Bull

AIX Versions 3.2 and 4
Asynchronous Communications Guide

AIX

ORDER REFERENCE
86 A2 26AQ 05

Bull

AIX Versions 3.2 and 4
Asynchronous Communications Guide

AIX

Software

December 1998

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 26AQ 05

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

Trademarks and Acknowledgements

Contents

Communications Statements

The following statement applies to products covered in this book. The statement for other products intended for use with these products appear in their accompanying documentation.

Federal Communications Commission (FCC) Statement

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. Neither the provider nor the manufacturer are responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications to this equipment. Unauthorized changes or modifications could void the user's authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference,and (2) this device must accept any interference received, including interference that may cause undesired operation.

European Union (EU) Statement

This product is in conformity with the protection requirements of EU Council Directive 89/336/EEC on the approximation of the laws of the Member States relating to electromagnetic compatibility. The manufacturer cannot accept responsibility for any failure to satisfy the protection requirements resulting from a non–recommended modification of the product, including the fitting of option cards supplied by third parties. Consult with your dealer or sales representative for details on your specific hardware.

This product has been tested and found to comply with the limits for Class A Information Technology Equipment according to CISPR 22/European Standard EN 55022. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

Attention:

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

International Electrotechnical Commission (IEC) Statement

This product has been designed and built to comply with IEC Standard 950.

United Kingdom Telecommunications Safety Requirements

This equipment is manufactured to the International Safety Standard EN60950 and as such is approved in the UK under the General Approval Number NS/G/1234/J/100003 for indirect connection to the public telecommunication network.

The network adapter interfaces housed within this equipment are approved separately, each one having its own independent approval number. These interface adapters, supplied by the manufacturer, do not use or contain excessive voltages. An excessive voltage is one which exceeds 70.7 V peak ac or 120 V dc. They interface with this equipment using Safe Extra Low Voltages only. In order to maintain the separate (independent) approval of the manufacturer's adapters, it is essential that other optional cards, not supplied by the manufacturer, do not use main voltages or any other excessive voltages. Seek advice from a competent engineer before installing other adapters not supplied by the manufacturer.

Avis de conformité aux normes du ministère des Communications du Canada

Cet appareil numérique de la classe A respecte toutes les exigences du Réglement sur le matériel brouilleur du Canada.

Canadian Department of Communications Compliance Statement

This Class A digital apparatus meets the requirements of the Canadian Interference–Causing Equipment Regulations.

VCCI Statement

電波障害自主規制　届出装置の記述

この装置は，第一種情報装置（商工業地域において使用されるべき情報装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会（VCCI）基準に適合しております。 従って，住宅地域またはその隣接した地域で使用すると，ラジオ，テレビジョン受信機等に受信障害を与えることがあります。 取扱説明書に従って正しい取り扱いをしてください。

The following is a summary of the VCCI Japanese statement in the box above.

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may arise. When such trouble occurs, the user may be required to take corrective actions.

Electromagnetic Interference (EMI) Statement – Taiwan

この装置は，第二種情報装置（住宅地域又はその隣接した地域において使用されるべき情報装置）で住宅地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）基準に適合しております。

しかし，本装置をラジオ，テレビジョン受信機に近接してご使用になると，受信障害の原因となることがあります。

取扱説明書に従って正しい取り扱いをして下さい。

The following is a summary of the EMI Taiwan statement above.

Warning: This is a Class A product. In a domestic environment this product may cause radio interference in which case the user will be required to take adequate measures.

Radio Protection for Germany

Dieses Gerät ist berechtigt in Übereinstimmung mit Dem deutschen EMVG vom 9.Nov.92 das EG–Konformitätszeichen zu führen.

Der Aussteller der Konformitätserklärung ist die IBM Germany.

Dieses Gerät erfüllt die Bedingungen der EN 55022 Klasse A. Für diese von Geräten gilt folgende Bestimmung nach dem EMVG:

Geräte dürfen an Orten, für die sie nicht ausreichend entstört sind, nur mit besonderer Genehmigung des Bundesministers für Post und Telekommunikation oder des Bundesamtes für Post und Telekommunikation betrieben werden. Die Genehmigung wird erteilt, wenn keine elektromagnetischen Störungen zu erwarten sind.

(Auszug aus dem EMVG vom 9.Nov.92, Para.3, Abs.4)

Hinweis

Dieses Genehmigungsverfahren ist von der Deutschen Bundespost noch nicht veröffentlicht worden.

Safety Notices

Note: For a translation of the safety notices, refer to the *System Unit Safety Information*, Order Number 86 X1 11WD.

A *danger* notice indicates the presence of a hazard that has the potential of causing death or serious personal injury. See *Danger* notices, on page 8-13.

A *caution* notice indicates the presence of a hazard that has the potential of causing moderate or minor personal injury. See *caution* notice, on page 8-13.

About This Book

This book is for AIX system administrators who want to program the AIX tty subsystem interface and to configure asynchronous adapters for use with the system.

Overview and planning information for asynchronous adapter subsystems is presented first, followed by information on installing, configuring, and troubleshooting the adapters.

Highlighting

The following highlighting conventions are used in this book:

Bold	Identifies commands, keywords, files, directories, and other items whose names are predefined by the system.
Italics	Identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

ISO 9000

ISO 9000 registered quality systems were used in the development and manufacturing of this product.

Related Publications

AIX and Related Products Documentation Overview, order number 86 A2 71WE.

Common Diagnostics and Service Guide, order number 86 A1 52WE.

AIX Version 3.2 System Management Guide: Communications and Network, order number 86 A2 50WG.

AIX Version 3.2 System Management Guide: Operating System and Devices, order number 86 A2 49WG.

AIX Versions 4.1 and 4.2 System Management Guide: Communications and Network, order number 86 A2 54AP.

AIX Versions 4.1 and 4.2 System Management Guide: Operating System and Devices, order number 86 A2 53AP.

AIX 4.3 System Management Guide: Communications and Networks, order number 86 A2 31JX.

AIX 4.3 System Management Guide: Operating System and Devices, order number 86 A2 99HX.

7318 Network Terminal Accelerator Guide and Reference, order number 86 A2 72WG.

AIX Versions 4.1 and 4.2 Problem Solving Guide and Reference, order number 86 A2 56AP.

AIX Version 4.3 Problem Solving Guide and Reference, order number 86 A2 32JX.

Adapters Information for Micro Channel Architecture Systems, order number 86 A1 76AT.

Ordering Publications

You can order publications from your sales representative or from your point of sale.

To order additional copies of this book, use order number 86 A2 26AQ.

Use *AIX and Related Products Documentation Overview* for information on related publications and how to obtain them.

Chapter 1. Asynchronous Communication Planning

This section provides an overview of the available asynchronous adapters as well as an introduction on how to use each of the adapters. Both the hardware and software aspects of using serial ports are also covered.

Asynchronous Adapter Overview

AIX is a multiuser operating system allowing many users to access system resources and applications. Each user must be connected through a terminal session. The connection can be local or remote via a serial port or network connection using Transmission Control Protocol/Internet Protocol (TCP/IP).

Each system unit has at least two standard serial ports available (some systems have three serial ports). These ports can support asynchronous communication and device attachment.

Two standard ports are sufficient for users requiring an additional ASCII terminal or a modem. Many users require additional asynchronous ports.

Asynchronous communications products offer the advantages of low cost, multi–user, medium– to high– performance terminal and device communications. Asynchronous ports allow attachment of asynchronous peripheral devices that meet EIA–232, EIA–422, or EIA–423 standards such as:

- Asynchronous modems
- Bar code scanners
- Graphic and character printers
- Keyboard and display terminals
- Personal computers
- Plotters and printers
- Point–of–sale terminals
- Sensors and control devices
- Text scanners
- Time clocks

Evaluating Asynchronous Communications Options

Expanded asynchronous capability can be added to the system unit with direct–attached adapters using Micro Channel (MC), Industry Standard Architecture (ISA), or Peripheral Component Interconnect (PCI) buses; distributed subsystems; or by local area network (LAN) attached communications servers. Several factors will influence the type of asynchronous connectivity you choose. The following table summarizes these products.

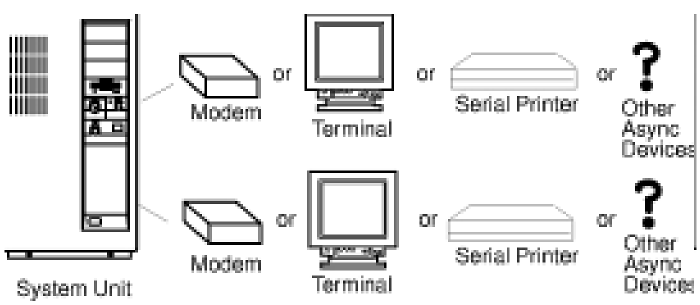
Asynchronous Attachment	Bus Type	Feature Code or Machine Type (Model)	Maximum Data Rate per Port (KBits/sec)	Salient Features
Standard serial port	System planar	n/a	Selectable based on baud rate generator clock speed of universal asynchronous receiver and transmitter (UART).	Standard feature
8–Port EIA 232	Micro Channel	2930	76.8	Pervasive standard
8–Port EIA 422A	Micro Channel	2940	76.8	Greater distance
8–port MIL–STD 188	Micro Channel	2950	Selectable based on baud rate generator clock speed of UART.	MIL–STD 188–114 for unbalanced voltage digital interface
8–port EIA 232	ISA	2931	115.2	Greater efficiency
8–port EIA 232	ISA	2932	115.2	Greater efficiency
8–port EIA 422	PCI	2943	230	Greater efficiency
16–Port EIA 232	Micro Channel	2955	76.8	Local connection focus
16–Port EIA 422A	Micro Channel	2957	76.8	Greater distance
232 RAN		8130	57.6	Remote capability
Enhanced 232 RAN		8137	230	Remote capability
16–Port RAN EIA 422		8138	230	Remote capability
128–Port Controller	Micro Channel	8128	230	Efficiency, higher device counts
	ISA	2933		
	PCI	2944		
Serial Communications Network Server (P10)	Network Attachment	7318–P10	115.2	Greater distance, multihost, two parallel ports

Asynchronous Attachment	Bus Type	Feature Code or Machine Type (Model)	Maximum Data Rate per Port (KBits/sec)	Salient Features
Serial Communications Network Server (S20)	Network Attachment	7318–S20	115.2	Greater distance, multihost, two parallel ports, Serial Line Interface Protocol (SLIP), complressed SLIP (CSLIP), Point–to–Point Protocol (PPP), Kerberos security, TN3270
Network Terminal Accelerator	Micro Channel	2402 2403	n/a	Offload Telnet and rlogin processing, 256 or 2048 sessions
Note: 1. Rack Mount RAN FC is 8136				

The first feature in this table represents the standard–attached serial ports that are standard with every system unit. The next five features are the direct–attached adapters. The 128–port asynchronous subsystem includes the remote asynchronous nodes (RANs) that attach to it. The next two features, models of the machine type 7318, are the LAN–attached asynchronous products that are communications servers attached to Ethernet LANs for connectivity to the system units. The last feature is the Network Terminal Accelerator that offloads protocol processing for LAN–attached terminal sessions.

Standard–Attached Asynchronous Ports

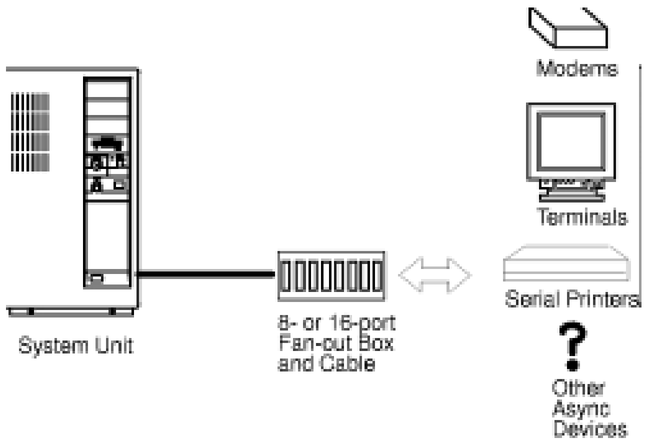
Most system unit models have two integrated (standard) EIA 232 asynchronous serial ports as shown in the following figure. The Model M20/M2A features a single integrated asynchronous serial connector that can be converted to support two serial devices using an optional fan–out cable. EIA 232 asynchronous serial devices can be attached directly to the standard serial ports using standard serial cables with 25–pin D–shell connectors. Some multiprocessor systems have a third serial port used for communication to the remote service center.



Native-Attached Asynchronous

Direct–Attached Asynchronous Ports

Each of the direct–attached adapters requires a bus slot and can only be used in systems that support the required bus type. The Micro Channel 8– and 16–port adapters are relatively simple, meaning that very little processing of asynchronous traffic is offloaded from the system processor. The 128–port, ISA 8–port adapters, and PCI 8–port adapters are intelligent adapters that provide significant offload of the main system processor. As shown in the following figure, the 8–port and 16–port adapters use fan–out cables to connect to devices and require no additional power supply.

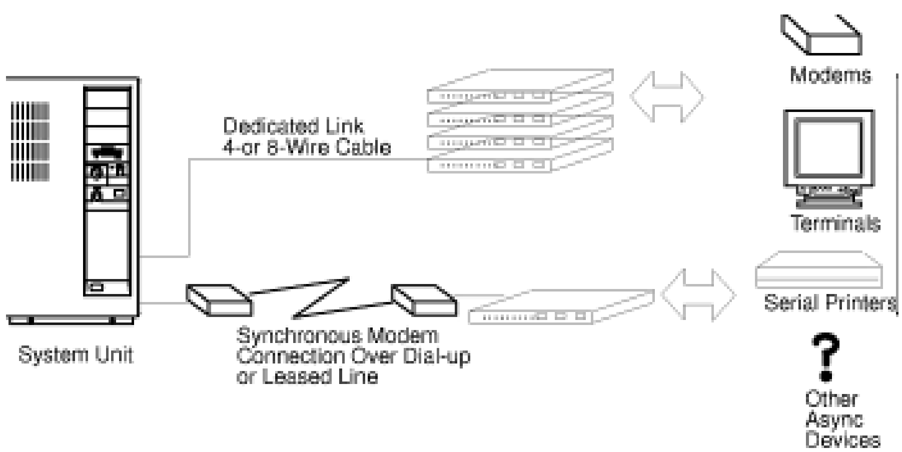


Direct-Attached Asynchronous

EIA 232 is the most common communication standard, but the EIA 422A (used when a longer cable distance is needed) is also supported by each direct–attached adapter for the Micro Channel. The EIA 422A implementation does not include device status detection capability. A Micro Channel 8–port adapter complying with the military standard (MIL–STD) 188–114 is also available. All adapters are supported on AIX Version 3.2 and AIX Version 4 with the exception of the ISA and PCI bus adapters, which are not supported on Version 3.2.

