
3  00.00.00.00.00.00
.
.
.
20    00.00.00.00.00.00
```

Select an address, then type 6 hexadecimal pairs for the new address at the prompt:

```
Range is:
6 hex pairs
```

Static addresses become permanent entries in the route table. This is useful when configuring designated 6710 Mobile Bridges for secondary (RS-485) LANs, since it reduces the need to flood frames to wired stations on the secondary (RS-485) LAN segment.

[Normal RX Filter]

Ethernet filters allow elimination of frame types that do not need to be forwarded to RS-485 stations. The main benefit of filtering is reduction in unnecessary RS-485 transmissions. Options are:

```
[Frame Types]
[SubTypes 1]
[SubTypes 2]
```

[Frame Types] allows filters to be established for common networking protocols such as IP, Novell IPX, and 802.2 LLC (Logical Link Control). Separate selections are available for each of the three Ethernet standards: DIX (Ethernet 2.0), 802.3, and 802.3 SNAP.

A filter may be configured to pass or drop all frames of a given type. Alternatively, filters may be set to operate on selected subtypes within each frame type category. [SubTypes 1] lists several *predefined* frame types as well as *user-defined* frame types. Settings under [SubTypes 2] allow additional *user-defined* frame subtypes to be specified. The default 6710 Mobile Bridge configuration passes all frame types.

[Frame Types]

[Frame Types] options are:

	Action	Scope
DIX-IP-TCP Ports	<Pass>	<Unlisted>
DIX-IP-UDP Ports	<Pass>	<Unlisted>
DIX-IP-Other Protocols	<Pass>	<Unlisted>
DIX-IPX Sockets	<Pass>	<Unlisted>
DIX-Other EtherTypes	<Pass>	<Unlisted>
SNAP-IP-TCP Ports	<Pass>	<Unlisted>
SNAP-IP-UDP Ports	<Pass>	<Unlisted>
SNAP-IP-Other Protocols	<Pass>	<Unlisted>
SNAP-IPX Sockets	<Pass>	<Unlisted>
SNAP-Other EtherTypes	<Pass>	<Unlisted>
802.3-IPX Sockets	<Pass>	<Unlisted>
802.2-IPX Sockets	<Pass>	<Unlisted>
802.2-Other SAPs	<Pass>	<Unlisted>

Frame Type	Description
DIX–IP–TCP Ports DIX–IP–UDP Ports SNAP–IP–TCP Ports SNAP–IP–UDP Ports	Primary Internet Protocol Suite (IP) transport protocols.
DIX–IP–Other Protocols SNAP–IP–Other Protocols	IP protocols other than TCP or User Datagram Protocol (UDP).
DIX–IPX Sockets SNAP–IPX Sockets 802.3–IPX Sockets	Novell NetWare protocol.
DIX–Other EtherTypes SNAP–Other EtherTypes	DIX or SNAP registered protocols other than IP or IPX.
802.2–IPX Sockets	Novell running over 802.2 LLC.
802.2–Other SAPs	SAPs other than IPX or SNAP.

" **NOTE:** *Some IP protocol ports cannot be filtered because they are used for configuration and management of the 6710 Mobile Bridge. These include HTTP, Telnet, SNMP, and Internet Control Message Protocol (ICMP). Filters set for these protocols are ignored for the Ethernet frame type configured in the 6710 Mobile Bridge's [Tcpip] menu.*

Frame types have the following settings:

```
Action    <Pass>
Scope     <Unlisted>
```

Setting	Description	
Action	Defines how the frame is processed:	
	Pass *(default)*	Frame is passed to the bridging function for further processing.
	Drop	Frame is discarded.

Setting	Description	
Scope	Defines whether the action applies to all frames of this type, or is restricted to selected subtypes:	
	Unlisted *(default)*	Applies only to subtypes that are not configured under [SubTypes 1] or [SubTypes 2].
	All	Applies to all frames of this type. [SubTypes 1] and [SubTypes 2] settings for this frame type are ignored.

[SubTypes 1]

The predefined subtypes in the [SubTypes 1] menu provide preconfigured filters that are useful in many networks. The values for these subtypes cannot be changed. Subtypes are:

	Action	SubType	Value
DIX-ARP	<Pass>	<DIX-EtherType>	08 06
SNAP-ARP	<Pass>	<SNAP-EtherType>	08 06
802.2-IPX-RIP	<Pass>	<802.2-IPX-Socket>	04 51
802.2-IPX-SAP	<Pass>	<802.2-IPX-Socket>	04 53
NNL	<Pass>	<DIX-EtherType>	87 5b
NETBIOS	<Pass>	<802.2-SAP>	f0 f0
1	*(User-defined subtypes.)*		
(through)			
16	*(User-defined subtypes.)*		

User-Defined Subtypes in [SubTypes 1] and [SubTypes 2]

The value under *user-defined subtypes* allow individual protocol ports, sockets, or SAPs to be specified for each of the listed frame types. The filter takes action if either the source or destination fields in the frame match the specified port, socket, or SAP. A value of 00 00 denotes the subtype as *Unlisted*.

Subtypes for [SubTypes 2] are:

```
  Action      SubType              Value
1 <Pass>  <DIX-IP-TCP-Port>   00 00
2 <Pass>  <DIX-IP-TCP-Port>   00 00
3 <Pass>  <DIX-IP-TCP-Port>   00 00
.
.
.
22 <Pass>  <DIX-IP-TCP-Port>   00 00
```

Subtype	Value
DIX–IP–TCP–Port	Port value in hexadecimal.
DIX–IP–UDP–Port	Port value in hexadecimal.
DIX–IP–Protocol	Protocol number in hexadecimal.
DIX–IPX–Socket	Socket value in hexadecimal.
DIX–EtherType	Specify the registered DIX type in hexadecimal.
SNAP–IP–TCP–Port	Port value in hexadecimal.
SNAP–IP–UDP–Port	Port value in hexadecimal.
SNAP–IP–Protocol	Port value in hexadecimal.
SNAP–IPX–Socket	Socket value in hexadecimal.
SNAP–EtherType	SAP in hexadecimal. To filter on both SAP and OUI (Organizationally Unique Identifier), use advanced filters.
802.3–IPX-Socket	Socket value in hexadecimal.
802.2–IPX-Socket	Socket value in hexadecimal.
802.2–SAP	SAP in hexadecimal.

" **NOTE:** *Port values may be entered in decimal by adding a period to the entry. For example, "23." for port 23. The Value field displays the hexadecimal equivalent.*

[Advanced RX Filter]

If you need more flexibility than that provided by [Normal RX Filter], you can use the tables for [Advanced RX Filter] to specify additional filters. Settings for [Advanced RX Filter] execute after those for [Normal RX Filter]. For example, if [Normal RX Filter] dropped a frame, the frame cannot be "undropped." If [Normal RX Filter] passed a frame, [Advanced RX Filter] then executes.

Specifying an advanced filter for [Advanced RX Filter] is more complicated than specifying one for [Normal RX Filter]. If possible, use [Normal RX Filter] to set filters.

[Advanced RX Filter] options are:

```
[Expressions]
[Values]
```

[Expressions]

Use [Expressions] to enter expressions used to match the patterns stored in pattern lists to consecutive bytes in received Ethernet frames. Settings for Expressions execute in sequence until a determination is made to pass or drop the frame, as follows:

- If the last Then listed in this table is Then Drop, the table's default is an implied Else Pass.
- If the last Then is Then Pass, the default is an implied Else Drop.

The Value Table displays the filter expressions to be executed:

```
ExprSeq Offset      Mask    Op     Value Id  Action
1       0           0         <EQ>                0 <And>
2       0           0         <EQ>                0 <And>
3       0           0         <EQ>                0 <And>
.
.
.
22      0           0         <EQ>                0 <And>
```

Filter expressions have the following settings:

```
ExprSeq          0
Offset           0
Mask
Op               <EQ>
Value Id         0
Action           <Pass>
```

ExprSeq

ExprSeq contains a sequence number that orders expressions in ascending order. It is a method of changing the sequence execution. The prompt is:

```
Range is:
0..65535
```

The default is 0. Change these numbers as needed for reordering. After you save the changes (through the Write command), the statements are physically reordered and renumbered.

Offset

This setting defines the offset in a received Ethernet frame to match the patterns. The prompt is:

```
Range is:
0..65535
```

A frame matches a pattern list if the masked bytes at the specified offset in the frame match any of the masked patterns in the pattern list. The default is 0.

Mask

This setting indicates the bits that are significant at the specified offset. The prompt is:

```
Range is:
8 hex pairs
```

The default is "" *(an empty string).*

The length of this mask determines the number of characters compared at the offset. If this field is "" *(an empty string*, the default*)*, the length of the field is determined by the longest value in the Value Table with the matching Value Id.

Op

Op is a memory comparison operator in the following chart.

Operator	Description
LT	Less than one value.
LE	Less than or equal to one value.
EQ *(default)*	Equal to any in the list.
NE	Not equal to any in the list.
GE	Greater than or equal to one value.
GT	Greater than one value.

Value Id

The field at the specified offset is compared with values in the Value Table with the Value Id. The prompt is:

```
Range is:
0..255
```

The default is 0. When using a comparison operator that requires a single value (LT, LE, GE, and GT), only the first value found will be compared.

Action

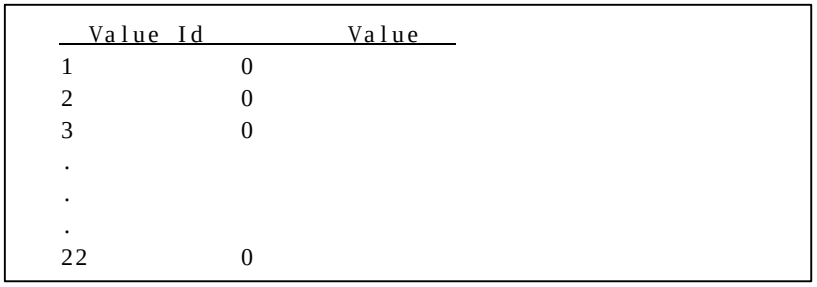
The Action setting instructs the Ethernet driver and indicates what should happen when this expression is true. Settings are:

```
And
Pass
Drop
```

Setting	Description
And	Instructs Ethernet driver to continue with the next simple expression, if the expression condition is satisfied. Two or more simple expressions are ANDed together to form a complex expression.
Pass *(default)*	Instructs Ethernet driver to accept the frame for further processing and pass the frame up to the bridging layer (the frame is not discarded).
Drop	Instructs Ethernet driver to reject the frame.

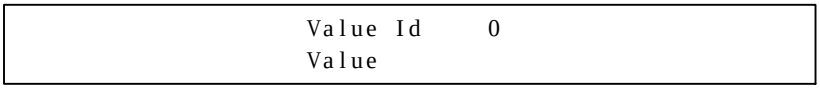
[Values]

Use [Values] to enter pattern lists that contain byte patterns that match consecutive bytes in received Ethernet frames. Settings for [Values] are referenced by the Value Id from the Expression Table. The values to be used in a filter expression are as follows:



Value Id	Value
1	0
2	0
3	0
.	
.	
.	
22	0

Filter expression values have the following settings:



Value Id 0
Value

Setting	Description
Value Id	An identifier used by an expression in the Expression Table. The range is 0 *(default)* to 255. All values with the same identifier are considered to be in the same list. When used in an expression that allows only one value (that is, LT, LE, GE, or GT), only the first value in the list is used.
Value	One of the values to be compared. The range is 8 hexadecimal pairs.

RS-485 Options

Use [B_B485] to set the baud rate of the RS-485 port:

```
Baud Rate        <115200>
```

Options are:

```
115200
460800
```

The default is 115200K baud. Use the following chart as a guide to setting the baud rate.

Stations	Baud Rate
PEN*KEY® 6210 Computers	115200
PEN*KEY 6100 Computers	115200
PEN*KEY 6110 Computers	115200 or 460800

Security Options

Use [Security] to set these passwords:

```
Password              "************"
Service Password      <Enabled>
```

Password

This option is the top-level password you need to access the configuration menus. The prompt is:

```
Range is:
16 chars
```

Enter 16 or fewer alphanumeric characters for this password. It is case insensitive and can be any combination of letters, numbers, and symbols. For security, the password appears as asterisks on the screen.

Service Password

Intermec maintains a service password so its Customer Response Center can configure this 6710 Mobile Bridge if necessary. For example, if you forget what the configuration menus' top-level password is, the Customer Response Center can access the menus through the service password.

By default, the Service Password is enabled. If setting a service password violates your security guidelines, you can disable it. If it is already disabled and you forget the configuration menus' password, you may need to send this 6710 Mobile Bridge to a Service Center to be reconfigured.

Contact the Customer Response Center for more information about the service password. See the Preface for contact information.

Creating a Web Browser Session

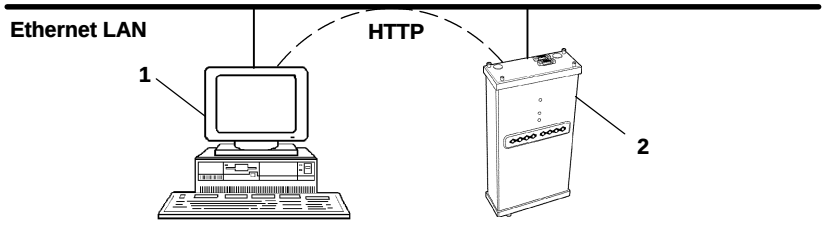
▪ **NOTE:** *The 6710 Mobile Bridge's configuration menus are designed for HTML Level 2.0 or higher.*

Before you can configure the 6710 Mobile Bridge through a Web browser, you must connect the device to the Ethernet cable. (See Section 3, "Installation," for help.) You must also perform initial configuration through the DIAG port to:

- Set an IP address or DHCP server name. You should also configure a subnet mask and IP router address.
- Set the Ethernet cable type.

" **NOTE:** *The 6710 Mobile Bridge includes an autodetect feature that senses the Ethernet medium if traffic is present. If no traffic is present on the cable, the system software defaults to 10BASE-T. For most installations, it is recommended that you explicitly set the Ethernet type.*

The 6710 Mobile Bridge must go through its boot sequence before you can configure it through the browser. If you reboot it while you are configuring it through the browser, the session terminates. You can create a new session after the 6710 Mobile Bridge reboots. To establish a browser session, see Figure 4-3 and the procedure following it.



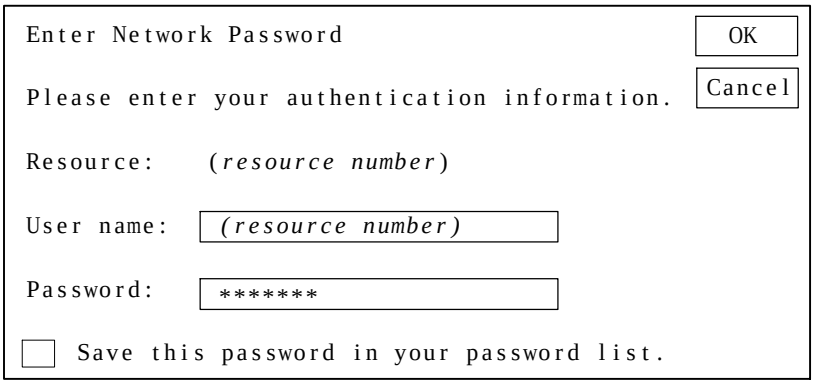
1. **Platform with Web browser**
2. **6710 Mobile Bridge**

Figure 4-3
Web Browser Session

1. Ensure the 6710 Mobile Bridge is connected to the Ethernet cable and has an assigned IP address.
2. Ensure the Web browser is installed on the platform.
3. Start the browser application.
4. Enter the 6710 Mobile Bridge's IP address in the browser's Uniform Resource Locator (URL) field. For example:

   http://999.999.99.99

After you enter the correct IP address, the Enter Network Password screen appears:



5. Enter the user name and password. Refer to the following chart for help with all fields.

Setting	Description
Resource	A number assigned to this session. The number changes every time you access the Enter Network Password screen.
User name	The Resource number. For example, if the Resource number is 3550, the user name is 3550.
Password	Configuration menus' top-level password. The default password is CR52401 (case insensitive). For security, the password appears as asterisks on the screen.
Save this password in your password list	If you check this option, the browser saves the password. You do not need to select this option because the Resource number changes every time you access this screen.

" **NOTE:** *Only one computer at a time can access the configuration menus. If you unexpectedly receive a request for the user name and password, another user may be trying to view or modify the configuration through Telnet or the DIAG port. If a different computer tries to start another Web browser session, it is refused until the current session logs out.*

Following is the initial screen that appears after you enter the correct user name and password. Configure the 6710 Mobile Bridge by following the directions provided on this screen and the help screens. In addition, this user guide's Index lists the page numbers for all menu options.

Intermec 6710 Mobile Bridge Configuration

- **F** Configuration Menus
- **F** Review/write Changes
- **F** Logout

- **F** Reboot

Security

Once you've successfully entered the user name and password into your browser, you are authorized to access the configuration. If five (5) minutes elapses without configuration activity, you will become unauthorized again.

Only one computer at a time is allowed to be authorized for access to the configuration menus. If you unexpectedly receive a request for user name and password, it's possible that another user is currently attempting to view or modify the configuration via telnet or the serial port. An attempt to start another HTTP session from another computer will be refused until this session has been logged out.

How to use the Configuration Menu

1. Select Review/write changes to see the existing changes from the defaults. Optionally select an Undo option to remove changes.
2. Locate and change configuration items by navigating in the Configuration Menu.
3. Select the Submit Changes button to submit the changes on each form.
4. Repeat the above steps until all changes have been submitted.
5. Select Review/write changes and review the changes you have made.
6. Select Commit to write the changes.

Note: Some early browser versions can't be convinced by the server to always get a fresh copy of a document that was previously fetched. This might cause old and misleading information to be displayed. Updating the browser, pressing the reload button, or disabling the browser's cache may be helpful.

Configuration Guidelines

Planning Your Installation

To plan your installation, refer to "Finding the Best Location" in Section 3 for installation guidelines.

Using the Configuration Guide

The configuration guide in Table 4-1 summarizes the necessary steps to set up an RS-485–Ethernet LAN. Table 4-1 provides basic setup information for a simple network using 6710 Mobile Bridges on a single Ethernet segment. This type of network may contain bridges or hubs.

Table 4-1
Configuration Guide

z = required step; **Z** = recommended step

1. Configuration Planning

- **z** If more than one 6710 Mobile Bridge will be installed on this Ethernet LAN segment, select which 6710 Mobile Bridges will be primary and fallback super root candidates *(page 4-22)*.

2. Preliminary Configuration Before Installation

- **z** Configure 6710 Mobile Bridges with IP address configuration *(page 4-14)* or the DHCP server name *(page 4-18)*.
- **Z** Configure the Ethernet port cable type *(page 4-34)*.
- **Z** For the OWL frame type, use DIX (the default) *(page 4-33)*.
- **z** If more than one 6710 Mobile Bridge will be installed on this Ethernet LAN segment, configure the root priority in super root candidates to specify primary and fallback 6710 Mobile Bridges. Set all other 6710 Mobile Bridges to root priority 0. *(Page 4-22.)*

  " **NOTE:** *Set the highest root priority in the primary super root, and lower root priorities in no more than two fallback super roots.*

3. Additional Configuration

- **z** Set Global Flooding parameters in super root candidates *(page 4-24)*.
- **z** Configure the baud rate for each RS-485 port (page 4-45).
- **Z** Enable the Proxy ARP Server if RS-485 IP stations are being supported *(page 4-28)*. (The default is No Flooding.)
- **Z** Set Ethernet filters to optimize frames forwarded to RS-485 stations *(page 4-35)*.

Section 5

Software Download

This section describes the file system structure, File Menu commands, and the ROM command monitor for the 6710 Mobile Bridge.

File System Structure

The 6710 Mobile Bridge's file system has four separate segments (analogous to a directory in most computer file systems).

Boot Segments 1 and 2

The first two segments (1 and 2) are .75 MB boot segments. Either boot segment can hold the bootable (executable) FLASH file MSD29.BIN, which loads when you reboot the 6710 Mobile Bridge. You can store different versions of MSD29.BIN in the boot segments and then configure the 6710 Mobile Bridge to use one of them.

Data Segments 3 and 4

The next two segments (3 and 4) are .25 MB data segments. Either data segment can hold the self-extracting HTML configuration files.

Active and Inactive Segments

The 6710 Mobile Bridge has an active boot and data segment, as well as an inactive boot and data segment:

- The inactive segment is where you can download a new file.
- The active segment contains the files that are loaded at boot time. An active boot segment pointer and an active data segment pointer point to the appropriate "active" segments. The segment not pointed to by one of these "active" pointers is the inactive boot or data segment.

After you load an inactive segment with a new file, you can change the "active" pointers to the segment that holds the new file. You then reboot the 6710 Mobile Bridge so the changes take effect. At this point the following occur:

- The 6710 Mobile Bridge is running the new version of software.
- The segment holding the new files is now the "active" segment.
- The old version of software is in an inactive segment.

▪ **NOTE:** *If the active segment is empty when you reboot the 6710 Mobile Bridge, you must establish a new session through the DIAG port to reload the unit with software.*

RAM Segment

The file system supports a fifth segment known as the RAM segment. The RAM segment is similar to the other segments, except the file contents are stored in RAM and the segment's contents are lost when you reboot the 6710 Mobile Bridge.

The RAM segment is limited to a maximum of 4096 bytes. It is used to hold small script files during the software download process.

When you view the file directory, the program currently executing displays as if it were in the RAM segment. This program, however, is not really a part of the RAM segment. You cannot delete or erase it, and TFTP commands cannot read or write to it. It displays as part of the file directory so you can determine which version of software is running.

Segment Names

You must enter a segment for most 6710 Mobile Bridge file system commands. You can type the numeric digits (1, 2, 3, or 4) corresponding to the respective file segments, or you can use the following mnemonics (the 6710 Mobile Bridge translates them to a segment number):

Mnemonic	Description
AB	Active boot segment.
IB	Inactive boot segment.
AD	Active data segment.
ID	Inactive data segment.
RAM	RAM segment.

File Names

Several file system commands require you to enter file names. You can precede file names by a segment number or name followed by a colon.

EXAMPLE 1: AB:MSD29.BIN refers to the file MSD29.BIN in the active boot segment (segment 1 or 2).

EXAMPLE 2: 1:MSD29.BIN refers to the file MSD29.BIN in segment 1.

If you omit the segment number or segment name, the 6710 Mobile Bridge searches the segments in this order until it finds a file matching the file name:

RAM, 1, 2, 3, 4

Downloading Programs

You can download new programs to the 6710 Mobile Bridge while it is operating. The unit has two program FLASH directories so that if an issue exists with the download of the new FLASH, the system can reboot to the previous version. An internal timer allows the activation of the new software program to be immediate or activated later.

File Menu Commands

Commands for software download and other processes are located on the File Menu. To display the commands, type the following at the Main Menu prompt:

>file

The File Menu appears:

Loading configuration from EEPROM

Command	Description
Fb	fb <boot segment> <data segment>
Fd	fd (<segment> | all) - directory list
Fdel	fdel <filename> - delete file
Fe	fe (<segment> | all) - erase segment(s)
Tftp	File transfer
Script	Execute script files
SDVars	Software download variables
Exit	Return to main menu

File>

Fb Command

Use Fb to make inactive segments active. The format is:

File>fb <boot segment> <data segment>

- <boot segment> is the name or number of the boot segment to be activated. Boot segments are 1 and 2, or AB (active boot) and IB (inactive boot).
- <data segment> is the name or number of the data segment to be activated. Data segments are 3 and 4, or AD (active data) and ID (inactive data).

EXAMPLE 1: This command makes segment 1 the active boot segment:

File>fb 1

EXAMPLE 2: This command makes segment 1 the active boot segment and segment 4 the active data segment:

```
File>fb 1 4
```

You can use an asterisk (*) in place of either <boot segment> or <data segment> to tell the 6710 Mobile Bridge to not change that segment. For example, this command leaves the active boot segment unchanged and changes the active data segment to 4:

```
File>fb * 4:
```

This can also be accomplished by:

```
File>fb ab: 4:
```

" **NOTE:** *Colons are optional but you can use them for better command consistency.*

Fd Command

Use Fd to display the FLASH file system directory, including information about the boot file. For example:

```
Boot File=MSD29.BIN      <FLASH boot file>
Boot Address=250ef0      <boot file's starting address>
Boot Segment=1           <active boot segment>
Data Segment=3           <active data segment>

File Directory:  seg  type  length   date      time      ver
MSD29.BIN        R    E     279299   03-17-98  15:25:58  v01.31
MSD29.BIN        1    E     279299   03-17-98  15:25:58  v01.31
```

Following are field descriptions:

- "File Directory name" lists the names of all files currently loaded in FLASH.
- "seg" is the segment in which the boot file is loaded. (R indicates the RAM segment.)
- "type" is the type of file: E for executable (boot file), D for data.
- "length" is the file size in bytes.
- "date" and "time" are the date and time the file was created.
- "ver" is the file version number in the format vxx.xx.

You should use the Fd command often to ensure that the correct version of FLASH file MSD29.BIN is in the active boot segment.

" **NOTE:** *If the active segment contains no files when you reboot the 6710 Mobile Bridge, the unit enters the ROM command monitor and you lose the ability to Telnet to it during this session. In this case you must access the unit through its DIAG port to correct the problem.*

Fdel Command

Fdel deletes the file name from the 6710 Mobile Bridge file system. When you delete a file, it is marked as invalid but remains in the file system. To reclaim the space from a deleted file, you must erase the segment in which the file resides.

The command's format is:

```
File>fdel <file name>
```

EXAMPLE: This command deletes the file MSD29.BIN saved in the inactive boot drive:

```
File>fdel ib:MSD29.BIN
```

Fe Command

Fe erases files in a specified segment of FLASH memory. Once you have erased the files, you can restore them only by reloading them from another source. The command's format is:

File>fe <segment>

<segment> is a segment number, a segment name, or the word "all." Specifying "all" erases all FLASH file segments but does not erase the RAM segment.

EXAMPLE 1: This command erases FLASH segment 1:

```
File>fe 1:
```

EXAMPLE 2: This command erases the inactive boot segment:

```
File>fe ib:
```

TFTP Command

Use the Tftp command to display the following screen:

```
Argument   Description
Get        Get <host IP addr> <foreign File> <local File>
Put        Put <host IP addr> <foreign File> <local File>
Server     Start/Stop/Query TFTP Server
?          Display this help

File>
```

A 6710 Mobile Bridge (client) can obtain files from a TFTP server. The server may be one 6710 Mobile Bridge configured to act as the server, or another device on the network. The server must operate in octet (8 bit) mode.

- As a server, the 6710 Mobile Bridge can service read and write requests from a 6710 Mobile Bridge client. To operate as a TFTP server, the 6710 Mobile Bridge must be loaded with these software versions:
  - ROM version 1.14 or greater
  - FLASH (MSD29.BIN) version 1.31 or greater
- As a client, the 6710 Mobile Bridge can read files from and write files to any TFTP server on the network. The client always requests octet mode.

In general, TFTP client sessions should fail only if the server is not responding because it is busy serving other clients or because it has not been started. In either case, the 6710 Mobile Bridge back-off algorithm should prevent excessive network traffic when many 6710 Mobile Bridges are trying to contact a TFTP server. When you type TFTP client commands at the command line, the 6710 Mobile Bridge does not retry failed transfers.

" NOTE: *Near the end of this section is a detailed example of how to use TFTP to upgrade a 6710 Mobile Bridge with a new version of FLASH. The example incorporates most of the TFTP commands.*

TFTP Server

Use the Server command to display TFTP commands. The format is:

```
File>tftp server
```

These commands are supported:

```
Help for Server command:
Argument    Description
Start       Start TFTP server
Stop        Stop TFTP server
Log         Display TFTP server message log
?           Display this help

File>
```

Server Start

Use Server Start to enable the 6710 Mobile Bridge as a TFTP server. The format is:

```
File>tftp server start
```

After you issue this command, the 6710 Mobile Bridge responds to TFTP client requests directed to its IP address. When acting as a server, the 6710 Mobile Bridge TFTP supports up to four concurrent TFTP sessions.

Server Stop

Use Server Stop to stop the 6710 Mobile Bridge from being a TFTP server when you are done transferring files. The format is:

```
File>tftp server stop
```

After you issue this command, the 6710 Mobile Bridge no longer responds to TFTP client requests. Current TFTP sessions with the server are completed, however.

Server Log

Server Log saves a history of TFTP client requests:

```
File>tftp server log
```

The TFTP server log contains useful TFTP server status information starting from when you set up the server. You must reboot the 6710 Mobile Bridge to clear the log.

TFTP Client Commands

The TFTP client in the 6710 Mobile Bridge supports standard Get and Put commands.

Get

Use Get on a 6710 Mobile Bridge client to download software from a TFTP server (a PC or another 6710 Mobile Bridge). The format is:

```
File>tftp get <ip address> <foreign file name>
<local file name>
```

- <ip address> is the IP address of the server (or "*" which indicates the value of the ServerIpAddress variable, described later in this section).
- <foreign file name> is the name of the file to get from the server. The file name can contain directory path information and must be in the format required by the server's operating system.

  The file must have an appropriate 29K file header. Boot files and data files are normally delivered with the proper file header attached, but script files you create must have the file header appended before transfer to a 6710 Mobile Bridge.
- <local file name> is the name of the file to be stored in the 6710 Mobile Bridge. The name must include a segment number or name followed by a colon and an optional file name. If only the segment name is supplied, the file name is set equal to the file name embedded in the file header.

EXAMPLE 1: This command line gets file MSD29.BIN from a directory on a PC server with IP address 1.2.3.4, and stores the file in the 6710 Mobile Bridge's inactive boot segment:

```
File>tftp get 1.2.3.4 c:\flash\ap\MSD29.BIN ib:
```

EXAMPLE 2: This command line gets file MSD29.BIN from segment 2 on the 6710 Mobile Bridge server with IP address 1.2.3.4, and puts the file in segment 1 on the 6710 Mobile Bridge client:

```
File>tftp get 1.2.3.4 2:MSD29.BIN 1:
```

Put

Use Put on a 6710 Mobile Bridge client to copy a file to the server (a PC or another 6710 Mobile Bridge). The format is:

```
File>tftp put <ip address> <foreign file name>
<local file name>
```

- <ip address> is the IP address of the server, or "*" which stands for the value of the ServerIpAddress variable (described on page 5-18).
- <foreign file name> is the name of the file as it will appear on the server. The file name can contain directory path information and must be in the format required by the operating system running on the server.
- <local file name> is the name of the file to be sent from the 6710 Mobile Bridge client.

EXAMPLE: This command takes boot file MSD29.BIN saved in the active boot drive on the 6710 Mobile Bridge client, and stores it in the active boot segment on the 6710 Mobile Bridge server with IP address 1.2.3.4:

```
File>tftp put 1.2.3.4 MSD29.BIN ab:
```

Script Command

You can initiate an automatic software download through SNMP by setting the appropriate server IP address and script file name, and then setting the software download time.

The following two commands execute automatically as if you had typed them on the command line:

```
File>tftp get * <remote script file name>
ram:sysswd1
File>script ram:sysswd1
```

The first command establishes a TFTP session with the server, gets the script file from the server, and places the file in the RAM file segment, giving it the name "sysswdl." The second command runs the script from the RAM segment.

The script file should contain the same commands you would use from the command line to erase the appropriate file segments, download the new file(s), and reboot using the new software. To test the script file manually to ensure it functions, log onto a 6710 Mobile Bridge and type the script file commands from the command line.

Creating Script Files

Script files are ASCII text files with a 32-byte file system header appended. The total file size including the 32-byte header must be less than the 4096 bytes to fit into the RAM file segment.

Each script file line must be fewer than 80 characters and be terminated by a line feed or a carriage return. Only one command is permitted per line. Spaces on a line are insignificant; commands and arguments may be preceded by any number of tabs or spaces as long as the total line length is fewer than 80 characters.

The script files can contain comments, designated by the "#" character. All characters on a line after a "#" are ignored.

FHDR29K.EXE appends the file system header to the script file. Assuming the ASCII text script file is named SCRIPT.TXT, the following line appends the appropriate file header and places the output in file SCRIPT.DAT:

```
FHDR29K -d -v1.00 SCRIPT.TXT SCRIPT.DAT
```

- "-d" marks the file as data instead of executable. This prevents the 6710 Mobile Bridge from trying to execute the file.
- "-v1.00" sets the file's version to 1.00. The file type and version appear in the directory information on the 6710 Mobile Bridge.

Sample Script File

```
#This sample script file assumes the server IP address
#has been set either from the command line or via SNMP.
#It also assumes MSD29.BIN and the HTML configuration
#files can be accessed from the server using no path
#information.