Node–Attached Asynchronous Ports

The 128–port adapter, available for the Micro Channel, ISA, or PCI bus, allows attachments of one to eight remote asynchronous nodes (RANs). Each RAN has 16 asynchronous ports for connection to devices and are separately powered units. Up to four RANs can be daisy–chain connected from each of two connections on the 128–port adapter card (see figure). In AIX Versions 4.2 and later, RANs may support 16 EIA 232 devices or 16 EIA 422 devices. The 128–port controller is an intelligent adapter that increases the number of asynchronous sessions possible at a given CPU utilization level.



Node-Attached Asynchronous

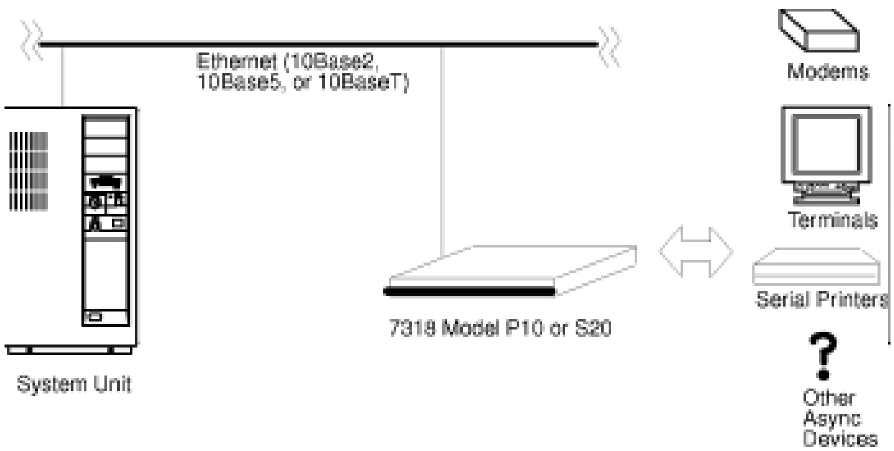
The following are additional characteristics of the 128–port feature:

- Supported on AIX Versions 3.2 and 4.
- RANs may be located up to 300 meters from the system processor using 8–wire shielded cabling while maintaining full performance ratings.
- Distance may be extended to 1200 meters by reducing the data rate between the RANs and the system processor.

- RANs may be remotely located from the system processor using a synchronous EIA 232 and EIA 422 modems. Each four–RAN daisy chain is allowed only one modem pair at some point in the chain.
- System performance is enhanced by offloading tty character processing from the system processor.

LAN–Attached Asynchronous Ports

LAN connectivity removes the physical connection limitations inherent with direct–attached or node–attached asynchronous ports. LAN connectivity also allows flexibility of connection to multiple system units on the network. There are two models of the 7318 LAN–attached communications server, Models P10 and S20. The two models have the same physical connectivity and appearance. The Model S20 supports a superset of Model P10 functionality. Each model offers asynchronous ports, two parallel ports, and the advantage of connection through Ethernet. This means that they can connect through the integrated Ethernet port available on some system units or through an Ethernet adapter card, thus requiring no additional Micro Channel slots dedicated to asynchronous I/O. The following figure depicts LAN–attached asynchronous ports.



LAN-Attached Asynchronous

The Model S20 also allows connection of its asynchronous ports to systems on the LAN. On each port, a user may have up to seven concurrent sessions to one or more systems and hot–key between sessions and systems.

All 16 asynchronous ports of the 7318 are RS–232D and EIA 423 compatible. Eight of the ports can support EIA 422A devices.

The 7318 has the capability to detect a problem and switch to its standby Ethernet port. In the event of a power outage, you can hot–key to a second system on the network and continue to be productive. This makes the 7318 an ideal choice for asynchronous connectivity in high availability cluster multi–processing (HACMP) installations.

The following are additional characteristics of the Models P10 and S20:

- Supported on AIX Version 3.2 and Version 4.
- Location can be anywhere in the your Ethernet network (10Base2, 10Base5, or 10BaseT).
- High data rates allow fast SLIP connections over inexpensive high–speed modems. (For the Model P10, AIX SLIP support is used.)
- Two PC–compatible parallel ports are provided on each 7318 for parallel device attachments, such as printers.
- AIX print queue facilities can be used.

- Daisy chaining allows connection of four 7318s with one Ethernet transceiver or 10BaseT hub port.
- Convenient rack or wall mounting kits available.
- Low–cost unshielded twisted–pair cabling may be used for both Ethernet and asynchronous port connections.
- DEC terminals with EIA 423 MMJ connectors and Apple MacIntosh EIA 423 port connections are supported (converters are available as 7318 features).

7318 Model P10

The Model P10 communication servers use Internet Packet Exchange/Sequence Packet Exchange (IPX/SPX) to transport asynchronous traffic to and from the system. The P10 also has the advantage of making its ports appear as AIX tty ports, thus simplifying configuration and system management tasks. P10 ports are configured using AIX's System Management Interface (SMIT) screens.

The P10 enhances system performance by offloading tty character processing from the system CPU. It provides more than 40% better system processor efficiency when compared to standard Telnet terminal servers for data entry intensive applications. This allows more users to utilize a system or provide faster application processing for a given number of users on a system.

7318 Model S20

The Model S20 ports may operate as P10–style ports, with the attributes described in the Model P10 section. As S20–style ports, they have the following capabilities.

TCP/IP Network Connectivity

The Model S20 ports provides connectivity to systems with Telnet protocols, rlogin protocols, and TCP/IP. With the S20, you can Telnet and rlogin to any similarly capable system on the network.

For large and mid–sized system connectivity (for example, S/390 and AS/400), the S20 supports 3270 data streams. This enables asynchronous terminals to emulate 3270 terminals. The S20 supports TN3270 on four user–configurable terminal types simultaneously, including support for color terminals. For AIX applications, the S20 offloads the TN3270 protocol processing from the system to improve system performance.

Serial Line Internet Protocol (SLIP) communications protocol is offered by AIX. In addition, the S20 has built–in support for SLIP as well as compressed SLIP (CSLIP) and point–to–point protocol (PPP). Each of these industry–standard protocols supports TCP/IP connections between AIX and other systems over asynchronous lines, enabling remote network access through modem connections.

Printer and Modem Capability

The S20 supports reverse Telnet (rtelnet) protocol for modem and printer applications. Unlike terminals, printers cannot initiate Telnet or rlogin sessions to a system. Printers require the system to initiate the connection. rtelent is used by S20 to accept incoming connections from systems for printing or modem applications. With the modem or printer device pooling feature, systems can share modems or printers and use the first available device. Another useful feature is transparent printing. This allows the S20, using rtelnet, to support auxiliary printer ports, provided on most asynchronous terminals.

For printing applications requiring printer spooling, the S20 port can be configured as a P10–style port and use the AIX print queue capability.

Management

The S20 supports standard Simple Network Management Protocol (SNMP) Management Information Base (MIB) II services with on–board SNMP agent. This enables the network administrator to query the S20 from any SNMP management node on the network.

The S20 comes with a default configuration file that enables terminals to be connected quickly and easily. This ASCII file may be customized to add additional sessions, configure SLIP options, or invoke applications, such as rlogin, to a specified host with a specific user name.

Security

The S20 has a full range of security features. For minimal security protection, the S20 supports local password protection. This means that the user must first login to the S20 and enter the correct password before gaining access to the network. For remote network access, password authentication protocol (PAP) is supported with PPP.

The next higher level of security provided by the S20 is Kerberos V.5 password authentication. Kerberos V.5 is a protocol used to prevent passwords from being sent in clear text over the network. This is an important security feature for those networks that are not physically secure. Kerberos V.5 authentication utilities can be provided by the distributed computing environment (DCE). This allows the S20 to participate in the centralized security services provided by DCE.

The highest level of security provided by the S20 is full data encryption. With encrypted rlogin, all data sent and received by the S20 port is Data Encryption Standard (DES) encrypted. This protects data from unauthorized access on the network.

For more information on the 7318 Model S20, see the *7318 Network Terminal Accelerator Guide and Reference*.

Network Terminal Accelerator

Note: The Network Terminal Accelerator is not supported in AIX Version 4.3 or later.

The Network Terminal Accelerator adapter is an Ethernet adapter that improves system performance in applications where user terminals connect to the host through a TCP/IP network. For host systems supporting large numbers of users, the Network Terminal Accelerator can significantly reduce host CPU usage by offloading the processing of Telnet and rlogin protocols from the host to the adapter. There are two models of the adapter which support up to 256 or 2048 maximum concurrent Telnet and rlogin user sessions.

Using the Network Terminal Accelerator in conjunction with the 7318 Model S20 terminal server maximizes system performance in addition to providing the flexibility of LAN–attached terminal connections.

Product Selection Considerations

This section will help you determine what asynchronous product you should choose for a particular situation.

Product Selection Aid

The following questions will help you choose an AIX offering for your installation.

Expandability How many asynchronous ports are needed?

How may ports will be needed in the future?

Topology Will devices be in other buildings or remote locations?

Does an Ethernet LAN exist?

Where will system/network administration be done?

Is there an HACMP cluster?

What type of cabling is required or already there?

Is connection to multiple hosts a requirement?

Performance Is your AIX application CPU–intensive?

What types of devices will be attached?

What is the relative asynchronous bandwidth demand for the aggregate sum of the devices?

Relative Device Bandwidth Demand		
Low Demand	**Moderate Demand**	**High Demand**
ASCII terminals, Point–of–sale terminals, Asynchronous modems	Printers, Low–speed FAX/modems, Bar code scanners	Serial X–terminals, High–speed FAX/modems, High–speed printers, File transfer applications

Micro Channel and ISA Bus Slots

How many slots are available for asynchronous adapters?

Device Interface Requirement

What asynchronous interface is required, for example, EIA 232, EIA 422A, EIA 423, MIL–STD 188?

Do the devices or applications require the full EIA 232 interface?

Is a remote parallel port required for printing?

Security Is dedicated calling a requirement?

Is a device name to a physical device mapping required?

Is system assurance kernel (SAK) required?

The following table shows the detailed product characteristics.

Asynchronous Attachment Product Characteristics									
	Native Serial Ports	8–port			16–port	128–port with RAN			7318 Terminal Server
		MC	ISA	PCI		MC	ISA	PCI	
Number of asynchronous ports per adapter	n/a	8	8	8	16	128	128	128	n/a
Maximum number of adapters	n/a	8	7	8	8	7	7	8	n/a
Maximum number of asynchronous ports	2 or 3	64	56	64	128	896	896	1024	1584 (P10) no limit (S20)
Number of asynchronous ports per 7318 or RAN	n/a	n/a	n/a	n/a	n/a	16	16	16	16
Number of parallel ports	n/a	0	0	0	0	0	0	0	2
Maximum number of 7318s or RANs	n/a	n/a	n/a	n/a	n/a	56	56	64	99 (P10) no limit (S20)
Maximum speed (KBits/sec)	Selectable based on baud rate generator clock speed of UART.	76.8	115.2	230	76.8	230	230	230	115.2
Attachment method	standard	direct	direct	direct	direct	node	node	node	LAN

Asynchronous Attachment Product Characteristics									
	Native Serial Ports	8–port			16–port	128–port with RAN			7318 Terminal Server
		MC	ISA	PCI		MC	ISA	PCI	
Asynchronous electrical interfaces supported	EIA 232	EIA 232 EIA 422 A[4] MIL–STD[4] 188–114[4]	EIA 232 EIA 422 A	EIA 232 EIA 422 A	EIA 232 EIA 422 A	EIA 232 EIA 422	EIA 232 EIA 422	EIA 232 EIA 422	EIA 232 EIA 423
Standard connector	DB25M/ MODU	DB25M	DB25M	DB25M	DB25M	RJ–45[2]	RJ–45[2]	RJ–45	RJ–45
DB25 cable options	n/a	n/a	n/a	n/a	n/a	RJ–45 – DB25	RJ–45 – DB25	RJ–45 – DB25	Term./printer: RJ–45–DB25 Modem: RJ–45–DB9 RJ–45–MiniDIN RJ–45–MMJ
Rack mount option	n/a	n/a	n/a	n/a	n/a	yes	yes	yes	yes
Power supply	n/a	n/a	n/a	n/a	n/a	external	external	external	internal
Signals supported (EIA 232)	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTSRTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS[3] – DTR – DCD –	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR DCD RI	TxD RxD RTS CTS DTR DSR[1] DCD RI[1]

Notes:

1. DSR and RI are available in buddy mode in the 7318.
2. Socket accepts 8p RJ–45, 6p RJ–11, or 4p RJ–11 plugs with a reduction in signals supported.
3. RTS is tied high (+12V) in the fanout connector box of the 16–port interface cable EIA 232 (FC 2996).
4. Micro Channel Only.

Adapter Applications

Each product offering is characterized by a representative scenario for its strengths. The following are recommendations for each:

8–Port Micro Channel

- A Micro Channel bus slot available for asynchronous I/O.
- Fewer than eight ports with little or no expansion.
- All local terminals located within 61 meters (200 feet) from the system.
- Remote terminals needed (support through OEM multiplexer/modem).
- Device bandwidth demand low to moderate (up to 76.8 Kbps).

8–Port ISA Bus EIA 232 or EIA 232/EIA 422

- ISA slot available.
- Fewer than eight ports required with little or no expansion.
- Requires all EIA 232, all EIA 422, or a mix of EIA 232 and EIA 422 ports.
- Offload character interrupt and terminal I/O processing from the main CPU.
- Asynchronous speeds to 115.2 Kbps.
- Maximum performance for high speed (28.8 Kbps) modems with data compression.

8–Port PCI Bus EIA 232/EIA 422

- PCI slot available.
- Fewer than eight ports required with little or no expansion.
- Requires all EIA 232, all EIA 422, or a mix of EIA 232 and EIA 422 ports.
- Offload character interrupt and terminal I/O processing from the main CPU.
- Asynchronous speeds to 230 Kbps.
- Maximum performance for high speed (33.6 Kbps) modems with data compression.

16–Port Micro Channel

- A Micro Channel bus slot available for asynchronous I/O.
- Eight ports now, fewer than 16 ports with little or no expansion.
- All local terminals located within 61 meters (200 feet) from the system.
- Remote terminals needed (support through OEM multiplexer/modem).
- Devices do not need all EIA 232 signals.
- Device bandwidth demand low to moderate (up to 38.4 Kbps for asynchronous devices).

128–Port Adapter (Micro Channel, ISA, and PCI)

- A Micro Channel, ISA, or PCI bus slot available for asynchronous I/O.
- Sixteen ports now with expansion of up to 128 ports without additional slots.
- Most distant terminal located about 300 meters (1000 feet) from the system at maximum data rate for MC and ISA Adapters, 90 meters (300 feet) from the system at maximum data rate for the PCI adapter.
- Terminals planned: nearby or on premises, distant on premises, and remote.
- Need high asynchronous throughput with low processor demand.
- Need terminal attached printer capability.
- Need to connect to remote premises through fiber–optic or synchronous modems.

7318 Model P10

- Ethernet adapter available or integrated in the system planar.
- Limited number of available slots.

- Need flexibility, investment protection, and distance capability of Ethernet.
- Need high throughput asynchronous ports.
- Need to maximize system performance for Ethernet–attached asynchronous devices while sharing the standard system.
- Need AIX multisystem connectivity.
- Mission–critical applications require standby Ethernet availability.
- Desire network transparency for asynchronous and printer applications.
- Need greater system–to–device distances than the limits of EIA 232.
- Need full EIA 232 modem controls for eight or fewer ports.
- Need high–speed parallel printer ports.
- Standalone diagnostic capability without disruption of the system.
- Need static mapping of tty device names to LAN–attached asynchronous devices.
- Need support for attachment of EIA 232, EIA 423, and EIA 422 devices.
- Asynchronous speeds to 115.2 Kbps.

7318 Model S20

- Desire flexibility, investment protection, and distance capability of Ethernet.
- Need option to maximize system performance with the Network Terminal Accelerator (NTA) adapter.
- Need to communicate with other systems.
- Require system independence for availability.
- Mission–critical applications require standby Ethernet availability.
- Growth requirements up to thousands of ports anticipated.
- Users need access to the network from home or branch offices.
- Applications or users need 3270 terminal emulation.
- Network requires standard TCP/IP protocols (Telnet or rlogin).
- Administrators need network security features provided by DES data encryption.
- Users need transparent print.
- Need high–speed parallel printer ports.
- Administrators need to use SNMP support for LAN–attached devices.
- Need support for attachment of EIA 232, EIA 423, and EIA 422 devices.
- Asynchronous speeds to 115.2 Kbps.

Network Terminal Accelerator for AIX 4.1 and AIX 4.2

- A Micro Channel bus slot is available for the adapter.
- Need flexibility, investment protection, and distance capability of Ethernet.
- Need to maximize system performance for support of Telnet and rlogin sessions.
- Need to communicate with other systems.
- Growth requirements up to thousands of users anticipated.
- Network requires standard TCP/IP protocols.

Note: The NTA is no longer supported in AIX Versions 4.3 and later.

Customer Scenarios

The following represents some typical customer scenarios with suggested asynchronous solutions:

Real Estate Office

- Simplicity and cost are high priority.
- AIX server.
- Six to ten devices tied into the server accessing the database.
- One slot is available for asynchronous.
- Devices are less than 61 meters (200 feet) from the server.

Solution: 8– or 16–port asynchronous adapter.

Retail Point–of–Sale

- Cost per seat is high priority.
- AIX server.
- 20 or more ASCII terminals: for example, cash registers.
- One slot is available for asynchronous.
- Future expansion for additional terminals is planned.

Solution: 128–port asynchronous controller with two RANs. Future expansion with additional RANs.

Small Manufacturing/Commercial

- AIX server.
- Two to ten users or terminals locally attached.
- Limited slots are available.
- Planning to install LAN environment for longer distances from the server.

Solution: 7318 Model P10 communications server.

Large Enterprise

- High availability is a priority.
- Desktop or rack AIX servers.
- Potential for hundreds of sessions supported by AIX.
- HACMP software.
- Ethernet–attached terminal server.
- Heterogeneous network requiring multiple applications.

Solution: 7318 Model S20 communications server.

Topology Considerations

The asynchronous family of adapters offers a wide variety of choices where distance topology is concerned.

The maximum cable lengths from the standard– and direct–attach adapters is generally the distance between the port and the asynchronous device, operating at the maximum specified data rate. The 128–port adapter is measured from the adapter card to the daisy–chained RAN attached to it. With the 128–port, unlimited distances can effectively be achieved by using the EIA 422 synchronous modems to attach the RANs to the adapter.

The distance from an Ethernet port on a system 7318 communications server is dependent on standard Ethernet wiring limitations. Depending on whether the server is 10BaseT, 10Base2, or 10Base5, and depending on the number of segments used, the distance can vary between 100 and 2500 meters.

Proper cabling is extremely important and is unique to each environment.

Chapter 2. Serial Communication

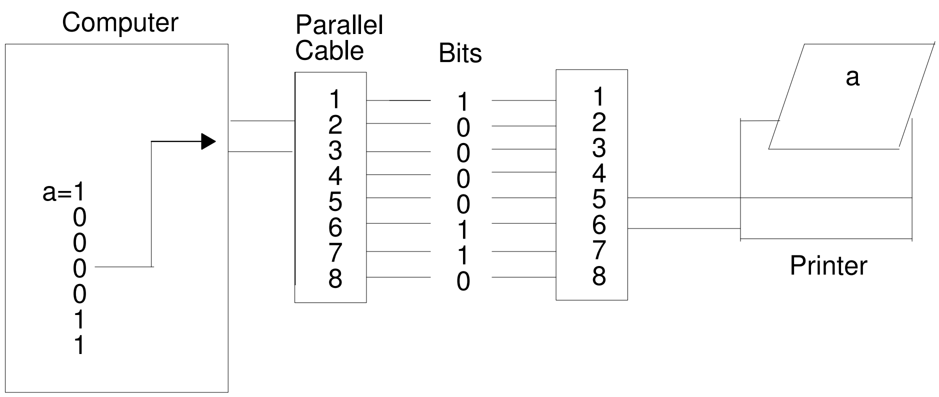
This chapter provides information on asynchronous communication standards, hardware, terminology, and concepts used throughout this book.

To understand serial ports, consider:

- Synchronization
- Serial Communication Parameters
- The RS–232–C Standard

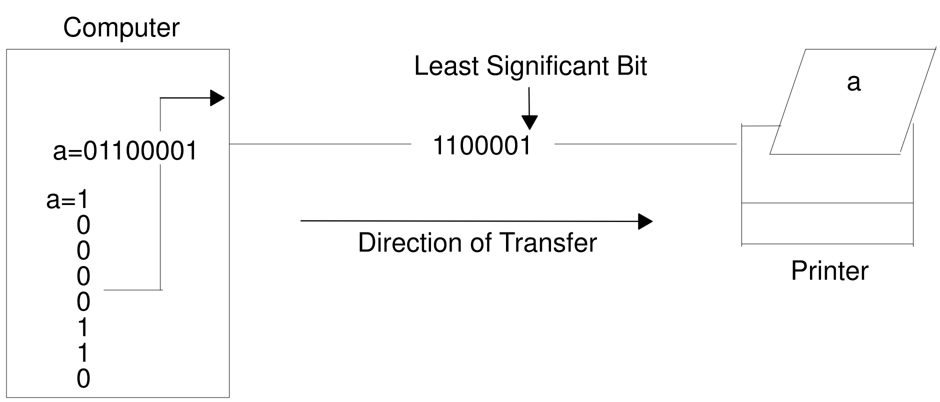
Serial ports are used to physically connect asynchronous devices to a computer. They are located on the back of the system unit, using the multiport adapter, such as, the 8–, 16–, 64– and 128–port asynchronous adapters and the 7318 terminal servers.

To understand the functionality of a serial port, it is necessary to first examine the parallel communications. A standard parallel port uses eight pins, or wires, to simultaneously transmit the data bits, making up a single character. The following figure shows the parallel transmission of the letter `a`.



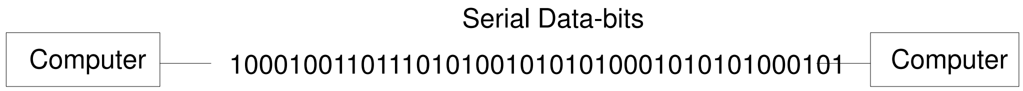
Parallel Communications Port

Serial ports require only a single pin, or wire, to send the same data character to the device. To accomplish this, the data is converted from a parallel form (sent by the computer), to a sequential form, where bits are organized one after the other in a series. The data is then transmitted to the device with the least significant bit (or zero–bit) sent first. Once received by the remote device, the data is converted back into parallel form. The following figure shows the serial transmission of the letter `a`.



Serial Communications Port

Serial transmissions of a single character are simple and straight forward; however, complications arise when a large number of characters are transmitted in series as shown in the following figure. The receiving system does not know where one character ends and the other begins. To solve this problem, both ends of the communication link must be synchronized or timed.



Serial Transmission

Synchronization

Synchronization is the process of timing the serial transmission to properly identify the data being sent. The two most common modes are synchronous and asynchronous.

Synchronous

The term synchronous is used to describe a continuous and consistent timed transfer of data blocks. These types of connections are used when large amounts of data must be transferred very quickly from one location to the other. The speed of the synchronous connection is attained by transferring data in large blocks instead of individual characters.

The data blocks are grouped and spaced in regular intervals and are preceded by special characters called `syn` or synchronous idle characters. See the following figure.

Synchronous Transmission

Once the `syn` characters are received by the remote device, they are decoded and used to synchronize the connection. Once the connection is correctly synchronized, data transmission may begin.

An analogy of this type of connection would be the transmission of a large text document. Before the document is transferred across the synchronous line, it is first broken into blocks of sentences or paragraphs. The blocks are then sent over the communication link to the remote site. With other transmission modes, the text is organized into long strings of letters (or characters) that make up the words within the sentences and paragraphs. These characters are sent over the communication link one at a time and reassembled at the remote location.

The timing needed for synchronous connections is obtained from the devices located on the communication link. All devices on the synchronous link must be set to the same clocking.

The following is a list of characteristics specific to synchronous communication:

- There are no gaps between characters being transmitted.
- Timing is supplied by modems or other devices at each end of the connection.
- Special `syn` characters precede the data being transmitted.
- `Syn` characters are used between blocks of data for timing purposes.

Asynchronous

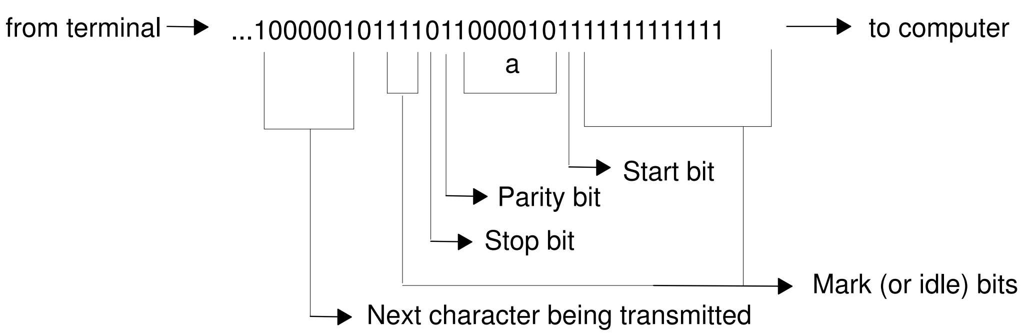
The term asynchronous is used to describe the process where transmitted data is encoded with start and stop bits, specifying the beginning and end of each character as shown in the following figure.

Start-bit									Stop-bit
0	1	2	3	4	5	6	7	8	1

Asynchronous Transmission

These additional bits provide the timing or synchronization for the connection by indicating when a complete character has been sent or received; thus, timing for each character begins with the start bit and ends with the stop bit.

When gaps appear between character transmissions, the asynchronous line is said to be in a mark state. A mark is a binary 1 (or negative voltage) that is sent during periods of inactivity on the line as shown in the following figure.



Mark (idle) Bits in the Data Stream

When the mark state is interrupted by a positive voltage (a binary 0), the receiving system knows that data characters are going to follow. It is for this reason that the start bit, which precedes the data character, is always a space bit (binary 0) and that the stop bit, which signals the end of a character, is always a mark bit (binary 1).

The following is a list of characteristics specific to asynchronous communication:

- Each character is preceded by a start bit and followed by one or more stop bits.
- Gaps or spaces between characters may exist.

Serial Communication Parameters

The following describes the parameters used during serial communication.

Bits–Per–Character

The number of bits–per–character (bpc) indicates the number of bits used to represent a single data character during serial communication. This number does not reflect the total amount of parity, stop, or start bits included with the character. Two possible settings for bpc are 7 and 8.

When using the seven bits–per–character setting, it is possible to only send the first 128 characters (0–127) of the Standard ASCII character set. Each of these characters is represented by seven data bits. The eight bits–per–character must be used in order to send the ASCII Extended character set (128–255). Each of these characters may only be represented using eight data bits. Refer to the Binary Numbers figure for a graphic representation.

Decimal Value	Binary Number
0	0
1	1
3	11
7	111
15	1111
31	11111
63	111111
127	1111111
255	11111111

Binary Numbers

Bits–Per–Second (bps)

Bits–per–second (bps) is the number of data bits (binary 1s and 0s) that are transmitted, per second, over the communication line.

Baud Rate

The baud rate is the number of times per second a serial communication signal changes states; a state being either a voltage level, a frequency, or a frequency phase angle. If the signal changes once for each data, then one bps is equal to one baud. For example, a 300 baud modem changes its states 300 times a second.

Parity

The parity bit, unlike the start and stop bits, is an optional parameter, used in serial communications to determine if the data character being transmitted is correctly received by the remote device.

Start-bit Stop-bit

0	1	2	3	4	5	6	7	8 or parity	1

Parity

The parity bit can have one of the following five specifications:

none Specifies that the local system must not create a parity bit for data characters being transmitted. It also indicates that the local system does not check for a parity bit in data received from a remote host.

even Specifies that the total number of binary 1s, in a single character, adds up to an even number. If they do not, the parity bit must be a 1 to ensure that the total number of binary 1s is even.