#Erase the inactive file segments.
file fe ib: #inactive boot segment
file fe id: #inactive data segment

#Get the new files into the inactive segments.
file tftp get 1.2.*.4 c:\flash\ap\msd29.bin ib:
file tftp get 1.2.*.4 c:\flash\ap\bkgrnd.29k id:
file tftp get 1.2.*.4 c:\flash\ap\help.29k id:
file tftp get 1.2.*.4 c:\flash\ap\hlp.29k id:
file tftp get 1.2.*.4 c:\flash\ap\intrmec.29k id:
file tftp get 1.2.*.4 c:\flash\ap\menu.29k id:
file tftp get 1.2.*.4 c:\flash\ap\mswelcom.29k id:
file tftp get 1.2.*.4 c:\flash\ap\write.29k id:

#Make the inactive segments active.
file fb ib: id:

#Reboot so changes take effect.
reboot
```

Script File Command Summary

Following is a description of the commands you can include in a download script file. You can issue these commands manually from the 6710 Mobile Bridge from the ">" prompt on the command line.

Most script file commands are executed from within the file command submenu. You or the script file can issue these commands in either of two ways:

- Use the File command to descend into the file submenu level where you can execute file system commands, until you use the Exit command to return to the ">" prompt.
- You can preface any file level command with the word "File," which causes the command processor to execute one command in the file command level and return to the ">" prompt.

For example, the command sequences in the following two charts are equivalent:

Command Sequence 1	Description
file	Descend to the "File>" command prompt.
fe ib:	Erase the inactive boot segment.
fe id:	Erase the inactive data segment.
exit	Return to the ">" prompt.

Command Sequence 2	Description
file fe ib:	Erase the inactive boot segment.
file fe id:	Erase the inactive data segment.

In addition, all commands are case insensitive, so:

FILE FE ID:

is the same as:

file fe id:

TFTP Client Command Retry

When executing a script file, the 6710 Mobile Bridge retries TFTP client commands GET and PUT until the command completes successfully. If the first attempt to transfer the file fails, the 6710 Mobile Bridge retries after a one-minute delay. With each successive failure, the retry time doubles until it reaches 8 minutes. Once this limit is reached, it remains at 8 minutes until the command completes.

Reboot Command

The Reboot command (located on the Main Menu) forces the 6710 Mobile Bridge to restart immediately. If it is issued within a script file, commands following it are not executed. If used, Reboot should always be the last command in the file.

Because Reboot causes an immediate reboot before the script file processing is completed, the software download status is not updated to accurately reflect the result of the software download. While this has no adverse effect on 6710 Mobile Bridge operation, you can not tell whether the download completed successfully without examining the version numbers of the active boot and data files after the 6710 Mobile Bridge has rebooted.

The preferred method of rebooting after a script file has completed is to set the next power up time variable. This should be the last thing the script file does, to allow the script file processing to complete and the software download status variable to be updated before the reboot occurs.

SDVars Command

Use the SDVars command to display the following arguments:

```
Argument  Description
Get       Get <software download object>
Set       Set <software download object> <value>
?         Display this help

File>
```

Use the Set argument with a range of software download variables. To display the variables, type:

```
File>sdvars set
```

The following variables are supported:

```
Argument          Description
ServerIpAddress   serveripaddress <ip address>
ScriptFilename    scriptfilename <filename> - filename can include path
StartTime         starttime <dd:hh:mm:ss> - days:hours:minutes:seconds
Status            status is read-only
CheckPoint        checkpoint <value>
Terminate         terminate - stop the current software download
SetActivePointers setactivepointers (none | boot | data | both)
NextPowerUpTime   nextpoweruptime <dd:hh:mm:ss> - days:hours:minutes:second
?                 Display this help

File>
```

Use the Get argument to display the value you enter for a variable.

ServerIpAddress

ServerIpAddress contains the IP address of the TFTP server to use to retrieve the download script file. This address is also used when you specify an asterisk as the IP address of the `tftp get` or `tftp put` command.

The format of the ServerIpAddress variable is:

```
File>sdvars set serveripaddress <ip address>
```

EXAMPLE: This command line sets the IP address of the server to 1.2.3.4:

```
File>sdvars set serveripaddress 1.2.3.4
```

ScriptFilename

ScriptFilename contains the file name of the script to be retrieved from the TFTP server. The file name can contain directory path information and must be in the format required by the operating system running on the server.

The format of the ScriptFilename variable is:

```
File>sdvars set scriptfilename <foreign file name>
```

EXAMPLE: This command line sets the script name to SCRIPT.DAT:

```
File>sdvars set scriptfilename script.dat
```

StartTime

StartTime is a relative time at which to begin the software download process. The value of this variable is how long into the future the 6710 Mobile Bridge will begin the software download process by downloading the script file. If you do not want to start the software download process after setting this variable, you can set this variable to zero.

As long as the StartTime has not counted to zero on its own, the timer stops and the software download process halts. When the timer does count down to zero, it uses the ServerIpAddress value and the ScriptFilename value to get the script file. If either of these is not set, an error is noted in the status variable and the software download process is aborted.

The variable's format is:

```
File>sdvars set starttime <dd:hh:mm:ss>
```

EXAMPLE: This command line sets the download start time to begin in 5 minutes:

```
File>sdvars set starttime 00:00:05:00
```

Status

Status is a read-only variable set by the software download mechanism to indicate whether the download completed successfully.

CheckPoint

CheckPoint is a numeric variable that checks on the progress of an active download in a 6710 Mobile Bridge. By setting CheckPoint to a different value after each command in the script file, you can read the value to determine how far the 6710 Mobile Bridge progressed through the script file. You can also test for failure if a script file aborts. The variable's format is:

```
File>sdvars set checkpoint <value>
```

For example, consider the following script file commands:

```
file sdvars set checkpoint 1
file fe ab:
file sdvars set checkpoint 2
file tftp get * MSD29.BIN ab:
file sdvars set checkpoint 3
reboot
```

When the software download is started, you can use SNMP to query its progress by reading the checkpoint variable. If the variable has a value of 2, for example, you know that the 6710 Mobile Bridge is trying to execute the `tftp get` statement. If the value is 3, you know the script has completed and the reboot statement was executed.

Terminate

Use Terminate to stop the download process in a 6710 Mobile Bridge. The variable's format is:

`File>sdvars set terminate <value>`

If StartTime is counting down, setting this variable stops the timer, halting the software download process.

Use caution with this variable. If the script file is being downloaded, or the commands in the script file are being executed, setting this variable interrupts the processing at its current location, halting the software download process. This can leave the 6710 Mobile Bridge in an unknown state.

It is your responsibility to determine this state and take any corrective action necessary. If the NextPowerUpTime variable is counting down, setting this variable stops the timer, halting the reboot process.

SetActivePointers

Use SetActivePointers to make inactive 6710 Mobile Bridge segments active, but only immediately before rebooting. The variable's format is:

```
File>sdvars set setactivepointers
<none|boot|data|both>
```

The default value is "none." Setting the value to "boot" or "data" affects only the given segment. Setting the value to "both" changes both segments. When the NextPowerUpTime variable counts down to zero, this field is checked to see if any inactive segments should be made active. This value resets to its default of "none" whenever the 6710 Mobile Bridge reboots.

NextPowerUpTime

NextPowerUpTime is a relative time at which to reboot the 6710 Mobile Bridge. The variable's format is:

```
File>sdvars set nextpoweruptime <dd:hh:mm:ss>
```

The value of this variable is how long into the future the 6710 Mobile Bridge should reboot itself. If you do not want to reboot the 6710 Mobile Bridge after setting this variable, you can set this variable to zero.

As long as the NextPowerUpTime variable has not counted to zero on its own, the 6710 Mobile Bridge does not reboot. When the timer does count down to zero, it checks the value of the SetActivePointers variable, takes the appropriate action as described previously, and then reboots the 6710 Mobile Bridge.

ROM Command Monitor

Certain functions available through the ROM command monitor can erase your configuration information. Intermec STRONGLY RECOMMENDS that you only use this option when absolutely necessary (for example, to upgrade your FLASH software or when instructed to do so, and under the supervision or direction of qualified Intermec personnel).

Starting the Command Monitor

You can access the ROM command monitor only through the DIAG port. Start the command monitor by turning the 6710 Mobile Bridge off, then back on. After the 6710 Mobile Bridge has completed its power-up self tests, you have about 5 seconds to open the ROM command monitor by pressing any key on the PC keyboard. See "Creating a Local DIAG Port Session" in Section 4 for more information on how to access ROM mode.

▪ **NOTE:** *You cannot invoke the command monitor once the 6710 Mobile Bridge has started the FLASH program. If the 6710 Mobile Bridge enters its boot sequence, you have to reboot it to open the ROM command monitor.*

When the ROM command monitor opens, the following displays on the PC:

```
QXS6700K Vx.xx  MMM DD YYYY
ap>
```

- QXS6700K is the program name of the ROM.
- x.xx is the version of the ROM command monitor.
- MMM DD YYYY is the month, day, and year the version was released.
- ap> is the command prompt.

Viewing ROM Commands

To view ROM commands, type any invalid command (such as "?") to display the command monitor's Main Menu:

```
B       - Reboot                    FR      - Run Flash Boot File
FX s    - Ymodem File Download      PWD     - Password Menu
FC s    - Move file to Flash        NPWD    - Norand Password Menu
FD      - File System Directory     SR z    - Serial Baud Rate

ap>
```

The following paragraphs describe each option.

" **NOTE:** *When executing a command that has an option (for example FX s) separate the option from the command by a space.*

B

Reboot resets the 6710 Mobile Bridge's system software. Reboot is similar to turning the 6710 Mobile Bridge off (removing power), and then starting it up again (reapplying power).

FX s

FX s performs a Ymodem batch protocol download of a file into the specified **s** FLASH segment. Typing FX 1, 2, 3, or 4 (depending upon which FLASH segment you want the file loaded to) automatically copies the file to the specified FLASH segment.

FD

FD displays the FLASH file system directory, including information about the boot file. See "Fd Command" on page 5-6 for more information.

FR

" **NOTE:** *The first executable file in the 6710 Mobile Bridge boot segment must be the 6710 Mobile Bridge boot file.*

FR finds the first executable file in the 6710 Mobile Bridge's boot segment, and tries to run the file.

NPWD

NPWD is for internal use by service personnel only.

SR z

Serial baud rate command SR z sets the baud rate of the 6710 Mobile Bridge. The format is:

```
ap>sr <baud rate>
```

Baud rates are:

2400
4800
9600 *(default)*
14400
19200
28800
38400
57600

Type the desired baud rate as a whole number, with no commas. For example, to enter a baud rate of 19,200, type 19200 — not 19.2 or 19,200.

PWD

PWD opens a password-protected menu that contains file management commands. Some of the commands delete files. Others redefine the 6710 Mobile Bridge's file structure. In either case the commands can cause undesirable results if not properly executed. If in doubt on the proper procedure to use, contact Technical Support for assistance.

To open the password menu, type the following:

ap>pwd

The following displays:

Enter password:

The password is CR52401OWL (must be in uppercase). Following is the password menu.

```
FD          - File System Directory      FPD         - PCMCIA File Directory
FE<s/all>   - Erase Segment(s)           FPE         - Erase PCMCIA Card
FI          - File System Reset          FPX         - Ymodem File to PCMCIA
FS s n      - Define File Segment        PQ          - Power-Up Quiet
FB s        - Set Boot Segment           PN          - Power-Up Normal
FFR f       - Run File                   MI string   - Set Modem Init String
FPC f s     - PCMCIA File to Flash       RMI         - Reset Modem Init String
                                         X           - Exit
```

FD

The FD command displays the segment allocation table and file directory. See "Fd Command" on page 5-6 for more information.

FE <s|all>

FE erases specified or all segments of FLASH memory. See "Fe Command" on page 5-8 for more information.

FI

FI is a destructive command that erases all downloaded files in FLASH memory.

FS s n

FS is a destructive command that redefines the default (factory set) file segments in FLASH memory.

FB s

FB s designates which segment(s) of FLASH memory the boot program is located in. If the 6710 Mobile Bridge boot program is located in a segment other than what is designated, the boot program will not run. See "Fb Command" on page 5-5 for more information.

FFR f

FFR f runs the specified file (f). The file specified with the FFR command must be an executable file.

FPC f s

FPC f s copies a specified file (f) from an SRAM card installed in PC card slot NIC2 to a specified segment (s) of the 6710 Mobile Bridge FLASH.

" **NOTE:** *The FPC f s command copies the first file of the specified file name it finds. Attempting to load multiple files with the same file name will result in unreachable or unreadable files.*

FPD

FPD shows the file system directory of an installed SRAM card. The card must be inserted in PC card slot NIC2 on the 6710 Mobile Bridge. The command will not work on a card inserted in slot NIC1.

The following information about the PC card file system is returned:

- Names of all files on the card.
- Type of file (executable, data, text).
- Size (in bytes) of each file.
- Date of each file.
- Version number of each file, in the format Vxx.xx.

FPE

FPE erases the entire contents of a PC card installed in PC card slot NIC2 on the 6710 Mobile Bridge. *Individual files cannot be deleted*. When you issue the FPE command, each location on a PC card installed in slot NIC2 is overwritten with zeroes.

FPX

FPX performs a Ymodem batch protocol download of a file into an SRAM card installed in slot NIC2 on the 6710 Mobile Bridge. The downloaded file appends to any existing files on the PC card.

PN

PN turns off Power-Up Quiet mode (PQ).

PQ

PQ turns on Power-Up Quiet mode. When you configure the 6710 Mobile Bridge to boot in quiet mode, it does not display ROM power-up messages while it boots. More importantly, it does not allow a single received character to invoke the ROM command monitor and prevent the 6710 Mobile Bridge from booting when a host is connected to it and trying to communicate.

After you issue the PQ command, every ROM power-up message is done in quiet mode. When you turn on quiet mode you cannot access the ROM command monitor by pressing a single keystroke during the boot sequence.

When the ROM command monitor is in quiet mode you must send three or more consecutive exclamation points (!) to the DIAG port during the boot sequence to invoke the command monitor prompt (ap>).

Because the 6710 Mobile Bridge is in quiet mode, no prompts appear to show you when to type the exclamation points. The easiest way to do this is to apply power to the 6710 Mobile Bridge, wait until the WLINK indicator light stops flashing for about 1 second, and then type three or more exclamation points.

MI String

MI String allows a custom modem initialization string to be used in the 6710 Mobile Bridge. When a custom modem initialization string is entered, it overrides the default string issued by the ROM on powerup.

During powerup, the ROM checks the EEPROM for a valid modem initialization string (custom initialization strings *must* start with the letters AT or at). If the ROM finds a valid string in the EEPROM, it uses this string instead of the default string in the ROM. To remove a custom modem initialization string, issue the RMI command.

RMI

RMI removes a custom modem initialization string from the 6710 Mobile Bridge EEPROM (see “MI String”). The next time the 6710 Mobile Bridge is powered on, the default modem initialization string (located in ROM) is loaded.

X

Command X exits the password submenu and returns to the Main Menu.

Exiting the ROM Command Monitor

Exit the command monitor by running the Reboot command (B) or Run Flash Boot File command (FR) on the ROM command monitor's Main Menu.

Software Download Example

The FLASH program for the 6710 Mobile Bridge is called MSD29.BIN. A simple method for upgrading a 6710 Mobile Bridge with new FLASH is to set one up as a TFTP server and then download new FLASH into another 6710 Mobile Bridge (the client). This method is "simple" because you can easily configure a 6710 Mobile Bridge as a TFTP server.

The general procedure is as follows:

1. Upgrade one 6710 Mobile Bridge with a new version of FLASH through the DIAG port.
2. Enable the upgraded 6710 Mobile Bridge as a TFTP server.
3. Use Telnet and TFTP to upgrade another 6710 Mobile Bridge (client).

This procedure may also download new HTML and GIF files, which enable you to configure the 6710 Mobile Bridge through a Web browser.

Upgrading Through DIAG Port

The following pages show an example of how to upgrade a 6710 Mobile Bridge through its DIAG port. This unit will become the TFTP server.

The example assumes that:

- You have established a connection between a PC and the 6710 Mobile Bridge's DIAG port, and have accessed the ROM command monitor.
- The current active version of FLASH is in segment 2. Therefore, segment 2 is the active boot segment.
- An old version of FLASH is in segment 1.
- The new version of FLASH is going into segment 1.

1. When the 6710 Mobile Bridge has entered the ROM command monitor, type the following commands to upgrade it.

Command	Description
ap>**pwd**	Enter the password menu.
Enter the password.	The default password is CR52401OWL (must be uppercase).
passwd>**fe 1**	Erase FLASH segment 1.
passwd>**x**	Exit the password menu.
ap>**fx 1**	Download the new FLASH file into segment 1.
passwd>**fb 1**	Change the boot segment number from 2 (in this example, the active boot segment) to 1.
ap>**fr**	Run the new FLASH boot file by re-starting the 6710 Mobile Bridge.

2. After the 6710 Mobile Bridge reboots and is running in FLASH mode, use the Fd command to display the file directory and verify that the new version of FLASH is in segment 1.
3. Ensure that this 6710 Mobile Bridge has a valid IP address. Change the address if necessary and remember what it is. You need the IP address to configure TFTP clients in other 6710 Mobile Bridges.

Starting the TFTP Server

1. Configure the 6710 Mobile Bridge that you just upgraded to be the TFTP server by typing:

   `File>tftp server start`

2. To check the status of the TFTP server, type:

   `File>tftp server log`

   If the server is active its response is:

   `The TFTP server is running.`

Upgrading TFTP Clients

The following procedure assumes that you are downloading MSD29.BIN into an inactive boot segment and the self-extracting HTML configuration files into an inactive data segment. The HTML files are:

BKGRND.29K
HELP.29K
HLP.29K
INTRMEC.29K
MENU.29K
MSWELCOM.29K
WRITE.29K

1. After you have started the TFTP server, establish a TELNET session with the 6710 Mobile Bridge to be upgraded (the client).
2. Access the client's File Menu.
3. On the client, type the following commands to upgrade it.

Command	Description
File>**fe ib:**	Erase the client's inactive boot segment.
File>**fe id:**	Erase the client's inactive data segment.
File>**tftp get 1.2.3.4 msd29.bin ib:**	Copy executable file MSD29.BIN from the server (IP address is 1.2.3.4) to the client's inactive boot segment.
File>**tftp get 1.2.3.4 <HTML configuration file> id:**	Copy the self-extracting HTML files from the server to the client's inactive data segment. (Repeat this step for each HTML file.)
File>**fb ib: id:**	Make the client's inactive boot and data segments the active segments.
File>**fd**	Display the FLASH file directory to verify that the boot and data segments are the active segments.
File>**exit**	Exit the File Menu and return to the Main Menu.
>**reboot**	Run the new FLASH file by restarting the 6710 Mobile Bridge. Note that when you reboot the 6710 Mobile Bridge client, the Telnet connection is lost. Wait about 30 seconds for the unit to start up again before establishing another Telnet connection.
File>**fd**	Display the FLASH file directory to verify that the correct files are in the active segment.

4. Repeat the above commands for each 6710 Mobile Bridge client that needs a new version of FLASH.
5. After you have upgraded all 6710 Mobile Bridge clients, stop the TFTP server process in the 6710 Mobile Bridge server by typing:

   File>tftp server stop

Section 6
Indicator Lights

This section describes the 6710 Mobile Bridge's indicator lights and how to read them. This section also provides troubleshooting information you can use to isolate a faulty unit.

Overview

The eight indicator lights on the 6710 Mobile Bridge's front panel are the best indicators of how the device is working. By observing the indicator lights you can tell the following:

- Mode in which the 6710 Mobile Bridge is operating (error, network, boot, or command).
- Type of network link the 6710 Mobile Bridge has established when it is in network mode.
- Baud rate of DIAG port when in command mode.
- Possible cause of an error condition when in error mode.

The indicator lights are grouped into four pairs (Figure 6-1).

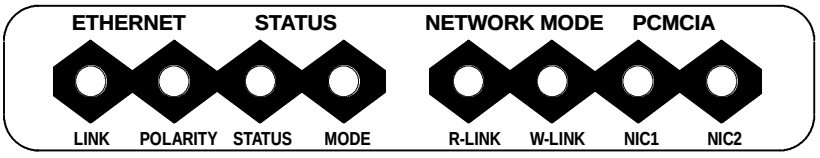


Figure 6-1
Indicator Lights

Each indicator light is labeled according to function. The following chart lists the groups and their indicator lights.

Group	Indicator Lights
ETHERNET	LINK and POLARITY
STATUS	STATUS and MODE
NETWORK MODE	R-LINK and W-LINK
PCMCIA	NIC1 and NIC2

ETHERNET Lights

ETHERNET indicator lights show the status of the 6710 Mobile Bridge's Ethernet connection. Table 6-1 shows what the lights mean when ON and OFF.

Table 6-1
ETHERNET Indicator Lights

Light	Status	Indication
LINK	ON	6710 Mobile Bridge has successfully attached to the Ethernet network.
	OFF	6710 Mobile Bridge has not attached to the Ethernet network.
POLARITY	OFF	TX and RX lines on a 10BASE-T cable are reversed.

STATUS Lights

STATUS indicator lights are labeled STATUS and MODE.

STATUS

The STATUS (left) light indicates the 6710 Mobile Bridge's operating status. When the light is OFF, the 6710 Mobile Bridge is operating normally. When the light is ON, it is in error mode.

" **NOTE:** *In certain cases the following text refers to the indicator lights by number. Lights are numbered from left to right. Light 1 is labeled LINK and Light 8 is labeled NIC2.*

The 6710 Mobile Bridge enters error mode when it detects a malfunction during the power-on self tests or when certain hardware malfunctions occur during normal operation. When the STATUS light is ON to indicate the error mode, indicator lights 4 through 8 (the STATUS MODE light and the NETWORK and PCMCIA light pairs) display a binary error status code between 1 and 31. This code indicates the specific condition causing the error status as shown in Table 6-2. Descriptions of errors are on page 6-4.

Table 6-2
Error Mode Status Codes

Status	Mode	R-LINK	W-LINK	NIC1	NIC2	Error Status
ON	OFF	OFF	OFF	OFF	ON	Hardware error 1.
ON	OFF	OFF	OFF	ON	OFF	Hardware error 2.
ON	OFF	OFF	OFF	ON	ON	Hardware error 3.
ON	OFF	OFF	ON	OFF	OFF	Internal serial loop-back test failed.
ON	OFF	OFF	ON	OFF	ON	Internal MACE AUI loopback test failed.
ON	OFF	OFF	ON	ON	OFF	Internal 10BASE-T loopback test failed.
ON	OFF	OFF	ON	ON	ON	Timer test failed.

" **NOTE:** *Contact the Customer Response Center for help with errors.*

Error Status	Description
Hardware errors 1, 2, and 3	These errors indicate an internal hardware error or malfunction. The errors can occur when you apply power to the 6710 Mobile Bridge. If it encounters a hardware error, it no longer functions.
Internal serial loopback test failed	This failure occurs if the 6710 Mobile Bridge does not successfully complete the power-on self-test. The error indicates a probable hardware malfunction associated with the DIAG port. In most cases, the unit continues to operate normally, but the error condition still exists.
Internal MACE (Media Access Controller for Ethernet) AUI loopback test failed	This failure occurs if the 6710 Mobile Bridge does not successfully complete the power-on self-test. The error indicates a probable hardware malfunction associated with the AUI port. In most cases, the unit continues to operate normally, but the error condition still exists.
Internal 10BASE-T loopback test failed	This failure occurs if the 6710 Mobile Bridge does not successfully complete the power-on self-test. The error indicates a probable hardware malfunction associated with the 10BASE-T port. In most cases, the unit continues to operate normally, but the error condition still exists.
Timer test failed	This failure occurs when the 6710 Mobile Bridge timer circuit malfunctions. If a timer error occurs, the unit no longer functions.

MODE

The right STATUS light is labeled MODE. It indicates the current status of the 6710 Mobile Bridge (Table 6-3).

Table 6-3
MODE Indicator Light

Status	Indication
ON	6710 Mobile Bridge is not functional and is locked up.
BLINK	6710 Mobile Bridge is in network mode, the normal operating condition.
OFF	6710 Mobile Bridge is in command mode. It enters this mode when it detects a key press from an attached PC *before it enters boot mode*, or when it detects an incoming signal from an attached modem.

NETWORK MODE Lights

NETWORK MODE indicator lights show the status of the 6710 Mobile Bridge's network link. The lights are labeled R-LINK (left indicator) and W-LINK (right indicator). They work together to indicate the type of network link the 6710 Mobile Bridge has established. Table 6-4 shows links.

Table 6-4
NETWORK MODE Indicator Lights

R-LINK	W-LINK	Network Link Description
OFF	OFF	No network link established (6710 Mobile Bridge is not connected to the network, or is reconfiguring).
OFF	ON	6710 Mobile Bridge is attached to the network through an Ethernet port.
ON	ON	6710 Mobile Bridge is operating as the super root.

PCMCIA Lights

PCMCIA indicator lights show the status of the two PC card slots, which are labeled NIC1 and NIC2. The left light shows the status of NIC1; the right light shows the status of NIC2. Table 6-5 shows light indications for both ports.

Table 6-5
PCMCIA Indicator Lights

Status	Indication
OFF	A functional or enabled RS-485 PC card is not installed in the slot.
ON	A functional and enabled RS-485 PC card is installed in the slot.

When the 6710 Mobile Bridge reboots in ROM mode, the R-LINK, W-LINK, NIC1, and NIC2 indicator lights show — for about 9 seconds — the DIAG port's baud rate. The lights show the status about 8 seconds after the W-LINK light stops blinking. Table 6-6 shows baud rates.

Table 6-6
DIAG Port Baud Rates, ROM Mode

R-LINK	W-LINK	NIC1	NIC2	Baud Rate
OFF	OFF	OFF	ON	2400
OFF	OFF	ON	OFF	4800
OFF	OFF	ON	ON	9600
OFF	ON	OFF	OFF	14400
OFF	ON	OFF	ON	19200
OFF	ON	ON	OFF	28800
OFF	ON	ON	ON	38400
ON	OFF	OFF	OFF	57600

Power-Up Sequence

When you power on the 6710 Mobile Bridge, it performs a power-up sequence that does the following:

- Tests the indicator lights.
- Tests the functional circuits.
- Determines the operational status.
- Determines the boot sequence.

You can monitor the power-up sequence through the indicator lights. During power-up the lights operate in this order:

1. LINK indicator light turns ON and stays ON.
2. STATUS, MODE, R-LINK, W-LINK, NIC1, and NIC2 lights blink three times to indicate they are operational.
3. After the previous lights stop blinking, W-LINK light blinks three more times.
4. About eight seconds after W-LINK stops blinking, R-WINK, W-LINK, NIC1 and NIC2 lights either turn ON or stay OFF to indicate the DIAG port's baud rate (see page 6-6).
5. MODE light blinks constantly to indicate it is in network mode, the normal operating condition.

After the 6710 Mobile Bridge completes its boot sequence, it enters its normal operating mode. During normal operation the STATUS indicator light is OFF.

Appendix A
Specifications

Product Specifications

Processor:	AMD 29200 RISC.
Memory:	4 MB RAM/2 MB FLASH ROM.
LAN compatibility:	ANSI/IEEE 802.3 (Ethernet communication standard) and DIX Version 2.0.
Interface:	10BASE2 (thinnet), 10BASE5 (AUI or thicknet), and 10BASE-T (twisted pair) through ports on bottom panel. RS-485 through ports on top panel.
Card slots:	Two RS-485 PC card slots.
Mounting options:	Tabletop or wall.

Electrical Specifications

The 6710 Mobile Bridge has one IEC connector for industry-standard three conductor ac input. The bridge's internal power supply automatically detects the voltage level and frequency of the source power. Following are source power specifications.