For example, if the letter a (binary 1100001) is transmitted under even parity, the sending system adds the number of binary 1s, which in this case is three, and makes the parity bit a 1 in order to maintain an even number of binary 1s. If the letter A (binary 1000001) is transmitted under the same circumstances, the parity bit would be a 0, thus keeping the total number of binary 1s an even number.

odd Operates under the same guidelines as even parity except that the total number of binary 1s must be an odd number.

space Specifies that the parity bit will always be a binary zero. When used for error detection, space will indicate a problem only if the parity bit is not a zero. Another term used for space parity is bit trimming, which is derived from its use as a filler for seven–bit data being transmitted to a device which can only accept eight bit data. Such devices see the space parity bit as an additional data bit for the transmitted character.

mark Operates under the same guidelines as space parity except that the parity bit is always a binary 1. The mark parity bit acts only as a filler.

Most modern serial communications devices (modems, terminals, etc.) use a 10–bit transmission character:

```
     1  bit  for the start bit
7 or 8  bits for the data character
1 or 0  bits for the parity setting
2 or 1  bits for the stop bits
----------------------------------
    10  bits for total character transmission.
```

If the number of data bits is set to eight:

```
8 bpc + 1 start +1 stop + 0 parity = 10 bits
```

there are no bit positions left for parity; therefore, parity must be set to none.

If the number of data bits is set to seven, parity can be even, odd, or none.

```
7 bpc + 1 start + 1 stop + 1 parity = 10 bits
```

Another alternative is having parity set to none and stop bits set to two (this is rarely used).

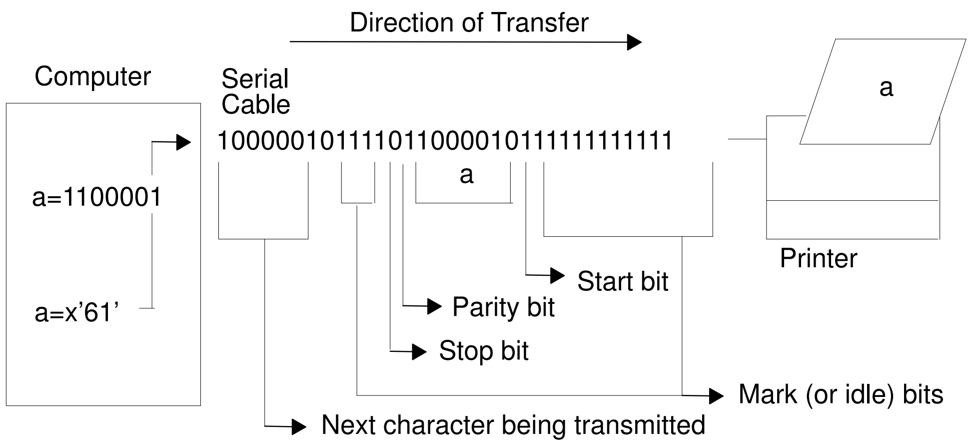
```
7 bpc + 1 start + 2 stop + 0 parity = 10 bits.
```

If the parity bit is used for error detection on a communication line, both the sending and receiving systems must be configured identically to avoid possible problems during transmissions.

Start, Stop, and Mark Bits

The start and stop bits are used in asynchronous communication as a means of timing or synchronizing the data characters being transmitted. Without the use of these bits, the sending and receiving systems will not know where one character ends and another begins.

Another bit used to separate data characters during transmission is the idle bit. This bit, a binary 1, is transmitted when the communication line is idle and no characters are being sent or received. When a start bit (binary 0) is received by the system, it knows that character data bits will follow until a stop bit (binary 1) is received. See the following figure.



Start, Stop, and Mark Bits

The RS–232–D Standard

The RS–232–D standard was developed in 1969 by the Electronics Industry Association (EIA) to specify the connections between a computer and a modem. The term itself is an acronym which can be read as follows:

```
Recommended Standard (RS), ID number 232 revision D.
```

RS–232–D specifies the characteristics of the physical and electrical connections between two devices. Names and abbreviations are assigned to each pin or wire necessary for serial communications, for example:

Signal	Symbol	Pin
Transmit Data	TxD	2
Receive Data	RxD	3
Request to Send	RTS	4
Clear to Send	CTS	5
Data Set Read	DSR	6
Signal Ground	SG	7
Carrier Detected	CD	8
Data Terminal Ready	DTR	20
Ring Indicator	RI	22

In RS–232–D, devices using pin 2 (TxD) for output (for example, computers and terminals) are given the name data terminal equipment (DTE). Devices using pin 2 (TxD) for input (for example, modems) are given the name data communication equipment (DCE).

RS–232–D also specifies the connectors. A DTE device normally has male connectors while DCE devices have female connectors. This standard is not always adhered to by manufacturers; therefore users should always review the device documentation before cable connection.

Communication Methods

Simplex, or one–way communication, is the simplest connection form between two devices. This mode of communication allows data to be transmitted in only one direction and requires only two lines be connected, for example, TxD (or RxD) and SG.

There are two forms of two–way communications, half–duplex and full–duplex. A connection in half–duplex mode allows data to be transmitted in two directions but not simultaneously.

An analogy of half–duplex would be the use of a CB–radio where two–way communication is possible but only one person can speak at a time.

In full–duplex, or duplex mode, data communication can take place in two directions simultaneously. An analogy for full–duplex is a typical telephone conversation between two people where both persons are talking at the same time.

Flow Control

Serial devices, such as printers and modems, do not process data as quickly or efficiently as the computers they are connected to. Some type of data flow control is needed by the serial device to limit the amount of data transmitted by the system.

The term flow control is used to describe the method where a serial device controls the amount of data being transmitted to itself. The three types of flow control discussed in this section are:

- RTS/CTS (hardware flow control)
- DTR/DSR (hardware flow control)
- XON/XOFF (software flow control).

RTS/CTS

Ready to send/clear to send (RTS/CTS) is sometimes called pacing or hardware handshaking instead of flow control. The term hardware handshaking comes from the use of cabling and voltages as a method of data transmission control. Unlike XON/XOFF, which sends control characters in the data stream, RTS/CTS uses positive and negative voltages along dedicated pins or wires in the device cabling.

DTR/DSR

Data terminal ready (DTR), another form of hardware flow control, is normally generated by the devices, such as printers to indicate that they are ready to communicate with the system. This signal is used in conjunction with data set ready (DSR) generated by the system to control data flow.

XON/XOFF

Transmitter on/transmitter off (XON/XOFF) flow controls involves the sending of data transmission control characters along the data stream (TxD and RxD). For this reason it is referred to as software flow control.

When data is sent to a modem, it is placed in a buffer. Just before that buffer reaches its maximum capacity, the modem will send an XOFF character to the system and the system will stop transmitting the data. When the modem's buffer is almost empty and ready for more data, it will send an XON character back to the system causing more data to be sent.

With both types of hardware flow control, a positive voltage means data transmission is allowed while a negative voltage signifies that data transmission should be suspended.

Configuring a Port for RTS/CTS Hardware Handshaking

Modems attached to the server operating at a speed of 9600 or above are recommended to use RTS/CTS hardware handshaking instead of XON/XOFF flow control. This will avoid buffer overrun in a system with limited resources. RTS is not a default value on any tty port and must be set by the system administrator accordingly.

AIX Version 3.2.5 provides the following methods of adding RTS to the tty port:

Method 1:

1. Edit the **/etc/uucp/Devices** file and add the line:

```
Direct tty# –Any direct
```

2. Save and exit the file.
3. Enter: `cu -ml tty# &`
4. The previous command will open the tty port and run as a background process, thus keeping carrier detect high on the port and allowing RTS to be placed on the device stack.

   Enter: `stty add rts < /dev/tty#`
5. Verify that RTS is now on the tty stack:

   Enter: `stty get < /dev/tty#`

   Stack information for a tty on the native S1 port should resemble:

   ```
    rs, dtropen, rts, xon, posix
   ```
6. Stop the cu process.

Method 2:
Use the following C code to add RTS to a tty port on an AIX 3.2 system. To make the program more permanent, insert the file name of the compiled version (complete with path) at the end of your **/etc/rc** file and the changes will take effect again at next reboot. This is necessary because RTS is removed from the tty stack after a reboot.

```
/* C Program to add RTS discipline to tty port(s).

NOTE: This program is supplied "as is" and is NOT supported.
It is intended as an aid to administrators only.
To create: vi addrts.c <enter> To compile: cc -o addrts addrts.c
/***************************************************************
                        Program starts now
************************************************************** */
#include <stdio.h>
#include <fcntl.h>
#include <termios.h>
#include <sys/tty.h> main()
{
   int fd;
   fd = open("/dev/tty0", O_NDELAY|O_RDWR);
   ioctl(fd, TXADDCD, "rts");
   /* adds rts to tty0 */
   close(fd);
}
/*****************************************************************
                         Program ends now
 ************************************************************** */
```

To enable RTS/CTS for a port in AIX Version 4, use the following steps:

1. Use the **smit tty** fast path.
2. Select **Change / Show Characteristics of a TTY**.
3. Select the tty on which RTS/CTS is to be enabled.
4. Set the FLOW CONTROL to be used field to **rts**.
5. Select **Do**.
6. Exit SMIT.

Chapter 3. Asynchronous Devices and Software

This section describes asynchronous devices and related necessary software. Asynchronous devices include terminals, printers, plotters, or any device that conforms to the RS-232C standard.

BQ306 Asynchronous Terminals

The following provides general information concerning the use of BQ306 asynchronous terminals with the AIX system's tty subsystem.The following information is specific to the BQ306 terminal:

- Overall Characteristics on page 3–1
- Connectivity Options on page 3–2
- Attributes for the BQ306 ASCII Display Station on page 3–3
- Problem Determination
  - Problems with the Keyboard on page 3–3
  - Problems with the Display on page 3–3
  - Miscellaneous Problems on page 3–3
  - Common Questions on page 3–3

Overall Characteristics

All BQ306 ASCII displays supports the following capabilities:

- EIA-232 interface.
- EIA-422.
- 24x80, 25x80, 24x132, and 25x132 screen sizes.
- Connectivity to the system unit.
  - The BQ306 can attach to the AIX system, but are intended for PC attachment in PC mode.

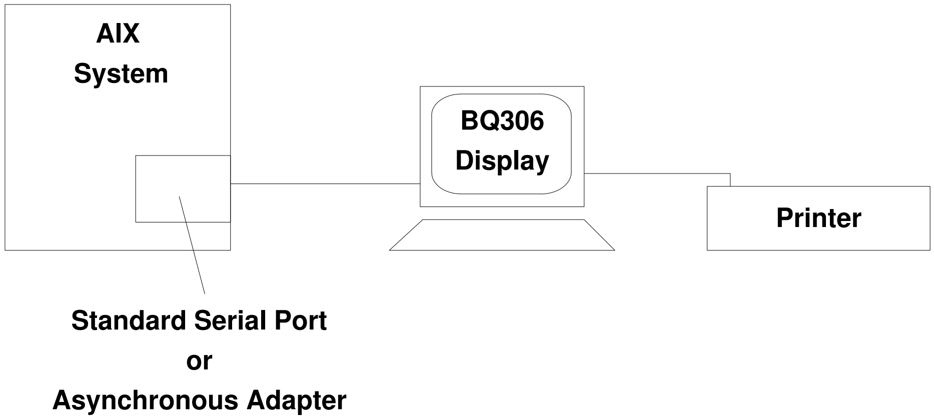
For a detailed description of the BQ306 display, refer to the following manuals:

- *BQ303 User Guide*
- *User's Guide for WSYE 50/50+ Emulation*

Connectivity Options

The BQ306 display can connect directly, or through a modem, to an AIX system. The connection to the AIX system can be made to one of the native serial ports or to an asynchronous adapter. Additionally, a printer can be connected to the BQ306 display and is supported by AIX as terminal attached printing.

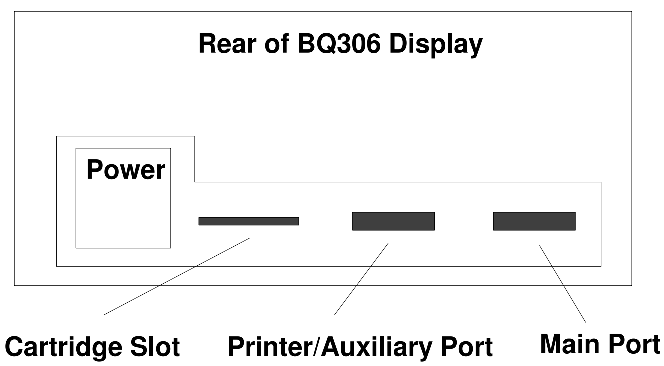
See the following figure for an illustration of the BQ306 connectivity options.



BQ306 Connectivity

The BQ306 display has two ports: the main port and the auxiliary port. In the scenario discussed previously, the AIX system is connected to the main port, and the printer is connected to the auxiliary port.

See the following figure for an illustration of the physical location of the main and auxiliary ports.



Locations of Main and Auxiliary Ports

The main port is DTE (DB-25) and does not require an interposer. The printer port already has an interposer (DCE); therefore, when connecting a serial printer to the BQ306 display, no interposer is necessary.

Attributes for the BQ306 ASCII Display Station

For a detailed description of the BQ306 setup values, refer to "How to Personalize a BQ303 or a BQ306 Terminal" in the "Terminals and Printers Configuration Guide".

Problem Determination

Problems with the Keyboard

If you choose PC Keyboard, you must put, in the general setup parameters select ISO latin–1 for Char Set, select Yes for PC Character set, IBM437 for PC Pref Character and in the Keyboard2 Setup parameters, Select ANSI for F.key mode. See the BQ303 User's Guide.

Problems with the Display

Double characters are displayed.

Possible Causes:

The **Operating Mode** attribute is not set correctly.

Procedures for Recovery:

Change **Operating Mode** attribute from char to echo.

The BQ306 displays backwards question marks.

Possible Causes:

- Frame error.
- Mismatch of the emulation type between the system and the display.

Procedures for Recovery:

- Insure that the **Parity**, **Word Length**, **Stop Bits**, and **Line Speed** attributes of the terminal match the AIX system tty line characteristics.
- Check with your administrator to verify the emulation type.

The screen is wavy.

Possible Causes:

The refresh rate of the terminal is incorrect.

Procedures for Recovery:

Vary the refresh rate between between 50 and 60 hertz. Press Ctrl-Shift-Hold. The BQ306 displays the keyboard test pattern. Press Ctrl-F10. The BQ306 displays A or B (refresh rate). Press F1 to toggle between A or B. Power off and on the BQ306.