Voltages:	Autosensing 100 to 240 V ac
Frequency:	50 to 60 Hz
Safety:	UL 1950 CAN/CSA–C22.2 No. 950–M93 IEC 950/EN 60950 (CB REPORT)

The 6710 Mobile Bridge complies with the following standards.

Immunity:	EN (Euro Norm) 50082-1 Generic Immunity Standard
Emissions:	FCC Class B verified, IC ICES–003 Class B, CISPR* 22 (EN 55022) Class B radiated and conducted emissions under EN 50081-1, Generic Emissions Standard

* Comite International Special des Perturpations Radio-electurques/International Special Committee on Radio Interference

Environmental Specifications

Operating temperature (standard):	–22 °F to 122 °F (–30 °C to 50 °C)
Humidity:	Remains operational when exposed to 90 percent humidity, noncondensing conditions.

Physical Characteristics

Approximate size:	3.75 in x 6.88 in x 14.5 in (LWH) (9.5 cm x 17.5 cm x 36 cm)
Approximate weight:	3.75 lbs (1.70 kg)

Appendix B

Port and Cable Pin-Outs

This appendix lists pin-outs for the 6710 Mobile Bridge's DIAG, RS-485, and AUI ports. It also shows pin-outs for the standard null modem (DIAG port) cable and RS-485 LAN cable.

DIAG Port Pin-Outs

The following chart defines the signals present on the pins for the DIAG port. Pin numbering is from left to right and top to bottom. For example, pin 1 is on the top left of the connector, and the last pin is on the bottom right.

Pin Number	Signal Name	Signal Level
1	Not used	
2	TXD	RS-232
3	RXD	RS-232
4	DTR	RS-232
5	GND	
6	DSR	RS-232
7	Not used	
8	Not used	
9	Not used	

RS-485 Port Pin-Outs

The following chart defines the signals present on the pins for the RS-485 port.

Pin Number	Signal Name
2	RS-485+
3	RS-485–
7	Ground
8	Ground

AUI Port Pin-Outs

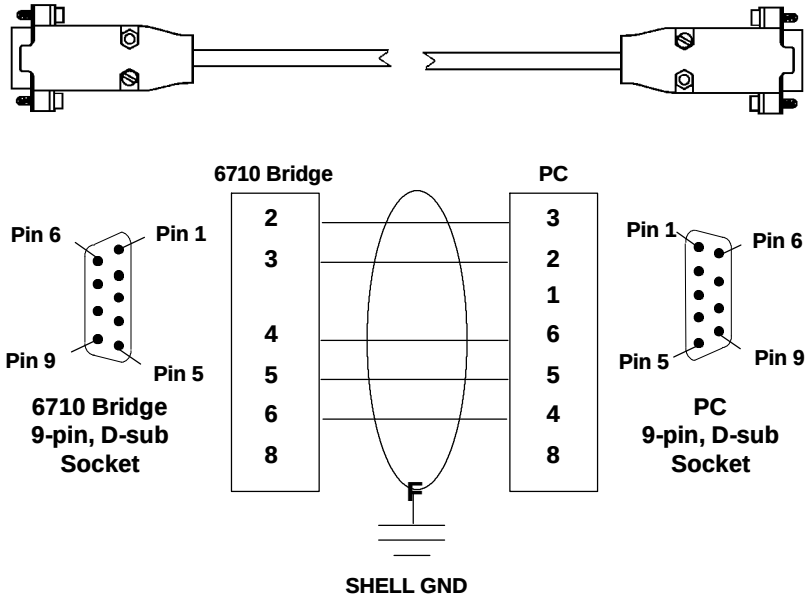
The following chart defines the signals present on the pins for the AUI port. Pin numbering is from left to right and top to bottom. For example, pin 1 is on the top left of the connector, and the last pin is on the bottom right.

Pin Number	Signal Name
1	GND
2	Data
3	Data
4	Not used
5	Data
6	GND
7	Not used
8	Not used
9	Data
10	Data
11	Not used
12	Data
13	12 V dc
14	Not used
15	Not used

DIAG Port Cable

DIAG Port
to 9-Pin Male PC Port
(Standard Null Modem Cable)

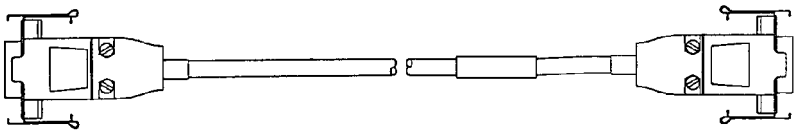
P/N 226-106-001 (6 feet)

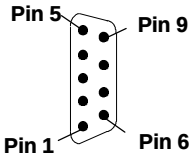


RS-485 LAN Cable

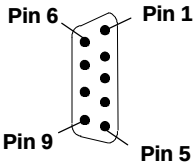
RS-485 Port to:
PEN*KEY^{R} 6100 Series Multidock
or
4960 Multidock

P/N 226-103-0XX
(lengths from 1.5 to 1000 feet)





Connector
D-sub plug, 9-pin



Connector
D-sub plug, 9-pin

FROM	TO	COLOR
P1-Shell	N/C	DRAIN/SHLD
P1-3	P2-3	BLACK
P1-2	P2-2	RED

Appendix C

MIB

MIB Families

Table C-1 lists MIB families for the 6710 Mobile Bridge.

Table C-1
MIB Information

MIB Family	OID	Purpose	MIB Groups
nSystem	1.3.6.1.4.1.469.1000.2.1	Model, device type, software, file system	hw, file, fsinfo, segment, dir, sysErrors, criticalErrors
nInterfaces	1.3.6.1.4.1.469.1000.2.2	I/O ports	nifx, portState, portStats, ptxq, pmsg
nSNMP	1.3.6.1.4.1.469.1000.2.11	SNMP	community, trapTarget
nBridge	1.3.6.1.4.1.469.1000.2.17	Bridging	rt, brg, addr, brgState, bridgeStats
nControl	1.3.6.1.4.1.469.1000.2.105	Device control	powerUp, softwareDownLoad

MIB Directory

The following pages describe the groups the 6710 Mobile Bridge supports. Table C-2 lists the groups and the page on which each group's table summary and definitions appear.

Table C-2
MIB Directory

Group	Meaning	Group Summary	MIB Definition
Product OIDs	NORAND® Products	C-3	C-18
System Information			
hw	Hardware Information	C-4	C-18
fsinfo	File System Information	C-5	C-19
segment	File Segment Information	C-5	C-21
dir	Software Directory Listing	C-6	C-23
criticalErrors	Critical Errors Information	C-6	C-26
Interface Information			
nifx	Norand Extensions to Interfaces Table	C-7	C-28
portState	Port State Information	C-8	C-32
portStats	Port Statistics	C-9	C-37
ptxq	Port Transmit Queue	C-10	C-43
pmsg	Pending Message Services	C-10	C-48
SNMP Version 1 Configuration			
community	Community Table	C-11	C-51
trapTarget	Trap Target Table	C-12	C-54
Bridging Parameters			
rt	Route Table	C-12	C-55
brg	Bridge Table	C-13	C-61
addr	Address Table	C-14	C-63
brgState	Bridge State Information	C-14	C-65
bridgeStats	Bridge Statistics	C-15	C-71

Table C-2 (Continued)
MIB Directory

Group	Meaning	Group Summary	MIB Definition
Control Groups			
powerUp	Power Up Objects	C-17	C-74
softwareDownLoad	Software Download	C-17	C-74

MIB Outline

This outline summarizes the various MIB groups supported for the 6710 Mobile Bridge.

Product OIDs

This group contains an OID for each INTERMEC device. The object name for the 6710 Mobile Bridge is “msd6710.”

Table C-3
products GROUP

Device Products
norand.manage.products.x
(1.3.6.1.4.1.469.1000.1.x)

OID	Object Name	Object Type	Access
1	ap6710	OBJECT ID	Not Applicable (N/A)
2	gw4030	OBJECT ID	(N/A)
3	wnas	OBJECT ID	(N/A)
4	ts6950	OBJECT ID	(N/A)
5	gwap6910	OBJECT ID	(N/A)
6	uap2100	OBJECT ID	(N/A)
7	msd6710	OBJECT ID	(N/A)

System Information

The following groups contain system level objects describing hardware and file system configuration properties. The groups also contain information about critical errors.

" NOTE: *The MIB definition for each group starts on the page given below.*

- hw — Hardware Information (page C-18)
- fsinfo — File System Information (page C-19)
- segment — File Segment Information (page C-21)
- dir — Software Directory Listing (page C-23)
- criticalErrors — Critical Errors Information (page C-26)

Table C-4
hw GROUP

Device Hardware Information
norand.manage.norandNet.nSystem.hw.x
(1.3.6.1.4.1.469.1000.2.1.1.x)

OID	Object Name	Object Type	Access
1	hwPartNo	INTEGER	read
2	hwDescription	DisplayString	read
3	hwRevision	INTEGER	read
4	hwSerialNo	INTEGER	read
5	hwID	INTEGER	read

Table C-5
fsinfo GROUP

Device File System Information
norand.manage.norandNet.nSystem.file.fsinfo.x
(1.3.6.1.4.1.469.1000.2.1.3.1.x)

OID	Object Name	Object Type	Access
1	fsEnabled	INTEGER	read
2	fsMaxSectors	INTEGER	read
3	fsSectorSize	INTEGER	read
4	fsNumSegments	INTEGER	read
5	fsNumFiles	Gauge	read
6	fsBootSegment	INTEGER	read
7	fsDataSegment	INTEGER	read

Table C-6
segment GROUP

Device File Segment Information
norand.manage.norandNet.nSystem.file.segment.x
(1.3.6.1.4.1.469.1000.2.1.3.2.x)

OID	Object Name	Object Type	Access
2.1.1	segID	INTEGER	read
2.1.2	segFirstSector	INTEGER	read
2.1.3	segLastSector	INTEGER	read
2.1.4	segStatus	INTEGER	read
2.1.5	segSize	INTEGER	read
2.1.6	segFree	INTEGER	read

Table C-7
dir GROUP

Device Software Directory Listing
norand.manage.norandNet.nSystem.file.dir.x
(1.3.6.1.4.1.469.1000.2.1.3.3.x)

OID	Object Name	Object Type	Access
2.1.1	dirIndex	INTEGER	read
2.1.2	dirName	DisplayString	read
2.1.3	dirSegment	INTEGER	read
2.1.4	dirType	INTEGER	read
2.1.5	dirSize	INTEGER	read
2.1.6	dirDate	DisplayString	read
2.1.7	dirTime	DisplayString	read
2.1.8	dirVersion	DisplayString	read

Table C-8
criticalErrors GROUP

Device Critical Errors Information
norand.manage.norandNet.nSystem.sysErrors.criticalErrors.x
(1.3.6.1.4.1.469.1000.2.1.4.1.x)

OID	Object Name	Object Type	Access
1	ceEnabled	INTEGER	read
2	ceOverflow	INTEGER	read
3	ceReset	INTEGER	write
4.1.1	ceLogErrorCode	INTEGER	read
4.1.2	ceLogErrorCount	Counter	read

Interface Information

The following groups relate information about interfaces, port state, port statistics, port transmit queue, and pending message services.

" **NOTE:** *The MIB definition for each group starts on the page given below.*

- nifx NORAND Extensions to Interfaces Table (page C-28)
- portState Port State Information (page C-32)
- portStats Port Statistics (page C-37)
- ptxq Port Transmit Queue (page C-43)
- pmsg Pending Message Services (page C-48)

Table C-9
nifx GROUP

NORAND Extensions to MIB-II Interfaces Table
norand.manage.norandNet.nInterfaces.nifx.x
(1.3.6.1.4.1.469.1000.2.2.2.x)

OID	Object Name	Object Type	Access
4.1.1	nifxIndex	INTEGER	read
4.1.2	nifxType	INTEGER	read
4.1.3	nifxInDisabledDiscards	Counter	read
4.1.4	nifxInOverruns	Counter	read
4.1.5	nifxInHWOverruns	Counter	read
4.1.6	nifxInUcastDPkts	Counter	read
4.1.7	nifxInNUcastDPkts	Counter	read
4.1.8	nifxInLenErrors	Counter	read
4.1.9	nifxExcessiveDeferrals	Counter	read
4.1.10	nifxInNetIDDiscards	Counter	read
4.1.11	nifxInFragDiscards	Counter	read
4.1.12	nifxInUFilterDiscards	Counter	read
4.1.13	nifxInNUFilterDiscards	Counter	read
4.1.14	nifxInQFullDiscards	Counter	read

Table C-10
portState GROUP

Device Port State Information
norand.manage.norandNet.nInterfaces.portState.x
(1.3.6.1.4.1.469.1000.2.2.3.x)

OID	Object Name	Object Type	Access
4.1.1	psPort	INTEGER	read
4.1.2	psIfIndex	INTEGER	read
4.1.3	psAddress	PhysAddress	read
4.1.4	psType	INTEGER	read
4.1.5	psState	INTEGER	read
4.1.6	psCost	INTEGER	read
4.1.7	psHelloPeriod	INTEGER	read
4.1.8	psHelloCount	Counter	read
4.1.9	psMacdWindow	INTEGER	read
4.1.10	psMacdQSize	Gauge	read
4.1.11	psMacdTimeouts	Counter	read
4.1.12	psIsPrimary	INTEGER	read
4.1.13	psIsSecondary	INTEGER	read
4.1.14	psIsSecondaryCandidate	INTEGER	read
4.1.15	psSecondaryUniFlooding	INTEGER	read
4.1.16	psSecondaryMultiFlooding	INTEGER	read
4.1.17	psIsRadio	INTEGER	read
4.1.18	psPendEnabled	INTEGER	read

Table C-11
portStats GROUP

Device Port Statistics
norand.manage.norandNet.nInterfaces.portStats.x
(1.3.6.1.4.1.469.1000.2.2.4.x)

OID	Object Name	Object Type	Access
4.1.1	pstcPort	INTEGER	read
4.1.2	pstcInOWLPkts	Counter	read
4.1.3	pstcInUcastOWLDataPkts	Counter	read
4.1.4	pstcInNUcastOWLDataPkts	Counter	read
4.1.5	pstcInOWLErrors	Counter	read
4.1.6	pstcOutOWLPkts	Counter	read
4.1.7	pstcOutUcastOWLDataPkts	Counter	read
4.1.8	pstcOutNUcastOWLDataPkts	Counter	read
4.1.9	pstcOutOWLErrors	Counter	read
4.1.10	pstcParentLinkErrors	Counter	read
4.1.11	pstcAlertLinkErrors	Counter	read
4.1.12	pstcInUcastRelayPkts	Counter	read
4.1.13	pstcInNUcastRelayPkts	Counter	read
4.1.14	pstcOutUcastRelayPkts	Counter	read
4.1.15	pstcOutNUcastRelayPkts	Counter	read
4.1.16	pstcInUcastInbound	Counter	read
4.1.17	pstcInUcastOutbound	Counter	read
4.1.18	pstcInUcastSec	Counter	read
4.1.19	pstcInUcastFlood	Counter	read
4.1.20	pstcUcastDiscards	Counter	read
4.1.21	pstcInNUcastDiscards	Counter	read
4.1.22	pstcInUcastToIFC	Counter	read
4.1.23	pstcInNUcastToIFC	Counter	read
4.1.24	pstcOutDelayDiscards	Counter	read

Table C-12
ptxq GROUP

Device Port Transmit Queue
norand.manage.norandNet.nInterfaces.ptxq.x
(1.3.6.1.4.1.469.1000.2.2.5.x)

OID	Object Name	Object Type	Access
1.1.1	ptxqPort	INTEGER	read
1.1.2	ptxqRegQSize	Gauge	read
1.1.3	ptxqRegQMax	INTEGER	read
1.1.4	ptxqExpQSize	Gauge	read
1.1.5	ptxqExpQMax	INTEGER	read
1.1.6	ptxqQHpCount	Counter	read
1.1.7	ptxqQExpCount	Counter	read
1.1.8	ptxqQRegCount	Counter	read
1.1.9	ptxqQHpDiscards	Counter	read
1.1.10	ptxqQExpDiscards	Counter	read
1.1.11	ptxqQRegDiscards	Counter	read
1.1.12	ptxqMultiQSize	Gauge	read
1.1.13	ptxqMultiQMax	INTEGER	read
1.1.14	ptxqMultiQDiscards	Counter	read

Table C-13
pmsg GROUP

Device Pending Message Service
norand.manage.norandNet.nInterfaces.pmsg.x
(1.3.6.1.4.1.469.1000.2.2.6.x)

OID	Object Name	Object Type	Access
1.1.1	pmsgPort	INTEGER	read
1.1.2	pmsgPendRecCurrent	Gauge	read
1.1.3	pmsgPendRecMax	INTEGER	read
1.1.4	pmsgPendMsgCurrent	Gauge	read
1.1.5	pmsgPendMsgMax	INTEGER	read
1.1.6	pmsgPendMsgTotal	Counter	read
1.1.7	pmsgPendMsgDiscards	Counter	read

Table C-13 (Continued)
pmsg GROUP

Device Pending Message Service
norand.manage.norandNet.nInterfaces.pmsg.x
(1.3.6.1.4.1.469.1000.2.2.6.x)

OID	Object Name	Object Type	Access
1.1.8	pmsgPendRecOverflowErrors	Counter	read
1.1.9	pmsgPendMsgOverflowErrors	Counter	read
1.1.10	pmsgPendAgedRecCount	Counter	read
1.1.11	pmsgPendAgedMsgCount	Counter	read

SNMP Version 1 Configuration Group

This group contains objects that configure the version 1 SNMP agent.

- community Community Table (page C-51)
- trapTarget Trap Target Table (page C-54)

Table C-14
community TABLE

Device SNMP v1 Configurations
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

OID	Object Name	Object Type	Access
2.1.1	communityIndex	INTEGER	read
2.1.2	communityStatus	INTEGER	write
2.1.3	communityName	DisplayString	write
2.1.4	communityPrivileges	INTEGER	write

Table C-15
trapTarget TABLE

Device SNMP v1 Configurations (page C-51)
norand.manage.norandNet.nSNMP.v1Config.x
(1.3.6.1.4.1.469.1000.2.11.1.x)

OID	Object Name	Object Type	Access
3.1.1	trapTargetIndex	INTEGER	read
3.1.2	trapTargetStatus	INTEGER	write
3.1.3	trapTargetName	DisplayString	write
3.1.4	trapTargetIpAddress	IpAddress	write

Bridging Parameters

The following group contains objects relating to the wireless transparent bridging operation.

- rt Route Table (page C-55)
- brg Bridge Table (page C-61)
- addr Address Table (page C-63)
- brgState Bridge State Information (page C-65)
- bridgeStats Bridge Statistics (page C-71)

Table C-16
rt GROUP

Device Route Table
norand.manage.norandNet.nBridge.rt.x
(1.3.6.1.4.1.469.1000.2.17.2.x)

OID	Object Name	Object Type	Access
2.1.1	rtDestination	PhysAddress	read
2.1.2	rtPort	INTEGER	read
2.1.3	rtAge	INTEGER	read
2.1.4	rtNodeId	INTEGER	read
2.1.5	rtAttachId	INTEGER	read
2.1.6	rtAttachTime	TimeTicks	read

Table C-16 (Continued)
rt GROUP

Device Route Table
norand.manage.norandNet.nBridge.rt.x
(1.3.6.1.4.1.469.1000.2.17.2.x)

OID	Object Name	Object Type	Access
2.1.7	rtApEaddr	PhysAddress	read
2.1.8	rtHopAddrLen	INTEGER	read
2.1.9	rtHopAddr16	INTEGER	read
2.1.10	rtHopEaddr	PhysAddress	read
2.1.11	rtIsBound	INTEGER	read
2.1.12	rtIsRemote	INTEGER	read
2.1.13	rtIsChild	INTEGER	read
2.1.14	rtIsAp	INTEGER	read
2.1.15	rtIsDistributed	INTEGER	read
2.1.16	rtIsRemoteLan	INTEGER	read
2.1.17	rtNS	INTEGER	read
2.1.18	rtNR	INTEGER	read

Table C-17
brg GROUP

Device Bridge Table
norand.manage.norandNet.nBridge.brg.x
(1.3.6.1.4.1.469.1000.2.17.3.x)

OID	Object Name	Object Type	Access
2.1.1	brgDestination	PhysAddress	read
2.1.2	brgPort	INTEGER	read
2.1.3	brgAge	INTEGER	read
2.1.4	brgType	INTEGER	read
2.1.5	brgIsPermanent	INTEGER	read
2.1.6	brgTimestamp	TimeTicks	read

Table C-18
addr GROUP

Address Table
norand.manage.norandNet.nBridge.addr.x
(1.3.6.1.4.1.469.1000.2.17.4.x)

OID	Object Name	Object Type	Access
2.1.1	addrDestination	PhysAddress	read
2.1.2	addrAge	INTEGER	read
2.1.3	addrNodeId	INTEGER	read
2.1.4	addrAlias	DisplayString	read
2.1.5	addrDeviceId	INTEGER	read
2.1.6	addrIpAddress	IPAddress	read

Table C-19
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

OID	Object Name	Object Type	Access
3	bsAddress	PhysAddress	read
4	bsLanId	INTEGER	read
5	bsCostToRoot	INTEGER	read
6	bsIsRoot	INTEGER	read
7	bsIsAttached	INTEGER	read
8	bsAttachId	INTEGER	read
9	bsMyRootPriority	INTEGER	read
10	bsRootPort	INTEGER	read
11	bsDesignatedRootAddress	PhysAddress	read
12	bsDesignatedRootPriority	INTEGER	read
13	bsDesignatedRootSequence	INTEGER	read

Table C-19 (Continued)
brgState GROUP

Bridge State Information
norand.manage.norandNet.nBridge.brgState.x
(1.3.6.1.4.1.469.1000.2.17.6.x)

OID	Object Name	Object Type	Access
14	bsParentAddress	PhysAddress	read
15	bsPortCount	INTEGER	read
16	bsNodeId	INTEGER	read
17	bsRootChangedCount	Counter	read
18	bsRootCount	Counter	read
19	bsAttachCount	Counter	read
20	bsDetachReason	INTEGER	read
21	bsNetworkTime	TimeTicks	read
22	bsUniFloodLevel	INTEGER	read
23	bsMultiFloodLevel	INTEGER	read
24	bsIsPrimaryBridge	INTEGER	read
25	bsIsSecondaryBridge	INTEGER	read
26	bsUniFilterExpr	INTEGER	read
27	bsMultiFilterExpr	INTEGER	read

Table C-20
bridgeStats GROUP

Bridge Statistics
norand.manage.norandNet.nBridge.bridgeStats.x
(1.3.6.1.4.1.469.1000.2.17.7.x)

OID	Object Name	Object Type	Access
3	bstcRouteCount	Gauge	read
4	bstcChildCount	Gauge	read
5	bstcChildApCount	Gauge	read

Table C-20 (Continued)
bridgeStats GROUP

Bridge Statistics
norand.manage.norandNet.nBridge.bridgeStats.x
(1.3.6.1.4.1.469.1000.2.17.7.x)

OID	Object Name	Object Type	Access
6	bstcRemoteCount	Gauge	read
7	bstcPrimaryCount	Gauge	read
8	bstcInboundCount	Gauge	read
9	bstcSecondaryCount	Gauge	read
10	bstcRemoteLanCount	Gauge	read
11	bstcRouteGetErrors	Counter	read
12	bstcEntryGetErrors	Counter	read
13	bstcRmtLanGetErrors	Counter	read
14	bstcRouteSeqErrors	Counter	read
15	bstcDeleteSeqErrors	Counter	read
16	bstcEntrySeqErrors	Counter	read
17	bstcInvalidUpdateErrors	Counter	read

Control Groups

The objects in the following groups exert control over the RS-485–Ethernet Bridge. Present functions include rebooting and scheduling software downloads.

- powerUp Power Up Objects (page C-74)
- softwareDownLoad Software Download (page C-74)

Table C-21
powerUp GROUP

Device Power Up Objects
norand.manage.norandNet.nControl.powerUp.x
(1.3.6.1.4.1.469.1000.2.105.1.x)

OID	Object Name	Object Type	Access
1	pwrPowerUpCount	Counter	read
2	pwrNextPowerUpTime	TimeTicks	write

Table C-22
softwareDownLoad GROUP

Device Software Download
norand.manage.norandNet.nControl.softwareDownLoad.x
(1.3.6.1.4.1.469.1000.2.105.2.x)

OID	Object Name	Object Type	Access
1	sdStartTime	TimeTicks	write
2	sdServerIpAddress	IpAddress	write
3	sdScriptFilename	DisplayString	write
4	sdStatus	INTEGER	read
5	sdErrorString	DisplayString	read
6	sdCheckPoint	INTEGER	write
7	sdSetActivePointers	INTEGER	write
8	sdTerminate	INTEGER	write

MIB Definitions

The following pages contain MIB definitions for the 6710 Mobile Bridge.

```
-- Norand Open Wireless LAN MIB - 6710 Mobile Bridge
-- Version 1.15
-- Version Date: 8/23/96
-- This MIB contains objects supported by V1.15 of the 6710 Mobile
-- Bridge.

-- Subject to Change

OWLAP-MIB DEFINITIONS ::= BEGIN

      IMPORTS
          enterprises, IpAddress, Counter, Gauge, TimeTicks
              FROM RFC1155-SMI
          PhysAddress, DisplayString
              FROM RFC1213-MIB
          OBJECT-TYPE
              FROM RFC-1212;

      --  This MIB module uses the extended OBJECT-TYPE macro as defined
      --  in RFC-1212;

norand                      OBJECT IDENTIFIER ::= { enterprises 469 }
    manage                  OBJECT IDENTIFIER ::= { norand 1000 }
        products            OBJECT IDENTIFIER ::= { manage 1 }
            ap6710          OBJECT IDENTIFIER ::= { products 1 }
            gw4030          OBJECT IDENTIFIER ::= { products 2 }
            wnas            OBJECT IDENTIFIER ::= { products 3 }
            ts6950          OBJECT IDENTIFIER ::= { products 4 }
            gwap6910        OBJECT IDENTIFIER ::= { products 5 }
            uap2100         OBJECT IDENTIFIER ::= { products 6 }
            msd6710         OBJECT IDENTIFIER ::= { products 7 }
        norandNet           OBJECT IDENTIFIER ::= { manage 2 }
            nSystem         OBJECT IDENTIFIER ::= { norandNet 1 }

                hw          OBJECT IDENTIFIER ::= { nSystem 1 }

                    -- The Hardware Parameters Group

                    hwPartNo      OBJECT-TYPE
                     SYNTAX INTEGER (0..2147483647)
                     ACCESS read-only
                     STATUS mandatory
                     DESCRIPTION
                     "The Norand part number of the hardware device."
                     ::= { hw 1 }
```

```
    hwDescription OBJECT-TYPE
     SYNTAX DisplayString (SIZE (0..40))
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The description of the hardware device."
     ::= { hw 2 }

    hwRevision    OBJECT-TYPE
     SYNTAX INTEGER (0..2147483647)
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The revision level of the hardware device."
     ::= { hw 3 }

    hwSerialNo    OBJECT-TYPE
     SYNTAX INTEGER (0..2147483647)
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The serial number of the hardware device."
     ::= { hw 4 }

    hwID          OBJECT-TYPE
     SYNTAX INTEGER (0..2147483647)
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The device identifier of the hardware device.
     Values = 3250, 4000, 4020, 4030, 4033, 3240, 1000,
     1100, 1700, 5940, 4650, 100 (ACE process), 200
    (DOSNMS), 300 (Norand Proxy Agent), 6710 (Access
     Point)."
     ::= { hw 5 }

file          OBJECT IDENTIFIER ::= { nSystem 3 }

    fsinfo    OBJECT IDENTIFIER ::= { file 1 }

    -- The File System Information Table
```

```
fsEnabled     OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the file system is enabled."
  ::= { fsinfo 1 }

fsMaxSectors  OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of physical sectors.  A file segment
  consists of one or more adjacent physical sectors."
  ::= { fsinfo 2 }

fsSectorSize  OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The size of a physical sector in bytes."
  ::= { fsinfo 3 }

fsNumSegments OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of logical file segments
 (0-MAX_SECTORS)."
  ::= { fsinfo 4 }

fsNumFiles    OBJECT-TYPE
  SYNTAX Gauge (0..25)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of files (0-25)."
  ::= { fsinfo 5 }
```

```
fsBootSegment OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The index of the current boot segment.  If the
  index is non-zero and the first file in the
  associated segment is executable, then control is
  passed to that file during the power-up sequence."
  ::= { fsinfo 6 }

    fsDataSegment OBJECT-TYPE
      SYNTAX INTEGER
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "The index of the active data segment.  Files
      stored in this segment will be accessible to an
      executing application."
      ::= { fsinfo 7 }

segment   OBJECT IDENTIFIER ::= { file 2 }

-- The File Segment Table

-- Table Definition

segTable        OBJECT-TYPE
 SYNTAX SEQUENCE OF SEGEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "A catalog of memory segments and their
 utilitization."
 ::= { segment 2 }

-- Row Definition

segEntry        OBJECT-TYPE
 SYNTAX SEGEntry
 ACCESS not-accessible
 STATUS mandatory
 INDEX { segID }
 ::= { segTable 1 }
```

```
-- Columnar Object Definitions

SEGEntry ::=
 SEQUENCE {
   segID          INTEGER,
   segFirstSector INTEGER,
   segLastSector  INTEGER,
   segStatus      INTEGER,
   segSize        INTEGER,
   segFree        INTEGER
}

segID           OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The segment ID (1 - (NUM_SEGMENTS+1)).  A non-zero
  number which uniquely identifies a segment."
  ::= { segEntry 1 }

segFirstSector  OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The first physical sector in the segment
 (1 - (MAX_SECTORS + 1))."
  ::= { segEntry 2 }

segLastSector   OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The last physical sector in the segment
  (FIRST_SECTOR - (MAX_SECTORS + 1))."
  ::= { segEntry 3 }
```

```
segStatus       OBJECT-TYPE
  SYNTAX INTEGER  {   valid(1),
                      invalid(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The segment status:
     valid   = 1,
     invalid = 2."
  ::= { segEntry 4 }

segSize         OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The segment size in bytes."
  ::= { segEntry 5 }

segFree         OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of available bytes in the segment which
  are not currently allocated to a file."
  ::= { segEntry 6 }

dir       OBJECT IDENTIFIER ::= { file 3 }

-- The File Directory Table

-- Table Definition

dirTable      OBJECT-TYPE
 SYNTAX SEQUENCE OF DIREntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "The FileSystem Directory."
 ::= { dir 2 }
```

```
-- Row Definition

dirEntry      OBJECT-TYPE
 SYNTAX DIREntry
 ACCESS not-accessible
 STATUS mandatory
 INDEX { dirIndex }
 ::= { dirTable 1 }

-- Columnar Object Definitions

DIREntry ::=
 SEQUENCE {
   dirIndex   INTEGER,
   dirName    DisplayString,
   dirSegment INTEGER,
   dirType    INTEGER,
   dirSize    INTEGER,
   dirDate    DisplayString,
   dirTime    DisplayString,
   dirVersion DisplayString
}

dirIndex      OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Directory Index"
  ::= { dirEntry 1 }

dirName       OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..14))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "File name"
  ::= { dirEntry 2 }
```

```
dirSegment    OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "File segment (1 - (NUM_SEGMENTS + 1)).  The
  segment ID which identifies the segment containing
  the file."
  ::= { dirEntry 3 }

dirType       OBJECT-TYPE
  SYNTAX INTEGER { executable(1),
                       data(2),
                   invalid(3) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "File type:
     executable = 1,
     data       = 2,
     invalid    = 3."
  ::= { dirEntry 4 }

dirSize       OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The file size in bytes."
  ::= { dirEntry 5 }

dirDate       OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..12))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The file date in MM-DD-YYYY display format."
  ::= { dirEntry 6 }
```

```
        dirTime       OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..10))
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "The file time in HH:MM:SS display format."
          ::= { dirEntry 7 }

        dirVersion    OBJECT-TYPE
          SYNTAX DisplayString (SIZE (0..8))
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "The file version in v99.99 display format."
          ::= { dirEntry 8 }

    sysErrors          OBJECT IDENTIFIER ::= { nSystem 4 }

        criticalErrors OBJECT IDENTIFIER ::= { sysErrors 1 }

            ceEnabled       OBJECT-TYPE
              SYNTAX INTEGER { true(1), false(2) }
              ACCESS read-only
              STATUS mandatory
              DESCRIPTION
              "A value of true(1) signifies that the critical
              error log was successfully initialized as part
              of the power-up sequence.  Any errors in that
              initialization process result in a value of
              false(2)."
              ::= { criticalErrors 1 }

            ceOverflow      OBJECT-TYPE
              SYNTAX INTEGER
              ACCESS read-only
              STATUS mandatory
              DESCRIPTION
              "Overflow error code.  If the overflow code is
              non-zero, it indicates that the log has
              overflowed and the overflow code contains the
              last displaced value."
              ::= { criticalErrors 2 }
```

```
ceReset         OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
  "A user can reset the critical error log by
  setting ceReset to true(1).  Valid values are
  true(1) or false(2)."
  ::= { criticalErrors 3 }

ceLogTable      OBJECT-TYPE
  SYNTAX SEQUENCE OF CELogEntry
  ACCESS not-accessible
  STATUS mandatory
  DESCRIPTION
  "Critical Error Log Table"
  ::= { criticalErrors 4 }

ceLogEntry      OBJECT-TYPE
  SYNTAX CELogEntry
  ACCESS not-accessible
  STATUS mandatory
  INDEX { ceLogErrorCode }
  ::= { ceLogTable 1 }

CELogEntry ::=
  SEQUENCE {
    ceLogErrorCode  INTEGER,
    ceLogErrorCount Counter
  }

ceLogErrorCode  OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Critical error code.  A 16-bit value which
  uniquely identifies a system software error.
  The error codes are intended for internal
  Norand use."
  ::= { ceLogEntry 1 }
```

```
            ceLogErrorCount OBJECT-TYPE
              SYNTAX Counter
              ACCESS read-only
              STATUS mandatory
              DESCRIPTION
              "Error count for the associated ceLogErrorCode"
              ::= { ceLogEntry 2 }

nInterfaces       OBJECT IDENTIFIER ::= { norandNet 2 }

    nifx          OBJECT IDENTIFIER ::= { nInterfaces 2 }

 -- The Norand Extended Interfaces Table

    nifxTable               OBJECT-TYPE
     SYNTAX SEQUENCE OF NIFXEntry
     ACCESS not-accessible
     STATUS mandatory
     DESCRIPTION
     "Norand Extended Interface Table"
     ::= { nifx 4 }

    nifxEntry               OBJECT-TYPE
     SYNTAX NIFXEntry
     ACCESS not-accessible
     STATUS mandatory
     INDEX { nifxIndex }
     ::= { nifxTable 1 }
```

```
NIFXEntry ::=
  SEQUENCE {
    nifxIndex                INTEGER,
    nifxType                 INTEGER,
    nifxInDisabledDiscards   Counter,
    nifxInOverruns           Counter,
    nifxInHWOverruns         Counter,
    nifxInUcastDPkts         Counter,
    nifxInNUcastDPkts        Counter,
    nifxInLenErrors          Counter,
    nifxExcessiveDeferrals   Counter,
    nifxInNetIDDiscards      Counter,
    nifxInFragDiscards       Counter,
    nifxInUFilterDiscards    Counter,
    nifxInNUFilterDiscards   Counter,
    nifxInQFullDiscards      Counter
}

nifxIndex               OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “Interface index”
 ::={ nifxEntry 1 }

nifxType                 OBJECT-TYPE
 SYNTAX INTEGER {       ether(4),
                    proxim24(132),
                   falcon902(197),
                        uhf(198),
                      nor24(195)
                }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 “Norand Interface Type”
 ::={ nifxEntry 2 }
```

```
nifxInDisabledDiscards   OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received unicast frames which do not
 require forwarding.  Unicast ethernet frames are
 discarded if ether-to-radio flooding is disabled and the
 destination is unknown; otherwise, unicast frames are
 discarded if the bridge has learned that the destination
 port is the same as the source port."
 ::={ nifxEntry 3 }

nifxInOverruns           OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received frames discarded because the
 frame could not be queued for the MAC-D task."
 ::={ nifxEntry 4 }

nifxInHWOverruns        OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received frames discarded due to hardware
 overruns."
 ::={ nifxEntry 5 }

nifxInUcastDPkts          OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received unicast frames successfully
 delivered to the MAC-D task."
 ::={ nifxEntry 6 }
```

```
nifxInNUcastDPkts         OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received multicast frames successfully
 delivered to the MAC-D task."
 ::={ nifxEntry 7 }

nifxInLenErrors           OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number received frames with length errors."
 ::={ nifxEntry 8 }

nifxExcessiveDeferrals    OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of aborted transmissions due to excessive
 deferrals."
 ::={ nifxEntry 9 }

nifxInNetIDDiscards       OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received frames discarded because the LAN
 ID did not match."
 ::={ nifxEntry 10 }

nifxInFragDiscards        OBJECT-TYPE
 SYNTAX Counter
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The number of received frame fragments discarded
 because a fragmented frame could not be reassembled."
 ::={ nifxEntry 11 }
```

```
    nifxInUFilterDiscards    OBJECT-TYPE
     SYNTAX Counter
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The number of enabled received unicast frames discarded
     due to a unicast filter expression."
     ::={ nifxEntry 12 }
    nifxInNUFilterDiscards   OBJECT-TYPE
     SYNTAX Counter
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The number of enabled received multicast frames
     discarded due to a multicast filter expression."
     ::={ nifxEntry 13 }
    nifxInQFullDiscards      OBJECT-TYPE
     SYNTAX Counter
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The number of received frames discarded because the
     frame could not be queued for the MAC-R task."
     ::={ nifxEntry 14 }
    portState     OBJECT IDENTIFIER ::= { nInterfaces 3 }
-- The Port State Table
    psTable                  OBJECT-TYPE
     SYNTAX SEQUENCE OF PSEntry
     ACCESS not-accessible
     STATUS mandatory
     DESCRIPTION
     "MAC-R port state variables"
     ::= { portState 4 }
    psEntry                  OBJECT-TYPE
     SYNTAX PSEntry
     ACCESS not-accessible
     STATUS mandatory
     INDEX { psPort }
     ::= { psTable 1 }
```

```
PSEntry ::=
  SEQUENCE {
    psPort                    INTEGER,
    psIfIndex                 INTEGER,
    psAddress                 PhysAddress,
    psType                    INTEGER,
    psState                   INTEGER,
    psCost                    INTEGER,
    psHelloPeriod             INTEGER,
    psHelloCount              Counter,
    psMacdWindow              INTEGER,
    psMacdQSize               Gauge,
    psMacdTimeouts            Counter,
    psIsPrimary               INTEGER,
    psIsSecondary             INTEGER,
    psIsSecondaryCandidate    INTEGER,
    psSecondaryUniFlooding    INTEGER,
    psSecondaryMultiFlooding INTEGER,
    psIsRadio                 INTEGER,
    psPendEnabled             INTEGER
}

psPort                    OBJECT-TYPE
  SYNTAX INTEGER (1..4)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "MAC-R port ID (1-4).  A number which uniquely
  identifies the port."
  ::= { psEntry  1 }

psIfIndex                 OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "MAC-D interface index.  The index matches the
  interface index of the associated row in the mib-II
  interface table."
  ::= { psEntry  2 }
```

```
psAddress               OBJECT-TYPE
  SYNTAX PhysAddress
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "802 address of the port."
  ::= { psEntry  3 }

psType                  OBJECT-TYPE
  SYNTAX INTEGER {       ether(4),
                      proxim24(196),
                     falcon902(197),
                          uhf(198),
                        nor24(199)
                 }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Norand port type:
    ether  = 4,
    Proxim = 196,
    Falcon = 197,
    UHF    = 198."
  ::= { psEntry  4 }

psState                 OBJECT-TYPE
  SYNTAX INTEGER { disabled(0),
                       idle(1),
                       open(2),
                    receive(3),
                   transmit(4) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Port state:
   disabled = 0,
   idle     = 1,
   open     = 2,
   receive  = 3,
   transmit = 4."
  ::= { psEntry  5 }
```

```
psCost                  OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Incremental path cost of the port.
   Default values:
   ether  = 20,
   Falcon = 100,
   UHF    = 255."
  ::= { psEntry  6 }

psHelloPeriod           OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Inter-HELLO time (.01 seconds)."
  ::= { psEntry  7 }

psHelloCount            OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "HELLO transmit count"
  ::= { psEntry  8 }

psMacdWindow            OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Maximum number of active MAC-D transmit requests."
  ::= { psEntry  9 }

psMacdQSize             OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current number of active MAC-D transmit requests."
  ::= { psEntry  10 }
```