Miscellaneous Problems

The BQ306 terminal does not communicate to the AIX system

Possible Causes:

- The **Line Control** attribute is not set to IPRTS.
- The cable is attached to the auxiliary port rather than the main port.
- The Transmit Data (TxD), Receive Data (RxD) pins do not match the AIX system's communications port.
- The system does not have the communications port defined and enabled.

Procedures for Recovery:

- Turn on **Status Line**. If `COMM NOT READY 2` is displayed on the status line, change the **Line Control** attribute to IPRTS.

- Move the cable to the main port. The main port is on the right-hand connector, looking from the back.
- Use correct cabling: Asynchronous cable EIA-232/V.24 cable D and Printer/Terminal Interposer EIA-232 cable E or equivalent.
- Use SMIT or the **penable** command to enable the port. Refer to “Change/Show Characteristics of a TTY” in *AIX System Management Guide: Communications and Networks* for further information.

Second session not working when using Connectivity Cartridge in Dual Session by Dual Port.

Possible Causes:

The directions of pins 2 (TxD) and 3 (RxD) may be reversed.

Procedures for Recovery:

Switch pins 2 and 3 on the communications cable attached to the auxiliary port.

The BQ306 display is missing or losing characters.

Possible Causes:

- Pacing or data flow problem
- Incorrect **TERM** environment variable

Procedures for Recovery:

- To confirm that a pacing problem exists, reduce line speed to 2400 or 4800. If problems persist, activate pacing at the BQ306 and the AIX system.
- Check **TERM** environment variable and change it so that it matches the terminal.

The BQ306 terminal is losing its setup values.

Possible Causes:

- Application could be changing setup values using BQ306 commands: Set Control, Create Viewport.
- User changes setup options in setup menu, but does not save the changes.

Procedures for Recovery:

- Problem must be corrected at the system.
- Place cursor on the Save option and press the space bar to save new setup.

Cannot place the BQ306 terminal in setup mode.

Possible Causes:

- Mismatch of keyboards with the video element.
- DEC VT200/VT100/VT52 emulation is selected, and keyboard does not have DEC Keycap kit.

Procedures for Recovery:

- Impossible to enter setup. Exchange the 84-key ASCII keyboard with the proper keyboard.
- Press the F3 key, the location of the setup key when in DEC emulation.

Cannot print or not is printing correctly.

Possible Causes:

- Attached printer is a parallel type printer, not a serial printer.
- Incorrect cable being used with printer.

Procedures for Recovery:

- Printer must be a serial printer.
- Use correct cable.
- Printer is not configured with pacing.

Line-drawing characters do not display.

Possible Causes:

A character set mismatch between system and terminal.

Procedures for Recovery:

The BQ306 offers the following character sets:

- Special Graphics
- PC 00437
- PC 00850
- ISO 8859/1.2

Determine which character set is defined on the BQ306, and which character set the system is expecting. The character set used by the system and the terminal must match.

The BQ306 has extra blank lines.

Procedures for Recovery:

- From the terminal, set Auto LF = OFF in general setup menu.
- From the AIX system, send the Set Control 3 command (`ESC " 9 = b`).

Common Questions About the BQ306 Terminal

How do you change the cursor style to Block, Underscore, or Blink?

- Use Display Setup Parameters and put Cursor Parameter to the appropriate value.

How do you change the status line (None, Machine Status, Host Message)?

- Use Display Setup Parameters and put Status Line Parameter to the appropriate value.

Does the auxiliary port support an EIA-422 interface?

- No, only supports EIA-232.

What is the difference between EIA-232 and EIA-422?

- EIA-232 defines 9 pins. Two pins are used for Transmit and Receive Data (TxD, RxD). EIA-422 defines 5 pins. Four pins are used for Transmit and Receive Data (TD+, TD–, RD+, RD–).
- EIA-232C can attach 50–75 feet (15.2–22.9 meters), EIA-232D can attach 200 feet (61 meters). EIA-422 can attach 4000 feet (1219 meters).

How do you change the function keys?

  - From the system: Issue the appropiate command sequence.
  - From the terminal: Use User setup in (DEF F) mode.
- Models 510 or 610 – not available.

How many function keys are available on the BQ306?

- 105 Key ANSI Keyboard
  - 19 physical keys, (Shift–F.key) redefinable from user setup with function key F14.
- PC style Keyboard

– 16 physical keys, (Shift–F.key) redefinable from user setup with function key F14.

How do you reset/clear a function key to default values?

- From User setup: Press Ctrl–Clear key.
- From the application: Send 'Reset All Default Default Function Keys' (`ESC SP t`) command.

Can an input device be attached to the auxiliary port?

Yes, the auxiliary port is bidirectional (for models 310, 410, 510 and 610).

Can a selected key (setup, break, ctrl...) be physically disabled?

- No, this requires a keystopper under keycap.

Can a selected key be remapped to another key.

- No.

Can you use a BQ306 model 310, 410, 510, 610 keyboard on a BQ306 model 110?

- No, you cannot mix any combination of 84-, 101-, and 102-key keyboards. The keyboard must match video display.

How do you automatically dial a modem?

- Use the connectivity cartridge, or assign an unused function key. Cartridge requires 6 keystrokes, unused function key requires 2 to 3 keystrokes.

How do you activate or deactivate the printer from the keyboard?

- Press Ctrl–Trace key.

How do you display CAPS information?

- Turn on the status line.

How do you change the screen size?

- From the terminal: Use the setup menu.
- From the application: use the Create Viewport Command:

24 X 80	`ESC SP r ! ! SP 8 " P`	
25 X 80	`ESC SP r ! ! SP 9 " P`	
28 X 80	`ESC SP r ! ! SP < " P`	(Expansion Cartridge)
24 X 132	`ESC SP r ! ! SP 8 $ D`	
25 X 132	`ESC SP r ! ! SP 9 $ D`	
28 X 132	`ESC SP r ! ! SP < $ D`	(Expansion Cartridge)

Are communication cables included?

- No.

What is the command to turn the printer on/off from the system?

- Begin/End Pass through mode (**DLE DC2** and **DLE DC4**)

Can you switch a BQ306 between a 110 and a 220 volt?

- No, each country has the appropriate video display with the proper power supply.

How can the BQ306 display line-drawing characters?

1. Assign G0/G1 using Select Character Set command.
2. Select either G0 or G1 by:

   8-bit mode, turn on most significant bit.

   7-bit mode, use SO/SI ASCII control characters.

What is the gender of the main and auxiliary ports?

- Both are 25-pin D-Shell female.

Which port is the main communication port?

- Looking at the back of the BQ306, the main port is on the right-hand side, closest to the edge. The auxiliary port is closest to the power cord.

How do you get a dual session?

- Use the Cartridge for Connectivity. Select `Dual Session by Dual Port` in **SESSION ASSIGNMENT** setup menu.

Is there a keyboard extension cable available?

- Bull does not offer or support any changes between the keyboard and display elements.

Is there support for a 7695 Bar Code Reader placed between keyboard and display elements?

- Only the models 51/61 supports the 7696 Bar Code Reader. Other bar code reader products are available that can be placed between the keyboard and display element, however, Bull supports only the 7696 Bar Code Reader.

Does the BQ306 offer a CRT Saver/Screen Saver?

- Yes, CRT is selectable between No Save or 15 minutes. The CRT is selectable by the user in the General Setup menu, or by the system using the Set Control 2 command.

Is a tilt/swivel stand included?

- Yes.

Does the BQ306 have a data entry keyboard?

- No.

Dynamic Screen Utility

The dynamic screen (dscreen) utility allows a single physical terminal to have several virtual terminal sessions or screens running at one time. This utility works with 128-port and ISA 8-port adapters. It is mainly intended for use with terminals that have two or more pages of screen memory. For example, the Bull BQ306 display with the Expansion Cartridge.

With such terminals, switching between virtual screens also switches between physical terminal screen pages allowing each virtual screen's image to be saved and restored. On terminals without multiple pages of screen memory, **dscreen** can still be used to switch among virtual screen sessions although the appearance of the screen will not be maintained.

For full support of **dscreen**, the terminal must be able to switch internal screen pages on command and must remember the cursor position for each page. While **dscreen** will work on both smart and dumb terminals, screen images are not saved during screen changes on dumb terminals.

dscreen Terminal Configuration Information File

The **dscreen** terminal configuration information file (or dsinfo file) is used to define a different set of keys to be used with **dscreen**. This might be done, for example, when the originally defined **dscreen** keys conflict with a software application in use on the system.

The dsinfo file terminal type assumes a single page of screen memory. Therefore, if a terminal supports additional pages of screen memory, the dsinfo file must be customized to use the appropriate sequence for page memory control. Consult the appropriate terminal reference guide for the specific control sequence.

The default dsinfo file is the **/usr/lbin/tty/dsinfo** file. Use the **-i** flag to specify a different dsinfo file. This remainder of this section will refer to the default dsinfo file. However, the same information applies to any customized dsinfo file you create.

For more infomation concerning the dsinfo file, refer to “Dynamic Screen Assignment” on page 3–10.

Key Action Assignments

When **dscreen** is run, it starts a virtual screen. Some of the keys on the terminal keyboard will not be passed through to the virtual screen; instead, **dscreen** will intercept these keys and perform certain actions when they are pressed. The actions include:

Select Selects a specified screen

Block Blocks all input and output.

New Starts a new screen session.

End Ends dscreen.

Quit Quits dscreen.

Previous Switches to previous screen.

List Lists the dscreen assigned keys and their actions.

The function of each key is dependent upon the terminal and the terminal description in the **/usr/lbin/tty/dsinfo** file.

Select Keys

When a new virtual screen is created, it is assigned a select key. Pressing the select key causes the following actions:

- A switch from the physical terminal to the video page associated with the particular virtual screen.
- Input and output is directed appropriately between the physical terminal and the virtual screen.

Once all of the select keys defined in the **dsinfo** file have virtual screens assigned to them, no more screens may be created. Individual screen sessions end when the original shell process exits. This frees the associated select key for use with another virtual screen. The **dscreen** utility is ended when there are no more active screens.

Block Keys

Block keys are used to stop output in a fashion similar to Ctrl-S key when using IXON flow control. The purpose of these keys is to allow for transparently setting up terminal sessions on two computers using a terminal that has two serial ports.

New Keys

Pressing a new screen key creates a new logical screen and assigns it to one of the select keys. Each new screen requires:

- A select key as defined in the dsinfo file.
- A **dscreen** pseudo terminal device.
- Enough memory for the various structures used in screen tracking.
- A process to run the shell from.

If any of these are not available, the new screen operation will fail with a message indicating the reason for the failure.

End and Quit Keys

Pressing an end key will cause the following to occur:

- Send a **SIGHUP** signal to all the screen sessions
- Clean up
- Exit with a status of 0.

Pressing a quit key will perform the same actions but will exit with status of 1.

Previous Key

Pressing a previous key will switch the terminal to the screen that was last displayed.

Notes:

1. Do not switch screens when the current screen is being written to; an escape sequence may be truncated and leave the terminal in an unknown state.
2. Some terminal displays may save the cursor position for individual screens but may not save other states such as insert mode, inverse video, etc. If this is the case, users should avoid these modes while switching screens.

List Key

Pressing a list key will display a list of keys and their actions on the terminal display. Only those keys recognized by **dscreen** will be shown. When a new screen is created using **dscreen**, the message `Press KEY for help`, where *KEY* is the name of the list key displayed on the terminal. Note that the message is displayed *only* if there is a list key defined.

Dynamic Screen Assignment

The terminal description entry in the **/usr/lbin/tty/dsinfo** file will have the same number of screen selection keys as the terminal has physical screen pages. If more screen selection keys are defined than the number of physical screen pages, **dscreen** will dynamically assign physical screen pages to virtual screens.

When a virtual screen is selected which does not have an associated page of screen memory, **dscreen** assigns the least recently used physical screen to the virtual screen. Depending on the specifications maintained in the **/usr/lbin/tty/dsinfo** description file, an indication that the physical screen is connected to a different virtual screen may be noticeable; for example, the screen is cleared.

dsinfo File

The **dsinfo** file is a database of terminal descriptions used by the **dscreen** multiple screen utility. The file contains the following information:

- **dscreen** key definitions and the functions they perform.
- Number of screen memory pages for the terminal.
- Terminal specific control sequences sent or received to control the above features.

The terminal type entries in the default **dsinfo** file resemble the following BQ306 ASCII terminal values.

Note: \r may be used in place of ^M.

```
# The Cartridge for Expansion (pn 64F9314) needed for this entry
BQ306|Bull BQ306 model,
dsks=\E!a^M|Shift-F1|,              # Selects first screen
dsks=\E!b^M|Shift-F2|,              # Selects second screen
dsks=\E!c^M|Shift-F3|,              # Selects third screen
dsks=\E!d^M|Shift-F4|,              # Selects fourth screen
dskc=\E!e^M|Shift-F5|,              # Creates a new screen
dske=\E!f^M|Shift-F6|\E pA\EH\EJ,   # Go to screen 1 and end
dskl=\E!g^M|Shift-F7|,              # Lists function keys (help)
dskp=\E!h^M|Shift-F8|,              # Go to previous screen
dskq=\E!i^M|Shift-F9|\E pA\EH\EJ,   # Go to screen 1 and quit
dsp=\E pA|\EH\EJ,                   # Terminal sequence for screen 1
dsp=\E pB|\EH\EJ,                   # Terminal sequence for screen 2
dsp=\E pC|\EH\EJ,                   # Terminal sequence for screen 3
dsp=\E pD|\EH\EJ,                   # Terminal sequence for screen 4
dst=10,                             # Allow 1 second timeout buffer
```

Entry Format for dsinfo

Entries in the dsinfo file consist of comma-separated fields. The first field is a list of alternate names for the terminal, each name is separated by a pipe (|) character. Any text preceeded by a pound (#) character is regarded as a comment and ignored by dscreen. The remaining fields are strings describing the capabilities of the terminal to the **dscreen** utility. Within these strings, the following escape codes are recognized:

Escape Sequence	Description
\E,\e	escape character

\n,\l	newline (or linefeed) character
\r	carriage return
\t	tab character
\b	backspace character
\f	formfeed character
\s	space character
\nnn	character with octal value nnn
^x	Ctrl-*x* for any appropriate *x* value

Any other character preceded by a backslash will yield the character itself. The strings are entered as *type=string*, where type is the type of string as listed below, and string is the string value.

It is important that the entry fields in the **dsinfo** file be separated by commas. If a comma is omitted or truncated from the end of a **dsinfo** file entry, the file will become unreadable by the **dscreen** utility and an error will be returned to the display.