```
psMacdTimeouts          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "MAC-D transmit timeout errors"
  ::= { psEntry 11 }

psIsPrimary             OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, for primary bridge ports."
  ::= { psEntry 12 }

psIsSecondary           OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, for secondary bridge ports."
  ::= { psEntry 13 }

psIsSecondaryCandidate  OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if secondary bridge port candidates."
  ::= { psEntry 14 }

psSecondaryUniFlooding  OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, for secondary bridge ports which require unicast
  flooding."
  ::= { psEntry 15 }
```

```
psSecondaryMultiFlooding OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, for secondary bridge ports which require
  multicast flooding."
  ::= { psEntry 16 }

psIsRadio               OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, for radio ports."
  ::= { psEntry 17 }

psPendEnabled           OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the port supports pending messages."
  ::= { psEntry 18 }

portStats     OBJECT IDENTIFIER ::= { nInterfaces 4 }

pstcTable                     OBJECT-TYPE
 SYNTAX SEQUENCE OF PSTCEntry
 ACCESS not-accessible
 STATUS mandatory
 DESCRIPTION
 "MAC-R port state variables"
 ::= { portStats 4 }

pstcEntry                     OBJECT-TYPE
 SYNTAX PSTCEntry
 ACCESS not-accessible
 STATUS mandatory
 INDEX { pstcPort }
 ::= { pstcTable 1 }
```

```
PSTCEntry ::=
  SEQUENCE {
    pstcPort                    INTEGER,
    pstcInOWLPkts               Counter,
    pstcInUcastOWLDataPkts      Counter,
    pstcInNUcastOWLDataPkts     Counter,
    pstcInOWLErrors             Counter,
    pstcOutOWLPkts              Counter,
    pstcOutUcastOWLDataPkts     Counter,
    pstcOutNUcastOWLDataPkts    Counter,
    pstcOutOWLErrors            Counter,
    pstcParentLinkErrors        Counter,
    pstcAlertLinkErrors         Counter,
    pstcInUcastRelayPkts        Counter,
    pstcInNUcastRelayPkts       Counter,
    pstcOutUcastRelayPkts       Counter,
    pstcOutNUcastRelayPkts      Counter,
    pstcInUcastInbound          Counter,
    pstcInUcastOutbound         Counter,
    pstcInUcastSec              Counter,
    pstcInUcastFlood            Counter,
    pstcInUcastDiscards         Counter,
    pstcInNUcastDiscards        Counter,
    pstcInUcastToIFC            Counter,
    pstcInNUcastToIFC           Counter,
    pstcOutDelayDiscards        Counter
}

    pstcPort                    OBJECT-TYPE
      SYNTAX INTEGER (1..4)
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "MAC-R port ID (1-4).  A number which uniquely
      identifies the port."
      ::= { pstcEntry 1 }
```

```
pstcInOWLPkts            OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Total received OWL packets"
  ::= { pstcEntry 2 }

pstcInUcastOWLDataPkts   OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received OWL unicast data packets"
  ::= { pstcEntry 3 }

pstcInNUcastOWLDataPkts  OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received multicast OWL data packets"
  ::= { pstcEntry 4 }

pstcInOWLErrors          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received OWL packets with errors"
  ::= { pstcEntry 5 }

pstcOutOWLPkts           OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Total OWL packets sent"
  ::= { pstcEntry 6 }
```

```
pstcOutUcastOWLDataPkts   OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “Total unicast OWL data packets sent”
  ::= { pstcEntry 7 }

pstcOutNUcastOWLDataPkts  OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “Total multicast OWL data packets sent”
  ::= { pstcEntry 8 }

pstcOutOWLErrors          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “OWL packet send errors”
  ::= { pstcEntry 9 }

pstcParentLinkErrors      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “Root port send link errors”
  ::= { pstcEntry  10 }

pstcAlertLinkErrors       OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “Link errors which generated an ALERT.”
  ::= { pstcEntry 11 }
```

```
pstcInUcastRelayPkts      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast relay packets"
  ::= { pstcEntry 12 }

pstcInNUcastRelayPkts     OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received multicast relay packets"
  ::= { pstcEntry 13 }

pstcOutUcastRelayPkts     OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Total unicast relay packets sent"
  ::= { pstcEntry 14 }

pstcOutNUcastRelayPkts    OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Total multicast relay packets sent"
  ::= { pstcEntry 15 }

pstcInUcastInbound        OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast data packets routed inbound or
  relayed onto the distribution LAN."
  ::= { pstcEntry 16 }
```

```
pstcInUcastOutbound       OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast data packets routed outbound"
  ::= { pstcEntry 17 }

pstcInUcastSec            OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast data packets relayed to a
  secondary LAN."
  ::= { pstcEntry 18 }

pstcInUcastFlood          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast data packets with an unknown
  destination."
  ::= { pstcEntry 19 }

pstcInUcastDiscards       OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received unicast data packets discarded"
  ::= { pstcEntry 20 }

pstcInNUcastDiscards      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Received multicast data packets discarded"
  ::= { pstcEntry 21 }
```

```
    pstcInUcastToIFC          OBJECT-TYPE
      SYNTAX Counter
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Received unicast packets passed to the data link
      interface."
      ::= { pstcEntry 22 }

    pstcInNUcastToIFC         OBJECT-TYPE
      SYNTAX Counter
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Received multicast packets passed to the data link
      interface."
      ::= { pstcEntry 23 }

    pstcOutDelayDiscards      OBJECT-TYPE
      SYNTAX Counter
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Send packets discarded due to excessive delay."
      ::= { pstcEntry 24 }

ptxq          OBJECT IDENTIFIER ::= { nInterfaces 5 }

    ptxqTable          OBJECT-TYPE
      SYNTAX SEQUENCE OF PTXQEntry
      ACCESS not-accessible
      STATUS mandatory
      DESCRIPTION
      "The Port Transmit Queue Table"
      ::= { ptxq 1 }

    ptxqEntry          OBJECT-TYPE
      SYNTAX PTXQEntry
      ACCESS not-accessible
      STATUS mandatory
      INDEX { ptxqPort }
      ::= { ptxqTable 1 }
```

```
PTXQEntry ::=
  SEQUENCE {
    ptxqPort           INTEGER,
    ptxqRegQSize       Gauge,
    ptxqRegQMax        INTEGER,
    ptxqExpQSize       Gauge,
    ptxqExpQMax        INTEGER,
    ptxqQHpCount       Counter,
    ptxqQRegCount      Counter,
    ptxqQExpCount      Counter,
    ptxqQHpDiscards    Counter,
    ptxqQRegDiscards   Counter,
    ptxqQExpDiscards   Counter,
    ptxqMultiQSize     Gauge,
    ptxqMultiQMax      INTEGER,
    ptxqMultiQDiscards Counter
  }

ptxqPort           OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "MAC-R port ID (1-4).  A number which uniquely
  identifies the port."
  ::= { ptxqEntry 1 }

ptxqRegQSize       OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current regular queue size (0-REG_Q_MAX).  The
  number of regular priority packets which are
  currently queued for transmission on the port."
  ::= { ptxqEntry 2 }
```

```
ptxqRegQMax        OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The maximum number of regular priority packets
  which can be queued for transmission on the port."
  ::= { ptxqEntry 3 }

ptxqExpQSize       OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current expedited queue size (0..EXP_Q_MAX).  The
  number of expedited packets which are currently
  queued for transmission on the port."
  ::= { ptxqEntry 4 }

ptxqExpQMax        OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The maximum number of expedited packets which can
  be queued for transmission on the port."
  ::= { ptxqEntry 5 }

ptxqQHpCount       OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of attempts to queue a high priority
  packet for transmission."
  ::= { ptxqEntry 6 }
```

```
ptxqQExpCount      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “The number of attempts to queue an expedited
  priority packet for transmission.”
  ::= { ptxqEntry 7 }

ptxqQRegCount      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “The number of attempts to queue a regular priority
  packet for transmission.”
  ::= { ptxqEntry 8 }

ptxqQHpDiscards    OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “The number of failed attempts to queue a high
  priority packet.”
  ::= { ptxqEntry 9 }

ptxqQExpDiscards   OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  “The number of failed attempts to queue an
  expedited priority packet.”
  ::= { ptxqEntry 10 }
```

```
ptxqQRegDiscards   OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of failed attempts to queue a regular
  priority packet."
  ::= { ptxqEntry 11 }

ptxqMultiQSize     OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current multicast queue size.  The number of
  multicast packets which are queued for transmission
  on the (radio) port.  Multicast packets are
  transmitted after HELLO packets on OWL radio
  ports."
  ::= { ptxqEntry 12 }

ptxqMultiQMax      OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The maximum number of multicast packets which will
  be queued for transmission on the (radio) port."
  ::= { ptxqEntry 13 }

ptxqMultiQDiscards OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of failed attempts to queue a multicast
  packet."
  ::= { ptxqEntry 14 }
```

```
pmsg            OBJECT IDENTIFIER ::= { nInterfaces 6 }

    pmsgTable                   OBJECT-TYPE
      SYNTAX SEQUENCE OF PmsgEntry
      ACCESS not-accessible
      STATUS mandatory
      DESCRIPTION
      "Pending Message Table"
      ::= { pmsg 1 }

    pmsgEntry                   OBJECT-TYPE
      SYNTAX PmsgEntry
      ACCESS not-accessible
      STATUS mandatory
      INDEX { pmsgPort }
      ::= { pmsgTable 1 }

    PmsgEntry ::=
      SEQUENCE {
        pmsgPort                    INTEGER,
        pmsgPendRecCurrent          Gauge,
        pmsgPendRecMax              INTEGER,
        pmsgPendMsgCurrent          Gauge,
        pmsgPendMsgMax              INTEGER,
        pmsgPendMsgTotal            Counter,
        pmsgPendMsgDiscards         Counter,
        pmsgPendRecOverflowErrors   Counter,
        pmsgPendMsgOverflowErrors   Counter,
        pmsgPendAgedRecCount        Counter,
        pmsgPendAgedMsgCount        Counter
      }

    pmsgPort                    OBJECT-TYPE
      SYNTAX INTEGER
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "MAC-R port ID (1-4).  A number which uniquely
      identifies the port."
      ::= { pmsgEntry 1 }
```

```
pmsgPendRecCurrent        OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current terminal record count"
  ::= { pmsgEntry 2 }

pmsgPendRecMax            OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Maximum terminal record count"
  ::= { pmsgEntry 3 }

pmsgPendMsgCurrent        OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Current pending message count"
  ::= { pmsgEntry 4 }

pmsgPendMsgMax            OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Maximum pending message count"
  ::= { pmsgEntry 5 }

pmsgPendMsgTotal          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Total pending message count"
  ::= { pmsgEntry 6 }
```

```
pmsgPendMsgDiscards       OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of pending messages in-queue which were
  discarded before they could be delivered because
  the terminal's queue was full."
  ::= { pmsgEntry 7 }

pmsgPendRecOverflowErrors OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of times that a terminal requested
  pending message services when no pending message
  records were available."
  ::= { pmsgEntry 8 }

pmsgPendMsgOverflowErrors OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of times when the maximum number of
  stored messages, per platform, was exceeded."
  ::= { pmsgEntry 9 }

pmsgPendAgedRecCount      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "The number of terminal records discarded due to
  maximum age (12 minutes)."
  ::= { pmsgEntry 10 }
```

```
        pmsgPendAgedMsgCount      OBJECT-TYPE
          SYNTAX Counter
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "The number of pending messages which were
          discarded due to maximum age.
         (default = 5 seconds)."
          ::= { pmsgEntry 11 }

nSNMP            OBJECT IDENTIFIER ::= { norandNet 11 }

    v1Config      OBJECT IDENTIFIER ::= {nSNMP 1}

       --Norand Community table defines the accepted
       --community strings and their access privileges

       -- The Community Table

        communityTable         OBJECT-TYPE
          SYNTAX  SEQUENCE OF CommunityEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION
          "The community table defines communities and their
          access privileges.  Norand's implementation of the
          community table has some special considerations:

          1) GETs and SETs to the community table can only be
             accomplished using the SUPER-USER community
             string which Norand has defined;
          2) This SUPER-USER community string, or password,
             is defined in the first row of the community
             table.  The communityName contained in the first
             row of the community table is always the
             SUPER-USER community string.  This community
             string (communityName) may be modified.
```

```
      3) All rows of the community table are modifiable
        (SET) when using the SUPER-USER community string.
         However, for the first row of the community
         table, only the communityName object is
         modifiable.  This ensures that the SUPER-USER
         will always have maximum access to the MIB data.
         All other rows in the community Table are
         accessible as defined in the MIB definition.
      4) The SUPER-USER and other default community
         string values can be found in Norand's User's
         Guide."
      ::= { v1Config 2 }

-- Row Definition

 communityEntry         OBJECT-TYPE
   SYNTAX  CommunityEntry
   ACCESS  not-accessible
   STATUS  mandatory
   DESCRIPTION
   "Each entry relates to a specific community and
   associates to its access privileges."
   INDEX   { communityIndex }
   ::= {communityTable 1 }

-- Columnar Object Definition

 CommunityEntry ::=
   SEQUENCE {
     communityIndex         INTEGER,
     communityStatus        INTEGER,
     communityName          DisplayString,
     communityPrivileges    INTEGER
     communityViewTblIndex  INTEGER
 }

 -- Leaf Definition
```

```
communityIndex     OBJECT-TYPE
  SYNTAX  INTEGER
  ACCESS  read-only
  STATUS  mandatory
  DESCRIPTION
  "Identifies the community row."
  ::= { communityEntry 1 }

communityStatus        OBJECT-TYPE
  SYNTAX  INTEGER {  enabled(1),
                     disabled(2),
                      deleted(3) }
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
  "Status of a community record.  Alterations to the
  table may only be performed by a manager using the
  SUPER-USER community name.  Status types:
         Enabled = Community record active
         Disabled = Community record not active
         Deleted = Disables and nulls objects in
                   record."
  ::= { communityEntry 2 }

communityName      OBJECT-TYPE
  SYNTAX  DisplayString (SIZE (0..15))
  ACCESS  read-write
  STATUS  mandatory
  DESCRIPTION
  "The authoritative name for the community.  Unless
  the Norand SUPER-USER community name is employed, a
  GET from this column yields an access violation."
  ::= { communityEntry 3 }
```

```
    communityPrivileges OBJECT-TYPE
      SYNTAX  INTEGER {    get-only(1),
                           set-and-get(3) }
      ACCESS  read-write
      STATUS  mandatory
      DESCRIPTION
      "SET and GET privileges of community."
      ::= { communityEntry 4 }
  -- Norand trap table defines all trap target IP
  -- addresses
    -- Table Definition
    trapTargetTable     OBJECT-TYPE
      SYNTAX  SEQUENCE OF TrapTargetEntry
      ACCESS  not-accessible
      STATUS  mandatory
      DESCRIPTION
      "The trap target table specifies the IP address of
      SNMPv1 managers that expect trap notifications."
      ::= { v1Config 3 }
    -- Row Definition
    trapTargetEntry     OBJECT-TYPE
      SYNTAX  TrapTargetEntry
      ACCESS  not-accessible
      STATUS  mandatory
      DESCRIPTION
      "Each entry relates to a specific named manager at
      a given IP address and belonging to given
      community."
      INDEX   { trapTargetIndex }
      ::= {trapTargetTable 1 }
    -- Columnar Object Definition
    TrapTargetEntry ::=
      SEQUENCE {
        trapTargetIndex     INTEGER,
        trapTargetStatus    INTEGER,
        trapTargetName      DisplayString,
        trapTargetIpAddress IpAddress
    }
    -- Leaf Definition
```

```
        trapTargetIndex     OBJECT-TYPE
          SYNTAX INTEGER
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "Identifies the trapTarget row"
          ::= { trapTargetEntry 1 }

        trapTargetStatus    OBJECT-TYPE
          SYNTAX  INTEGER {  enabled(1),
                             disabled(2),
                              deleted(3) }
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION
          "Status of a trapTarget record."
          ::= { trapTargetEntry 2 }

        trapTargetName      OBJECT-TYPE
          SYNTAX  DisplayString (SIZE (0..16))
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION
          "The authoritative name for the trapTarget."
          ::= { trapTargetEntry 3 }

        trapTargetIpAddress OBJECT-TYPE
          SYNTAX  IpAddress
          ACCESS  read-write
          STATUS  mandatory
          DESCRIPTION
          "IP Address of manager (which is assumed to be
          bound to and listening on port 162)."
          ::= { trapTargetEntry 4 }

nBridge           OBJECT IDENTIFIER ::= { norandNet 17 }

    rt            OBJECT IDENTIFIER ::= { nBridge 2 }

        -- The RT Table

        -- Table Definition
```

```
rtTable            OBJECT-TYPE
SYNTAX SEQUENCE OF RTEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION
“Each entry in this table provides routing
information for child nodes which are reachable via a
route.”
::= { rt 2 }

-- Row Definition

rtEntry            OBJECT-TYPE
SYNTAX RTEntry
ACCESS not-accessible
STATUS mandatory
INDEX { rtDestination }
::= { rtTable 1 }