String Types

The string types are as follows:

dskx A string type that starts with *dsk* describes a key. The type must be four letters long, and the fourth letter *x* indicates what action is taken when the key is received. The key types are:

Type	Action
dsks	Switch Screens
dskb	Block Input and Output
dske	End **dscreen**
dskq	Quit **dscreen** (exit status=1)
dskc	Create New Screen
dskp	Switch to Previous Screen
dskl	List Keys and Actions

Any other key type (that is, a string type *dskx* that doesn't end in s, b, e, q, p, or l) will cause no internal **dscreen** action, but will show up in the key listing and will be recognized and acted upon. A type of dskn (n for No Operation) should be used when no internal **dscreen** action is desired.

The value string for each key has three substrings, which are separated by pipe (|) characters.

Note: Use \| to include the | character in one of the substrings.

The first substring is the sequence of characters that the terminal sends when the key is pressed. The second substring is a label for the key that is printed when a list of keys is displayed. The third substring is a sequence of characters that **dscreen** sends to the terminal when this key is pressed before performing the action this key requests.

dsp A string type of dsp describes a physical screen in the terminal. One dsp string should be present for each physical screen in the terminal. The value string for each physical screen has two substrings, which are separated by a pipe (|) character.

The first substring is the sequence of characters to send to the terminal to display and output to the physical page on the terminal.

The second substring is sent to the terminal when the page is used for something new. This second substring is often set to the clear screen sequence. It is sent under the following two conditions:

1. When a new virtual terminal session is being created.
2. When there are more virtual terminals than there are physical screens. If a virtual terminal is selected which requires **dscreen** to reuse one of the physical screens, it will send this sequence to the screen to indicate that the screen contents do not match the output of the virtual terminal connected.

Note: Running with more virtual terminals than physical screens can be confusing and is not recommended; it can be avoided by defining no more screen selection keys (dsks=) than physical screens (dsp=) in the dsinfo entry.

dst A String with a type of dst adjusts **dscreen**'s input timeout. The value of the string is a decimal number. The timeout value is in tenths of seconds and has a maximum value of 255 (default=**1** [or .1 seconds]).

When **dscreen** recognizes a prefix of an input key sequence but does not have all the characters of the sequence, it will wait for more characters to be sent until it is recognizable. If the timeout occurs before more characters are received, the characters are sent on to the virtual screen and **dscreen** will not consider these characters as part of an input key sequence.

It may be necessary to raise this value if one or more of the keys **dscreen** is to trigger on is actually a number of keystrokes (that is assigning Ctrl-Z 1, Ctrl-Z 2, Ctrl-Z 3, etc. for screen selection, and Ctrl-Z N for new screen and so on).

Example 1

The following example **/usr/lbin/tty/dsinfo** entry is for a Wyse-60 with three screen sessions:

```
wy60|wyse60|wyse model 60,
dsks=^A`^M|Shift-F1|,
dsks=^Aa^M|Shift-F2|,
dsks=^Ab^M|Shift-F3|,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew0|\E+,
dsp=\Ew1|\E+,
dsp=\Ew2|\E+,
```

With this entry:

- Shift-F1 through Shift-F3 are used for selecting screens 1 through 3.
- Ctrl-F1 creates a new screen.
- Ctrl-F2 sends: `Esc w 0 Esc +` to the screen (switching to window 0 and clearing the screen) and then end **dscreen**.
- Ctrl-F3 will list the keys and their functions.

Each time a physical screen is used for a new screen, the sequence `Esc +` will be sent to the terminal, which will clear the screen.

Example 2

This example is for a Wyse-60 with three screen sessions, but one of the screens is on a second computer communicating through the second serial port on the terminal:

```
wy60-1|wyse60-1|wyse model 60 - first serial port
dsks=^A`^M|Shift-F1|,
dsks=^Aa^M|Shift-F2|,
dsks=^Ab^M|Shift-F3|\Ed#^Ab\r^T\Ee9,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ed#\201^T\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew0|\E+,dsp=\Ew1|\E+,
wy60-2|wyse60-2|wyse model 60 - second serial port
dsks=^A`^M|Shift-F1|\Ed#^A`\r^T\Ee8,
dsks=^Aa^M|Shift-F2|\Ed#^Aa\r^T\Ee8,
dsks=^Ab^M|Shift-F3|,
dskc=\200|Ctrl-F1|,
dske=\201|Ctrl-F2|\Ed#\201^T\Ew0\E+,
dskl=\202|Ctrl-F3|,
dsp=\Ew2|\E+,
```

dscreen must be run on both computers, with terminal type wy60-1 on the first computer and terminal type wy60-2 on the second computer (using the **-t** option to **dscreen**). The wy60-1 entry will be examined first.

The first two key entries are unchanged from the original wy60 entry. The third key, however, has type dskb, which means block both input and output. When this key is pressed, the sequence:

```
Esc d # Ctrl-A b CR Ctrl-T Esc e 9
```

is sent to the terminal; after this output is blocked and **dscreen** continues scanning input for key sequences but discards all other input.

The sequence `Esc d #` puts the terminal in Transparent Print Mode, which echoes all characters up to a Ctrl-T out through the other serial port.

The characters `Ctrl-A b CR` are sent out the other serial port, informing the **dscreen** process on the other computer that it should activate the window associated with the Shift-F3 key.

The Ctrl-T takes exits the Transparent Print Mode. The `Esc e 9` causes the terminal to switch to the other AUX serial port for data communications.

At this point the other computer takes over and sends an `Esc w 2` to switch to the third physical screen, and then resumes normal communication.

The wy60-2 entry follows the same general pattern for keys Shift-F1 and Shift-F2:

- Switch to transparent print mode
- Send function key string to other computer
- Switch transparent print off
- Switch to the other serial port.

The end key Ctrl-F2 works the same for both computers; it sends the end key sequence to the other computer through the transparent print mechanism, switches the terminal to window 0, clears the screen, then exits.

Transparent Printing

Most terminals have an auxiliary port that can be connected to a serial printer. These terminals support two print modes, Auxiliary and Transparent. If both print modes are OFF, data received by the terminal is simply displayed on the screen. With Auxiliary print mode ON, data received by the terminal is displayed on the screen, and is also transmitted to the printer. With Transparent Print Mode ON, the terminal transmits data received directly to the printer, without displaying it on the screen.

Transparent printing allows you to use your terminal in a normal manner, while information is also being sent over the same serial connection from the host to the printer connected to the terminal's auxiliary printer port. This is transparent printing. The transparent printing software determines whether packets of data are bound for the screen or for the printer, and precedes data bound for the printer with the Transparent Print Mode ON command, and follows it with the Transparent Print Mode OFF command.

Data for the terminal screen has the highest priority, and data is sent to the printer only when there is a break in information being sent to the screen. If continuous data is being transmitted to the terminal device, nothing gets sent to the printer.

Whenever an auxiliary printer port is used, flow control to the printer becomes an issue. If the printer falls behind and invokes flow control, output to both the printer and the terminal is stopped. The transparent print feature provides three parameters, accessible through smit, to limit printer output and avoid this situation.

The Transparent Print Maximum Characters per Second parameter limits the maximum printer port character-per-second data rate. This number should be set to the minimum character rate the printer can sustain in typical use.

The Transparent Print Maximum Character Packet Size parameter limits the number of characters queued to the printer ahead of terminal output. Lower numbers increase system overhead, higher numbers result in keystroke echo delays. A value of 50 is generally a good compromise at 9600 baud.

The Transparent Print Printer Buffer Size parameter should be set to a value just below the printer's buffer size. After a period of inactivity, the driver will burst up to this many characters to the printer to fill the print buffer before slowing to the maxcps rate.

The printer on/off strings are also set using smit. A cable must be connected between the auxiliary port of the terminal and the printer. The baud rate on the terminal auxiliary port and the printer must be the same, and the printer and the auxiliary port of the terminal must use the same handshaking mode. The auxiliary port must also be enabled. If your terminal is not one of those directly supported, you must know the escape sequence of your terminal.

Refer to your terminal and printer manuals for connection information, escape codes, and to see what handshaking modes are supported (for example, xon/xoff, busy/ready, rts/cts, etc.). Printer devices (xtty1, etc.) must not be in either the **/etc/inittab** or **/etc/ttys** files, and must not be enabled.

Transparent Print Activation

Transparent printing works with Micro Channel 128-port, ISA 8-port, and ISA 128-port adapters only. Use smit to enable transparent printing for a tty device either while creating the tty device, or with the Change/Show Characteristics of a tty option. The following parameters control the behavior of transparent print devices. The stty-cxma equivalents are shown in brackets.

Transparent Print ON String (**stty–cxma onstr** ***s***)
: Sets the terminal escape sequence to turn transparent printing on. The strings can be composed of standard ASCII printing and non-printing characters. Control (non-printing) characters must be entered by their octal values and must consist of three digits preceded by a back-slash character. For example, the Escape character, 33 octal, should be entered as \033. Thus, if transparent printing is turned on by the string <Esc>[5i, the parameter should be entered as \033[5i. The default is \033[5i (ANSI standard).

Transparent Print OFF String (**stty–cxma offstr** ***s***)
: Sets the terminal escape sequence to turn transparent printing off. Refer to Transparent Print ON String for the string format. The default is \033[4i (ANSI standard).

Transparent Print Maximum Characters per Second (**stty–cxma maxcps** ***n***)
: Sets the maximum characters per second (CPS) rate that characters are transmitted to the transparent print device. The rate chosen should be just below the average print speed. If the number is too low, printer speed will be reduced. If the number is too high, the printer uses flow control, and user entry time is reduced. The default is 100 CPS.

Transparent Print Maximum Character Packet Size (**stty–cxma maxchar** ***n***)
: Sets the maximum number of transparent print characters the driver will place in the output queue. Reducing this number increases system overhead; increasing this number delays operator keystroke echo times when the transparent printer is in use. The default is 50 characters.

Transparent Print Printer Buffer Size (**stty–cxma bufsize** ***n***)
: Sets the driver's estimate of the size of the transparent printer's input buffer. After a period of inactivity, the driver bursts the appropriate number of characters to the transparent printer before reducing to the **maxcps** rate. The default is 100 characters.

Once the parameters are set, any data directed to the transparent printer device is automatically wrapped in the appropriate codes to identify it to the terminal as printer data. For example:

```
cat filename > /dev/xtty1
```

Setting Terminal Options with stty-cxma

stty-cxma is a utility program that sets and displays the terminal options for the Micro Channel 128-port, ISA 8-port, and ISA 128-port adapters and is located in `/usr/lbin/tty` directory. The format is:

```
stty-cxma [-a] [option(s)] [ttyname]
```

With no options, **stty-cxma** displays all special driver settings, modem signals, and all standard parameters displayed by stty(1) for the tty device referenced by standard input. Command options are provided to change flow control settings, set transparent print options, force modem control lines, and display all tty settings. Any unrecognized options are passed to stty(1) for interpretation. The options are:

–a Displays all of the unique adapter option settings, as well as all of the standard tty settings reported by the **stty –a** command.

ttyname Sets and displays options for the given tty device, instead of standard input. This form can be used with a tty pathname prefixed by `/dev/` or with a simple tty name beginning with tty. This option may be used on a modem control line when no carrier is present.

The following options specify transient actions to be performed immediately:

break Sends a 250 MS break signal out on the tty line.

flush Indicated an immediate flush (discard) of tty input and output.

flushin Flushes tty input only.

flushout Flushes tty output only.

The following options specify actions that are reset when the device is closed. The device will use the default values the next time it is opened.

stopout Stops output exactly as if an XOFF character was received.

startout Restarts stopped output exactly as if an XON character was received.

stopin Activates flow control to stop input.

startin Releases the flow control to resume stopped input.

[–]dtr [drop] Raises the DTR modem control line, unless DTR hardware flow control is selected.

[–]rts [drop] Raises the RTS modem control line, unless RTS hardware flow control is selected.

The following options remain in effect until the system is rebooted or until the options are changed.

[–]fastcook Performs cooked output processing on the intelligent card to reduce host CPU usage, and increase raw mode input performance.

[–]fastbaud Alters the baud rate tables, so 50 baud becomes 57,600 baud, 75 baud becomes 76,800 baud and 110 baud becomes 115,200 baud.

[–]rtspace Enables / disables RTS hardware input flow control, so RTS drops to pause remote transmission.

[–]ctspace Enables / disables CTS hardware output flow control, so local transmission pauses when CTS drops.

[–]dsrpace Enables / disables DSR hardware output flow control, so local transmission pauses when DSR drops.

[–]dcdpace Enables / disables DCD hardware output flow control, so local transmission pauses when DCD drops.

[–]dtrpace Enables / disables DTR hardware input flow control, so DTR drops to pause remote transmission.

[–]forcedcd Disable [re-enable] carrier sense, so the tty may be opened and used even when carrier is not present.

[–]altpin Maps the RJ-45 connector pinouts to the default 10-pin connector values or the 8-pin connector values. When this paramter is **enabled**, the location of DSR and DCD is switched oso that DCD is avaialble when using an 8-pin RJ-45 connector instead of the 10-pin RJ-45 connector. (Default=**disable**.)

Possible values:

enable (specifies 8-pin connector values)

disable (specifies 10-pin connector values)

startc *c* Sets the XON flow control character. The character may be given as a decimal, octal, or hexadecimal number. Octal numbers are recognized by the presence of a leading zero, and hexadecimal numbers are denoted by a leading 0x. For example, the standard XON character, CTRL-Q, can be entered as 17 (decimal), 021 (octal), or 0x11 (hexadecimal).

stopc *c* Sets the XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see startc for format of octal and hexadecimal numbers).

astartc *c* Sets auxiliary XON flow control character. The character may be given as a decimal, octal, or hexadecimal number (see startc for format of octal and hexadecimal numbers).

astopc *c* Sets auxiliary XOFF flow control character. The character may be given as a decimal, octal, or hexadecimal number (see startc for format of octal and hexadecimal numbers).

[–]aixon Enables auxiliary flow control, so that two unique characters are used for XON and XOFF. If both XOFF characters are received, transmission will not resume until both XON characters are received.

[–]2200flow Uses 2200 style flow control on the port. The 2200 terminals support an attached printer and use four flow control characters: terminal XON (0xF8), printer XON (0xF9), terminal XOFF (0xFA) and printer XOFF (0xFB).

[–]2200print Determines how these flow control characters are interpreted. If 2200print is set, run independent flow control for terminal and transparent print devices. Otherwise, terminal and printer flow control are logically tied together. If either XOFF character is received, all output is paused until the matching XON character is received.

maxcps *n* Sets the maximum characters per second (CPS) rate that characters are output to the transparent print device. The rate chosen should be just below the average print speed. If the number is too low, printer speed will be reduced. If the number is too high, the printer uses flow control, and user entry time is reduced. The default is 100 CPS.

maxchar *n* Sets the maximum number of transparent print characters the driver places in the output queue. Reducing this number increases system overhead; increasing this number delays operator keystroke echo times when the transparent printer is *n* use. The default is 50 characters.

bufsize *n* Sets the driver's estimate of the size of the transparent printer's input buffer. After a period of inactivity, the driver bursts this many characters to the transparent printer before reducing to the maxcps rate. The default is 100 characters.

onstr *s* Sets the terminal escape sequence to turn transparent printing on. The strings can be composed of standard ASCII printing and non-printing characters. Control (non-printing) characters must be entered by their octal values, and must consist of three digits preceded by a back-slash character. For example, the Escape character, 33 octal, should be entered as \033. If transparent printing is turned on by the string <Esc>[5i (ANSI standard), it would be entered as: \033[5i.

offstr *s* Sets the terminal escape sequence to turn transparent printing off. Refer to **onstr *s*** for the format of the strings.

term *t* Sets the transparent printer on/off strings to values found in the internal default table. Internal defaults are used for the following terminals: adm31, ansi, dg200, dg210, hz1500, mc5, microterm, multiterm, pcterm, tvi, vp-a2, vp-60, vt52, vt100, vt220, wyse30, wyse50, wyse60, or wyse75. If the terminal type is not found in the internal default table, ditty reads the terminfo entry for the terminal type and sets transparent print on/off strings to values given by the mc5/mc4 attributes found there.

Chapter 4. Standard, 8–Port Micro Channel, and 16–Port Asynchronous Adapters

This chapter describes the standard, 8–port, and 16–port asynchronous adapters.

Topics discussed are:

Standard I/O Ports

Most system unit models have two integrated (standard) EIA 232 asynchronous serial ports. The model 105 features a single integrated asynchronous serial port that can be converted to support two serial devices using an optional fanout cable. EIA 232 asynchronous serial devices can be attached directly to the standard serial ports using standard serial cables with 25–pin D–shell connectors. The machines capable of multiprocessing have three serial ports.

Configuring an EIA 232 Asynchronous Terminal Device

This procedure allows you to define and configure a tty device connected to a standard serial port, an 8–port, or 16–port asynchronous adapter.

Procedure

1. Use the **smit mktty** fast path to access the **Add a TTY** menu.
2. Select **Add a TTY**.
3. Select **tty rs232 Asynchronous Terminal**.
4. Make a selection from the available standard I/O, 8–port, or 16–port adapters displayed on the screen. If no adapters are displayed or if they are in a defined state, check the configuration, cabling, and setup again.
5. In the displayed dialog fields, you can add or change the tty attributes.
6. When you have finished, select **Do**.

Configuring an EIA 232 Asynchronous Printer/Plotter Device

This procedure allows you to define and configure a printer/plotter device connected to a standard serial port, an 8–port asynchronous adapter, or a 16–port asynchronous adapter.

Procedure

1. To create a printer/plotter device on an asynchronous adapter, use the **smit pdp** fast path to access the **Printer/Plotter Devices** menu.
2. Select **Add a Printer/Plotter**.
3. Make a selection from the list of printer and plotter types shown on the screen, and press Enter. For this example, the following selection was made:

   ```
   osp Other serial printer
   ```
4. Select the **rs232** option.
5. Make a selection from the available 8–port controllers on the screen. If no controllers are displayed or if they are shown in a defined state, check the configuration, cabling, and setup again.
6. In the displayed dialog fields, you can add or change the printer/plotter device attributes.
7. When you have finished, select **Do**.

Moving a TTY to Another Port

This procedure allows you to move a tty device to another port and retain all the characteristics of the device. Use the System Manager Interface Tool (SMIT) to build the **chdev** command or issue the **chdev** command directly from the command line.

Prerequisites

1. A tty device must be defined to the system.
2. The tty must be disabled. Refer to the **pdisable** command.

Move a TTY to Another Port Using SMIT

1. Use the **smit movtty** fast path to access the **TTY** screen.
2. Select the tty device type to be moved.
3. Select the adapter device to which the tty is or will be attached.
4. At the PORT number field, use the **List** option to display a list of available ports. Select the port to which the tty device is to be moved.
5. Select **Do** to move the device.

Move a TTY to Another Port from the Command Line

Enter the **chdev** command at the command line, specifying the following flags:

–l *Name*	Identifies the tty device to be moved.
–p *ParentName*	Identifies the parent device.
–w *ConnectionLocation*	Identifies the port number to which the tty device is to be moved.

The following example moves a defined tty device `tty11` to port `0` on serial adapter `sa5`:

```
chdev –l tty11 –p sa5 –w 0
```

See the **chdev** command for a detailed description.

Removing a TTY

This procedure allows you to make a tty device unavailable and either keep its definition in the customized database or remove it from the database. To make a device unavailable is to unconfigure it. A tty device that is made unavailable but that has not been deleted from the database can be made available again with the same attributes. If the definition of the tty device has been deleted, the device cannot be made available again.

Prerequisites

1. A tty device must be defined to the system.
2. The tty must be disabled. Refer to the **pdisable** command.

Remove TTY Using SMIT

1. Use the **smit rmtty** fast path to access the **TTY** screen.
2. Select the tty device you want to make unavailable or remove.
3. Select **yes** to keep the definition of the tty in the database or **no** to remove the definition.
4. Select **Do** to remove the tty device.

Remove a TTY from the Command Line

Enter the **rmdev** command at the command line as follows:

- To make the logical device `tty0` unavailable but keep its definition in the customized database, enter:

  ```
  rmdev -l tty0
  ```

- To make the logical device `tty0` unavailable and delete its definition from the customized database, enter:

  ```
  rmdev -l tty0 -d
  ```

8–Port Micro Channel Asynchronous Adapters

The family of asynchronous adapters is based on a common functional design. The individual adapter characteristics, however, are determined by the supported device interfaces. The family consists of three adapters:

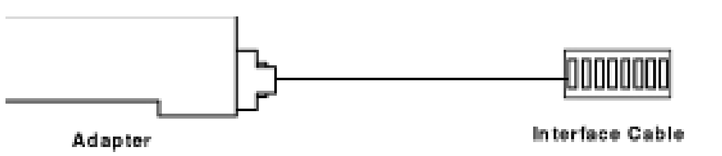
- 8–Port Asynchronous Adapter – EIA 232, on page 4-5
- 8–Port Asynchronous Adapter – MIL–STD–188, on page 4-9
- 8–Port Asynchronous Adapter – EIA 422A, on page 4-10

The family of 8–port adapters is based on the dual universal asynchronous receiver and transmitter (DUART) chip providing two serial communications channels.

The following sections contain detailed information about 8–port adapters:

8–Port Asynchronous Adapter – EIA 232 Description

The 8–Port asynchronous adapter, EIA 232, (see figure) provides support for attaching a maximum of eight EIA 232D asynchronous serial devices (such as modems, terminals, plotters, and printers) to a system unit. The system must be based on a Micro Channel bus or an ISA bus and support up to eight 8–port adapters.



8-Port Asynchronous Adapter

This adapter is fully programmable and supports asynchronous communications only. It can also add and remove start and stop bits and supports even, odd, or no parity on serial data. A programmable baud rate generator allows operation from 50 to 38,400 bps for the Micro Channel bus and 50 to 115,200 bps for the ISA bus. The adapters support 5–, 6–, 7–, or 8–bit characters with 1, 1.5, or 2 stop bits. A priority interrupt system controls transmit, receive, error, line status, and data set interrupts.

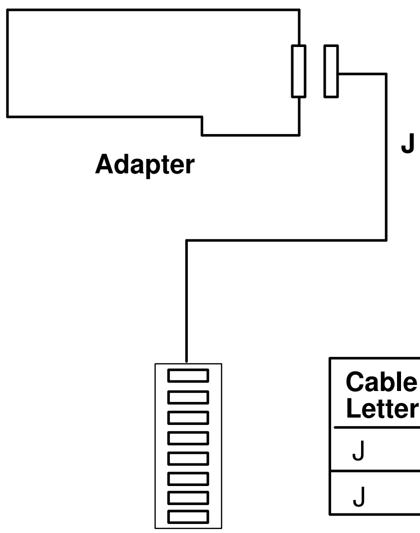
Installing the 8–Port Asynchronous Adapter

The 8–port asynchronous adapter fits into a single Micro Channel slot in AIX. To install the adapter use the following steps:

1. Verify that all users are logged off the system and run the following command:

   ```
   shutdown –F
   ```

2. When the **shutdown** command completes, turn the system power switch to the off position.
3. Open the AIX case and insert the 8–port asynchronous adapter into a free Micro Channel slot.
4. Attach the 78–pin D–shell connector from the 8–port interface cable to the 8–port adapter as shown in the following figure.



Cable Letter	Cable Name/Description	Part Number	Feature Code	Length m	ft.
J	Part of cable assembly	00F5524	2995	3	10
J	Part of cable assembly	11H5960	2931	3	10

8-Port Interface to Adapter Connection

5. Put the cover panels back on the system unit.
6. Push the system power switch to the on position.

   The system will recognize and configure the 8–port adapter during the boot process.

7. After the boot completes, log in using the root user ID boot process.

   ```
   lsdev –Cc adapter | pg
   ```

Only those adapters that are in an available state are ready for use by the system.

If the newly installed adapter is *not* available, then verify:

- The adapter is installed correctly into the Micro Channel slot.
- All necessary cabling is attached and fitted tightly into place.
- Run the command: **errpt –a | pg** and examine the system error report for problems relating to the adapters.
- Run the command: **cfgmgr –v | pg**. This command will attempt to reconfigure the adapter without rebooting. Observe the paged output for errors.

If running **cfgmgr** fails, a reboot will be necessary.

8–Port Asynchronous Adapter Hardware Information

The system interface presents a 3–bit address and 8–bit data as well as control lines to the DUART chip. Data from the system interface is serialized for transmission to an external device. The serial data can include a parity bit at the byte boundary. Conversely, data from an external device is deserialized for transmission to the system interface. This data may also include a parity bit, which can be optionally checked. As an option, the channel can operate in first–in–first–out (FIFO) mode.

In FIFO mode, up to 16 bytes can be buffered in both the transmitter and receiver. The serial interface uses start–stop protocol for both data transmission and reception. That is, each byte (plus the parity bit) is framed by one or more start bits and stop bits, which allows synchronization on an individual character (byte) basis.

The DUART chip uses a 12.288 MHz oscillator to generate its internal timing to drive the transmitter and receiver logic. The channel supports full duplex operation. Four DUART chips are implemented on each 8–port adapter.

Thirteen system–accessible registers are available. Programmable features on each channel include:

- Character length: 5, 6, 7, or 8 bits
- Parity generation/detection: Even, odd, or none
- Number of stop bits: 1, 1.5, or 2
- Enable/disable interrupts. Received data available
- Transmitter holding register empty
- Line status
- Overrun error
- Parity error
- Framing error
- Break.

The following table is a summary of port (device interface) characteristics for the adapters.

Parameter	**EIA 232**	**MIL–STD 188**	**EIA 422A**
Topology	Point to Point	Point to Point	Point to Point
Maximum data rate	138.4Kbps (MC)/115.2 (ISA)	138.4Kbps	138.4Kbps
Transmission media	Multiconductor	Multiconductor	Multiconductor
Number of cable wires	9 including signal ground	9 including signal ground	5 including signal ground
Maximum cable length	61 m (200 feet)	130 m at 38.4Kbps	1200 m < 90Kbps
Device connector	25–pin D	25–pin D	25–pin D
Electrical interface	Unbalanced	Unbalanced	Balanced
Bit encoding	Digital bi–level	Digital bi–level	Digital bi–level

The interrupt arbitration logic sets priority for adapters according to the following scheme:

```
Adapter   Priority
    1     Highest
    2        |
    3        |
    4        |
    5        |
    6        |
    7        |
    8     Lowest
```

Communications Channel Priority

The DUART channels with pending interrupts are serviced according to a fixed–priority scheme. The highest priority is assigned to port 0. Next in priority is port 1, and so forth. The lowest priority is port 7.

8–Port Asynchronous Adapters Interrupt Logic Description

The interrupt logic is divided into two sections:

- Interrupt generation logic
- Interrupt arbitration logic

Both logic sections are implemented on every 8–port adapter. The interrupt generation logic provides the interface to the system. This logic generates the system interrupt requests and contains the interrupt–sharing circuitry.

The function of the interrupt arbitration logic is to identify the 8–port adapter with the highest priority interrupt pending. The logic then places the interrupt information for the highest priority port in the Interrupt Arbitration register. This is accomplished in one read operation.

The interrupt arbitration logic is unique to the 8–port adapter and should not be confused with the Micro Channel arbitration logic.

Interrupt Generation Logic

The adapter implements the following eight system interrupt request lines:

- IRQ 3
- IRQ 5
- IRQ 9
- IRQ 10
- IRQ 11
- IRQ 12
- IRQ 14
- IRQ 15

Only one request line is active during normal operation. All 8–port adapters in one system should use the same interrupt level for optimal system performance. The active line is selected by writing to the appropriate POS register during the setup cycle. The adapter supports interrupt sharing and implements an open collector configuration. In this arrangement, the interrupt line is pulled high by a system pull–up resistor. The adapter pulls the line low to indicate an active interrupt request.

Interrupt Arbitration Logic

Up to eight 8–port adapters can co–reside and concurrently operate in a system. The interrupt arbitration logic determines the priority for software service when two or more 8–port or 16–port adapters generate interrupts. This logic provides the system with adapter and port identification as well as the interrupt type in a single read operation. Once an interrupt request is detected, the system reads the 16–bit interrupt arbitration register, which is located at I/O address 0130.

8–Port Asynchronous Adapters MIL–STD 188 Interface Signals

The following interface signals are implemented on each port of the adapter:

Signal	Definition
Tx Data	Transmit Data
RTS	Request To Send
CTS	Clear To Send
DSR	Data Set Ready
Rx Data	Receive Data
DCD	Data Carrier Detect
DTR	Data Terminal Ready
RI	Ring Indicator
Sig Gnd	Signal Ground

MIL–STD 188 Signal Voltage Levels

Voltage levels for the MIL–STD 188 Adapter are explained in the following sections:

- Normal Mark and Space Polarity
- Mark and Space Polarity Inversion.