-- Columnar Object Definition

RTEntry ::=
SEQUENCE {
    rtDestination   PhysAddress,
    rtPort          INTEGER,
    rtAge           INTEGER,
    rtNodeId        INTEGER,
    rtAttachId      INTEGER,
    rtAttachTime    TimeTicks,
    rtApEaddr       PhysAddress,
    rtHopAddrLen    INTEGER,
    rtHopAddr16     INTEGER,
    rtHopEaddr      PhysAddress,
    rtIsBound       INTEGER,
    rtIsRemote      INTEGER,
    rtIsChild       INTEGER,
    rtIsAp          INTEGER,
    rtIsDistributed INTEGER,
    rtIsRemoteLan   INTEGER,
    rtNS            INTEGER,
    rtNR            INTEGER
}
```

```
-- Leaf Definition

rtDestination      OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The 802 address of the destination."
::= { rtEntry 1 }

rtPort             OBJECT-TYPE
SYNTAX INTEGER (1..4)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The MAC-R port ID (1-4).  A number which uniquely
identifies the port."
::= { rtEntry 2 }

rtAge              OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time (in minutes) since the route was updated."
::= { rtEntry 3 }

rtNodeId           OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"16-bit node ID of the destination.  A 16-bit
identifier which uniquely identifies an OWL node in
an OWL LAN."
::= { rtEntry 4 }
```

```
rtAttachId          OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Attach sequence number.  The sequence number is
copied from an OWL ATTACH request PDU.  The sequence
number is not valid for 'remote' nodes."
::= { rtEntry 5 }

rtAttachTime        OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Last attach time (.01 seconds)."
::= { rtEntry 6 }

rtApEaddr           OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
"802 address of AP which is first hop on the path to
the destination."
::= { rtEntry 7 }

rtHopAddrLen        OBJECT-TYPE
SYNTAX INTEGER { twoByte(2),
                 sixByte(6) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
"MAC-D address length (2 or 6).  A MAC-D entity may
use either 16-bit locally assigned addresses or
48-bit 802 addresses."
::= { rtEntry 8 }
```

```
rtHopAddr16         OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
“16-bit MAC-D address ( if rtHopAddrLen is
twoByte(2) ).”
::= { rtEntry 9 }

rtHopEaddr          OBJECT-TYPE
SYNTAX PhysAddress
ACCESS read-only
STATUS mandatory
DESCRIPTION
“48-bit MAC-D address ( if rtHopAddrLen is
sixByte(6) ).”
::= { rtEntry 10 }

rtIsBound           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the destination is fully attached and the
path can be used to forward data.”
::= { rtEntry 11 }

rtIsRemote          OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the destination is a non-OWL node.”
::= { rtEntry 12 }

rtIsChild           OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the destination is a child node.”
::= { rtEntry 13 }
```

```
rtIsAp              OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the destination is an AP.”
::= { rtEntry 14 }

rtIsDistributed     OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the path is through a distributed AP (root
node only).”
::= { rtEntry 15 }

rtIsRemoteLan       OBJECT-TYPE
SYNTAX INTEGER { true(1), false(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION
“True if the destination is a remote LAN.”
::= { rtEntry 16 }

rtNS                OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION
“MAC-R send sequence number for terminal nodes.  The
16-bit (0-65535) sequence number of last OWL data
request PDU sent to the destination.”
::= { rtEntry 17 }
```

```
    rtNR                OBJECT-TYPE
    SYNTAX INTEGER (0..65535)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "MAC-R receive sequence number for terminal nodes.
    The 16-bit (0-65535) sequence number of the last OWL
    data request PDU received from the destination."
    ::= { rtEntry 18 }

brg           OBJECT IDENTIFIER ::= { nBridge 3 }

    -- The BRG Table

    -- Table Definition

    brgTable            OBJECT-TYPE
     SYNTAX SEQUENCE OF BRGEntry
     ACCESS not-accessible
     STATUS mandatory
     DESCRIPTION
     "Each entry in this table provides bridge
     information for child nodes which are reachable via
     a bridge."
     ::= { brg 2 }

    -- Row Definition

    brgEntry            OBJECT-TYPE
     SYNTAX BRGEntry
     ACCESS not-accessible
     STATUS mandatory
     INDEX { brgDestination }
     ::= { brgTable 1 }

    -- Columnar Object Definition

    BRGEntry ::=
     SEQUENCE {
       brgDestination PhysAddress,
       brgPort        INTEGER,
       brgAge         INTEGER,
       brgType        INTEGER,
       brgIsPermanent INTEGER,
       brgTimestamp   TimeTicks
    }
```

```
-- Leaf Definition

brgDestination     OBJECT-TYPE
 SYNTAX PhysAddress
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The 802 address of the destination."
 ::= { brgEntry 1 }

brgPort            OBJECT-TYPE
 SYNTAX INTEGER (1..4)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "MAC-R port ID (1-4).  A number which uniquely
 identifies the port."
 ::= { brgEntry 2 }

brgAge             OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Time (in minutes) since the entry was updated."
 ::= { brgEntry 3 }

brgType            OBJECT-TYPE
 SYNTAX INTEGER {   primary(1),
                  secondary(2),
                    inbound(4) }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Entry Type:
   primary   = 1,
   secondary = 2,
   inbound   = 4."
 ::= { brgEntry 4 }
```

```
    brgIsPermanent      OBJECT-TYPE
     SYNTAX INTEGER { true(1), false(2) }
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "TRUE, if the entry is permanent."
     ::= { brgEntry 5 }

    brgTimestamp        OBJECT-TYPE
     SYNTAX TimeTicks
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "The time when the primary or inbound entry was
     added or the time when the secondary entry was added
     or reattached."
     ::= { brgEntry 6 }

addr          OBJECT IDENTIFIER ::= { nBridge 4 }

    -- The Addr Table

    -- Table Definition

    addrTable           OBJECT-TYPE
     SYNTAX SEQUENCE OF AddrEntry
     ACCESS not-accessible
     STATUS mandatory
     DESCRIPTION
     "Each entry in this table provides address
     information for all OWL nodes in the network.  This
     table exists only at the root node."
     ::= { addr 2 }

    -- Row Definition

    addrEntry           OBJECT-TYPE
     SYNTAX AddrEntry
     ACCESS not-accessible
     STATUS mandatory
     INDEX { addrDestination }
     ::= { addrTable 1 }
```

```
-- Columnar Object Definition

AddrEntry ::=
 SEQUENCE {
   addrDestination PhysAddress,
   addrAge         INTEGER,
   addrNodeId      INTEGER,
   addrAlias       DisplayString,
   addrDeviceId    INTEGER,
   addrIpAddress   IpAddress
}

-- Leaf Definition

addrDestination    OBJECT-TYPE
 SYNTAX PhysAddress
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The 802 address of the registered port."
 ::= { addrEntry 1 }

addrAge            OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "The time (in minutes) since the entry was updated."
 ::= { addrEntry 2 }

addrNodeId         OBJECT-TYPE
 SYNTAX INTEGER (0..65535)
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "16-bit (0-65535) node/port ID"
 ::= { addrEntry 3 }

addrAlias          OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..16))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "An alias for the 802 address."
 ::= { addrEntry 4 }
```

```
    addrDeviceId       OBJECT-TYPE
     SYNTAX INTEGER (0..65535)
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "Device ID (0-65535).  An OWL node can, optionally,
     set a device ID in a registration request PDU."
     ::= { addrEntry 5 }

    addrIpAddress      OBJECT-TYPE
     SYNTAX IpAddress
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
     "32-bit IP address for IP nodes (e.g. APs)."
     ::= { addrEntry 6 }

brgState             OBJECT IDENTIFIER ::= { nBridge 6 }

    -- The Bridge State Group

    bsAddress                    OBJECT-TYPE
      SYNTAX PhysAddress
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "802 address of the AP."
      ::= { brgState 3 }

    bsLanId                      OBJECT-TYPE
      SYNTAX INTEGER (0..254)
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "OWL LAN ID (0-254)."
      ::= { brgState 4 }

    bsCostToRoot                 OBJECT-TYPE
      SYNTAX INTEGER
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Path cost to the root."
      ::= { brgState 5 }
```

```
bsIsRoot                    OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the AP is the root."
  ::= { brgState 6 }

bsIsAttached                OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the AP is attached."
  ::= { brgState 7 }

bsAttachId                  OBJECT-TYPE
  SYNTAX INTEGER (0..65535)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "16-bit (0-65535) ATTACH sequence number.  This
  number is incremented each time the AP sends an
  ATTACH request."
  ::= { brgState 8 }

bsMyRootPriority            OBJECT-TYPE
  SYNTAX INTEGER (0..7)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Root priority of the AP (0-7).  An AP with a root
  priority of 0 cannot become the root node.  The AP
  with the highest priority will become the root in
  an OWL LAN."
  ::= { brgState 9 }
```

```
bsRootPort                    OBJECT-TYPE
  SYNTAX INTEGER (1..4)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "MAC-R root port number.  The port number (1-4) of
  the port used to communicate with the parent node."
  ::= { brgState 10 }

bsDesignatedRootAddress       OBJECT-TYPE
  SYNTAX PhysAddress
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "802 address of the current root."
  ::= { brgState 11 }

bsDesignatedRootPriority      OBJECT-TYPE
  SYNTAX INTEGER (1..7)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Root priority of the current root (1-7)."
  ::= { brgState 12 }

bsDesignatedRootSequence      OBJECT-TYPE
  SYNTAX INTEGER (0..255)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Root sequence of the current root (0-255).  The
  sequence number identifies a single instance of the
  root."
  ::= { brgState 13 }

bsParentAddress               OBJECT-TYPE
  SYNTAX PhysAddress
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "802 address of the parent AP."
  ::= { brgState 14 }
```

```
bsPortCount                 OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Number of MAC-R ports"
  ::= { brgState 15 }

bsNodeId                    OBJECT-TYPE
  SYNTAX INTEGER (0..65535)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "16-bit node ID (0-65535).  The node ID uniquely
  identifies the node in an OWL LAN."
  ::= { brgState 16 }

bsRootChangedCount          OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Number of times that the root has changed."
  ::= { brgState 17 }

bsRootCount                 OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Number of times that the AP became the root."
  ::= { brgState 18 }

bsAttachCount               OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Number of times that the AP has changed from an
  unattached state to an attached state."
  ::= { brgState 19 }
```

```
bsDetachReason                OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Last detach reason code.  The code indicates the
  reason that the AP became unattached for the last
  occurrence.
  0 - Initial Value,
  1 - A new root node was detected,
  2 - The network inactivity timer expired,
  4 - A better path to the root was detected,
  5 - The node's parent became unattached,
  7 - The node was in a detach list in a HELLO PDU,
  8 - The node was functioning as the root and
      relinquished the root status,
  9 - The maximum number of attached retries was
      exceeded without receiving an ATTACH response
      PDU,
  900-90F - A MAC-D link error occurred while sending
            a PDU to the parent node."
  ::= { brgState 20 }

bsNetworkTime                 OBJECT-TYPE
  SYNTAX TimeTicks
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Distributed network time (.01 second)."
  ::= { brgState 21 }

bsUniFloodLevel               OBJECT-TYPE
  SYNTAX INTEGER (1..2)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Flooding level for unicast frames (1-2)."
  ::= { brgState 22 }
```

```
bsMultiFloodLevel            OBJECT-TYPE
  SYNTAX INTEGER (0..3)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Flooding level for multicast frames (0-3)."
  ::= { brgState 23 }

bsIsPrimaryBridge            OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the AP bridges to the distribution LAN."
  ::= { brgState 24 }

bsIsSecondaryBridge          OBJECT-TYPE
  SYNTAX INTEGER { true(1), false(2) }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "TRUE, if the AP is the designated bridge for a
  secondary LAN."
  ::= { brgState 25 }

bsUniFilterExpr              OBJECT-TYPE
  SYNTAX INTEGER (0..255)
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Unicast ethernet filter ID (0-255).  If the ID is
  non-zero, it points to a user-defined expression
  which filters unicast frames on the ethernet port."
  ::= { brgState 26 }
```

```
    bsMultiFilterExpr           OBJECT-TYPE
      SYNTAX INTEGER (0..255)
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Multicast ethernet filter ID (0-255).  If the ID
      is non-zero, it points to a user-defined expression
      which filters multicast frames on the ethernet
      port."
      ::= { brgState 27 }

bridgeStats   OBJECT IDENTIFIER ::= { nBridge 7 }

    bstcRouteCount          OBJECT-TYPE
      SYNTAX Gauge
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Total route table entries"
      ::= { bridgeStats 3 }

    bstcChildCount          OBJECT-TYPE
      SYNTAX Gauge
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Number of attached children"
      ::= { bridgeStats 4 }

    bstcChildApCount        OBJECT-TYPE
      SYNTAX Gauge
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Number of attached AP children"
      ::= { bridgeStats 5 }

    bstcRemoteCount         OBJECT-TYPE
      SYNTAX Gauge
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
      "Non-OWL bridge table entries"
      ::= { bridgeStats 6 }
```

```
bstcPrimaryCount        OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Distribution LAN bridge table entries"
  ::= { bridgeStats 7 }

bstcInboundCount        OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Inbound bridge table entries"
  ::= { bridgeStats 8 }

bstcSecondaryCount      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Secondary LAN bridge table entries"
  ::= { bridgeStats 9 }

bstcRemoteLanCount      OBJECT-TYPE
  SYNTAX Gauge
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Route entries for remote LANs"
  ::= { bridgeStats 10 }

bstcRouteGetErrors      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Route table overflow errors"
  ::= { bridgeStats 11 }
```

```
bstcEntryGetErrors      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Bridge table overflow errors"
  ::= { bridgeStats 12 }

bstcRmtLanGetErrors     OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Remote LAN overflow errors"
  ::= { bridgeStats 13 }

bstcRouteSeqErrors      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Out-of-sequence route update errors"
  ::= { bridgeStats 14 }

bstcDeleteSeqErrors     OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Out-of-sequence route delete errors"
  ::= { bridgeStats 15 }

bstcEntrySeqErrors      OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
  "Out-of-sequence bridge entry update errors"
  ::= { bridgeStats 16 }
```

```
        bstcInvalidUpdateErrors OBJECT-TYPE
          SYNTAX Counter
          ACCESS read-only
          STATUS mandatory
          DESCRIPTION
          "Invalid route update errors"
          ::= { bridgeStats 17 }

nControl          OBJECT IDENTIFIER ::= { norandNet 105 }

    powerUp       OBJECT IDENTIFIER ::= { nControl 1 }

        pwrPowerUpCount    OBJECT-TYPE
         SYNTAX Counter
         ACCESS read-only
         STATUS mandatory
         DESCRIPTION
         "Power-up count"
         ::= { powerUp 1 }

        pwrNextPowerUpTime OBJECT-TYPE
         SYNTAX TimeTicks
         ACCESS read-write
         STATUS mandatory
         DESCRIPTION
         "Next power-up time (Used to reboot the device)."
         ::= { powerUp 2 }

    softwareDownLoad   OBJECT IDENTIFIER ::= { nControl 2 }

        sdServerIpAddress   OBJECT-TYPE
         SYNTAX IpAddress
         ACCESS read-write
         STATUS mandatory
         DESCRIPTION
         "TFTP server IP address"
         ::= { softwareDownLoad 2 }

        sdScriptFilename    OBJECT-TYPE
         SYNTAX DisplayString (SIZE (0..80))
         ACCESS read-write
         STATUS mandatory
         DESCRIPTION
         "Download script filename (May include path)."
         ::= { softwareDownLoad 3 }
```

```
sdStartTime         OBJECT-TYPE
 SYNTAX TimeTicks
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "The amount of time to delay before beginning the
 software download."
 ::= { softwareDownLoad 1 }

sdStatus            OBJECT-TYPE
 SYNTAX INTEGER {
                      sdPending(1),
                    sdSTStopped(2),
                    sdInProcess(3),
                   sdTerminated(4),
                      sdSuccess(5),
                        sdError(6),
                        pwrNPUT(7),
                      tftpError(8)
                }
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Status of the current software download."
 ::= { softwareDownLoad 4 }

sdErrorString       OBJECT-TYPE
 SYNTAX DisplayString (SIZE (0..40))
 ACCESS read-only
 STATUS mandatory
 DESCRIPTION
 "Description of sdStatus field"
 ::= { softwareDownLoad 5 }

sdCheckPoint        OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "An application variable intended to contain a
 number relating the progress of the current software
 download."
 ::= { softwareDownLoad 6 }
```

```
sdSetActivePointers OBJECT-TYPE
 SYNTAX INTEGER {
                    none(1),
                    boot(2),
                    data(3),
                    both(4)
                  }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "If the device reboots due to the expiration of the
 pwrNextPwrUpTime timer, this value specifies which
 active pointers will be toggled prior to rebooting."
 ::= { softwareDownLoad 7 }

sdTerminate         OBJECT-TYPE
 SYNTAX INTEGER {
                   true(1),
                   false(2)
                  }
 ACCESS read-write
 STATUS mandatory
 DESCRIPTION
 "Terminate the current software download."
 ::= { softwareDownLoad 8 }

END
```

INDEX

SYMBOLS

NUMBERS

A

B

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

W

X

Y