Normal Mark and Space Polarity

The signal is in the mark state when the voltage on the interchange circuit, measured at the interface point, is less than –4 V dc with respect to the signal ground. The signal is in the space state when the voltage is greater than +4 V dc with respect to the signal ground. The region between +4 V dc and –4 V dc is defined as the transition region and is not a valid level. The voltage that is less than –6 V dc or greater than +6 V dc is also not a valid level.

During the transmission of data, the mark state denotes binary 1 and the space state denotes binary 0.

For interface control circuits, the function is ”on” when the voltage is greater than +4 V dc with respect to the signal ground and is ”off” when the voltage is less than –4 V dc with respect to the signal ground. MIL–STD 188 signal levels are shown in the following table:

Interchange Voltage	Binary State	Signal Condition	Interface Control Function
+ Voltage	0	Space	On
– Voltage	1	Mark	Off

Mark and Space Polarity Inversion

Military standard MIL–STD 188 requires that adapters provide the capability to optionally invert the polarities of the mark and space states of the transmit and receive lines. The capability is provided independently on each port.

The DUART modem control register bit 3 (Out 2) is used for this purpose. When bit 3 is set to a value of 1, the polarities for the mark and space states are set to the normal state. When bit 3 is set to a value of 0, the polarities for the mark and space states are inverted.

The signal is in the *space state* when the voltage is less than –4 V dc with respect to the signal ground. The signal is in the mark state when the voltage is greater than +4 V dc with respect to the signal ground.

The region between +4 V dc and –4 V dc is defined as the *transition region* and is not a valid level. The voltage that is less than –6 V dc or greater than +6 V dc is also not a valid level.

Standards Compliance

The electrical characteristics of the 8–Port asynchronous MIL–STD 188 adapter ports conform to those sections of MIL–STD 188–114 that address an unbalanced voltage interface. The standard is dated March 24, 1976.

The adapter ports meet the functional requirements for asynchronous operation (start–stop protocol) as described in the EIA Standard 232C dated October 1969 and in the EIA Standard 232D dated January 1987.

8–Port Asynchronous Adapters EIA 422A Interface Signals

The following EIA 422A interface signals are implemented on each port of the adapter:

Signal	Definition
TxA	Transmit Data
TxB	Transmit Data
RxA	Receive Data
RxB	Receive Data
Sig Gnd	Signal Ground

EIA 422A Signal Voltage Levels

The line driver produces a differential voltage in the range of 2 to 6 volts (measured at the generator interface point). The magnitude of the differential voltage at the receiver must be in the range of 200 millivolts to 6 volts (measured at the load interface point).

Measurements are taken at terminal A (positive lead) with respect to terminal B (negative lead). The following table describes the signal states with respect to voltage levels:

Interchange Voltage	**Binary State**	**Signal Condition**
+ Voltage	0	Space
– Voltage	1	Mark

Surge Protection Circuitry

The 8–Port asynchronous EIA 422A adapter supports indoor cabling up to 1200 m (4000 ft) in length. Cables of such lengths are susceptible to sudden voltage surges due to induced voltages such as indirect lightning strikes. Secondary surge protection circuitry is implemented on the EIA 422A adapter to protect it from these voltage surges. The surge protection circuitry is implemented on the adapter interface data lines.

Fail–Safe Circuitry

Fail–safe circuitry has been added to the input leads of each EIA 422A receiver to prevent fault conditions when the receiver is not connected to a driver (open cable). The fail–safe circuitry sets the receiver to the mark state (binary 1) whenever the receiver is not connected to a driver.

Standards Compliance

The electrical characteristics of the 8–Port asynchronous EIA 422A adapter ports comply with the EIA Standard 422A dated December 1978.

8–Port Asynchronous Adapters EIA 232 Interface Signals

The following interface signals are implemented on each port of the adapter:

Signal	Definition
TxD	Transmit Data
RTS	Request To Send
CTS	Clear To Send
DSR	Data Set Ready
RxD	Receive Data
DCD	Data Carrier Detect
DTR	Data Terminal Ready
RI	Ring Indicator
Sig Gnd	Signal Ground

EIA 232 Signal Voltage Levels

The signal is in the mark state when the voltage on the interchange circuit, measured at the interface point, is less than –3 V dc with respect to the signal ground. The signal is in the space state when the voltage is greater than +3 V dc with respect to the signal ground. The region between +3 V dc and –3 V dc is defined as the *transition region* and is not a valid level. Voltage less than –15 V dc or greater than +15 V dc is also not a valid level.

During the transmission of data, the mark state denotes binary state 1 and the space state denotes binary state 0.

For interface control circuits, the function is on when the voltage is greater than +3 V dc with respect to the signal ground and is off when the voltage is less than –3 V dc with respect to the signal ground. See the following table for EIA 232 signal levels:

Interchange Voltage	Binary State	Signal Condition	Interface Control Function
+ Voltage	0	Space	On
– Voltage	1	Mark	Off

Standards Compliance

The electrical characteristics of the 8–Port asynchronous EIA 232 adapter ports conform to the EIA Standard 232C dated October 1969 and to the EIA Standard 232D dated January 1987.

The adapter ports meet the functional requirements for asynchronous operation (start–stop protocol) as described in the EIA Standard 232C dated October 1969 and in the EIA Standard 232D dated January 1987.

8–Port Asynchronous Adapters Control Logic

The PAL–based control logic section coordinates the activities of all major adapter functions and is clocked with a 40 MHz square–wave generator. It interfaces with the Micro Channel, and its functions include decoding addresses, checking address parity, responding with the proper I/O control signals, and driving the selected interrupt request (IRQ) line (one of eight IRQ lines).

The control logic interfaces with the other adapter logic blocks and in this capacity provides the control lines to the communication channels (DUART) and the interrupt arbitration logic. The control logic also interfaces with the data bus driver logic and provides control for the direction of data flow and for the selection data bytes, which are placed onto the local bus. It controls the data parity generator, parity checker, and latches.

16–Port Asynchronous Adapters

The family of adapters is based on a common functional design. The individual adapter characteristics, however, are determined by the supported device interfaces. The family consists of two adapters, detailed in the following sections:

- 16–Port Asynchronous Adapter – EIA 422A, on page 4-12
- 16–Port Asynchronous Adapter – EIA 232, on page 4-16.

The family of 16–port adapters is based on the dual universal asynchronous receiver and transmitter (DUART) chip, which provides two serial communications channels. More information on the DUART chip and its functionality can be found in the ”16–Port Asynchronous Adapter Hardware Information”, on page 4-14.

The following sections contain detailed information about 16–port adapters:

16–Port Asynchronous Adapter – EIA 422A Description

The 16–Port asynchronous adapter – EIA 232 provides support for attaching a maximum of 16 EIA 232 asynchronous serial devices (printers and terminals) to a system unit. Up to eight adapters (any combination within the family) can be used in a single system unit.

This adapter is fully programmable and supports asynchronous communications only. It adds and removes start bits and stop bits. The adapters support even, odd, or no parity on serial data. A programmable baud–rate generator allows operation from 50 to 38400 bps. The adapters support 5–, 6–, 7–, or 8–bit characters with 1, 1.5, or 2 stop bits. A priority interrupt system controls transmit, receive, error, line status, and data set interrupts. The 16 connectors for device attachment are provided in a 16–port cable assembly, EIA 422A, as shown in the following figure:

16-port Asynchronous Adapter

Characteristics

- Standard Micro Channel form factor card.
- Data rates up to 38.4K bps per port.
- 16 byte buffering on transmit and receive.
- Single 78–pin output connector (Multiport interface cable attaches to this connector).
- Surge protection circuitry.
- Supports cabling up to 1200 m (4000 ft).

- Supports the TxD and RxD interface signals.
- 8–bit/16–bit Micro Channel slave interface.

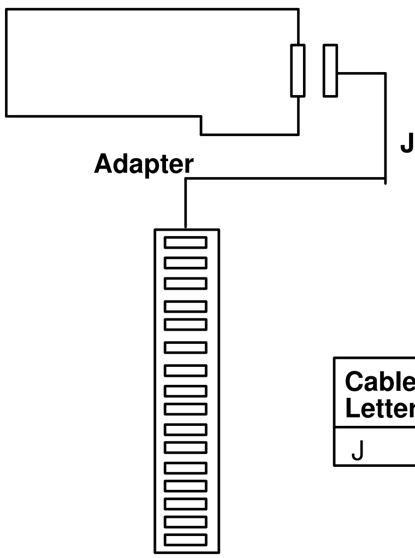
Installing the 16–Port Asynchronous Adapter

The 16–port asynchronous adapter fits into a single Micro Channel slot in the server. To install the adapter, use the following steps:

1. Verify that all users are logged off the system and run the following command:

   ```
   shutdown –F
   ```

2. When the **shutdown** command completes, turn the system power switch to the ”off” position.
3. Open the server case and insert the 16–port asynchronous adapter into a free Micro Channel slot.
4. Attach the 78–pin D–shell connector from the 16–port interface cable to the 16–port adapter (see figure).



Cable Letter	Cable Name/Description	Part Number	Feature Code	Length m	ft.
J	Part of cable assembly	00F5524	2995	3	10

16-port Interface to Adapter Connection

5. Put cover panels back on the system unit
6. Push the system power switch to the ”on” position.

   The system will recognize and configure the 16–port adapter during the boot process.

After the boot completes, login using the root user ID and issue the following command to check the adapter availability:

```
lsdev –Cc adapter | pg
```

Only those adapters in the available state are ready for use by the system.

If the newly installed adapter is NOT available, then verify the following:

1. The adapter is installed correctly into the Micro Channel slot.
2. All necessary cabling is attached and fitted tightly into place.
3. Run the command: **errpt –a | pg** and examine the system error report for problems relating to the adapters.
4. Run the command: **cfgmgr –v | pg**. This command will attempt to reconfigure the adapter without rebooting. Watch the paged output for errors.

5. If running **cfgmgr** fails, a reboot will be necessary.

16–Port Asynchronous Adapter Hardware Information

The system interface presents a 3–bit address and 8–bit data as well as control lines to the chip. Data from the system interface is serialized for transmission to an external device. The serial data may include a parity bit at the byte boundary. Conversely, data from an external device is deserialized for transmission to the system interface. This data may also include a parity bit, which can be optionally checked. As an option, the channel can operate in first–in–first–out (FIFO) mode.

In FIFO mode, up to 16 bytes can be buffered in both the transmitter and receiver. The serial interface uses start–stop protocol for both data transmission and reception. That is, each byte (plus parity bit) is framed by a start bit and stop bit, which allows synchronization on an individual character (byte) basis.

The DUART chip uses a 12.288 MHz oscillator to generate its internal timing to drive the transmitter and receiver logic. The channel supports full duplex operation. Eight DUART chips are implemented on each 16–port adapter.

Thirteen system–accessible registers are available. Programmable features on each channel include:

- Character length: 5,6, 7, or 8 bits
- Parity generation/detection: even, odd, or none
- Number of stop bits: 1, 1.5, or 2
- Enable/disable interrupts. Received data available
- Transmitter holding register empty
- Line status
- Overrun error
- Parity error
- Framing error
- Break.

The following table is a summary of port (device interface) characteristics for the adapters.

Parameter	**EIA 232**	**EIA 422A**
Topology	Point to Point	Point to Point
Maximum data rate (standard)	20Kbps	2Mbps
Maximum data rate (board)	38.4Kbps	38.4Kbps
Transmission media	Multiconductor	Multiconductor
Number of cable wires	5 including signal ground	5 including signal ground
Maximum cable length	61 m (200 ft)	1200 m < 90Kbps
Device connector	25–pin D	25–pin D
Electrical interface	Unbalanced	Balanced
Bit encoding	Digital bi–level	Digital bi–level

16–Port Asynchronous Adapters Adapter Board Priority

The interrupt arbitration logic sets priority for adapters according to the following scheme:

```
Adapter   Priority
    0     Highest
    1       |
    2       |
    3       |
    4       |
    5       |
    6       |
    7       |
    8       |
    9       |
    10      |
    11      |
    12      |
    13      |
    14      |
    15    Lowest
```

Communications Channel Priority

The DUART channels with pending interrupts are serviced according to a fixed–priority scheme. The highest priority is assigned to port 0. Next in priority is port 1, and so forth. The lowest priority is port 15.

16–Port Asynchronous Adapters Interrupt Logic Description

The interrupt logic is divided into two sections:

- Interrupt generation logic
- Interrupt arbitration logic.

Both logic sections are implemented on every 16–port adapter. The interrupt generation logic provides the interface to the system. This logic generates the system interrupt requests and contains the interrupt–sharing circuitry.

The function of the interrupt arbitration logic is to identify the 16–port adapter with the highest priority interrupt pending. The logic then places the interrupt information of the highest priority port in the interrupt arbitration register. This is accomplished in one read operation.

The interrupt arbitration logic is unique to the 16–port adapters and should not be confused with the Micro Channel arbitration logic.

Interrupt Generation Logic

The adapter implements the following eight system interrupt request (IRQ) lines:

- IRQ 3
- IRQ 5
- IRQ 9
- IRQ 10
- IRQ 11
- IRQ 12
- IRQ 14
- IRQ 15

Only one request line is active during normal operation. All 16–port adapters in one system should use the same interrupt level for optimal system performance. The active line is selected by writing to the appropriate POS register during the setup cycle. The adapter supports interrupt sharing and implements an open collector configuration as defined in the Micro Channel architecture. In this arrangement, the interrupt line is pulled high by a system pull–up resistor. The adapter pulls the line low to indicate an active interrupt request.

Interrupt Arbitration Logic

Up to eight 8–port or 16–port adapters can co–reside and concurrently operate in a system. The interrupt arbitration logic determines the priority for software service when two or more 8–port or 16–port adapters generate interrupts. This logic provides the system with adapter and port identification as well as the interrupt type in a single read operation. Upon detection of an interrupt request, the system reads the 16–bit interrupt arbitration register located at I/O address 0130.

16–Port Asynchronous Adapters EIA 232 Interface Signals

The following interface signals are implemented on each port of the adapter:

Signal	Definition
TxD	Transmit data
DCD	Data carrier detect
DTR	Data terminal ready
RxD	Receive data
Sig Gnd	Signal ground

EIA 232 Signal Voltage Levels

The signal is in the mark state when the voltage on the interchange circuit, measured at the interface point, is less than –3 V dc with respect to the signal ground. The signal is in the space state when the voltage is greater than +3 V dc with respect to the signal ground. The region between +3 V dc and –3 V dc is defined as the transition region and is not a valid level. Voltage less than –15 V dc or greater than +15 V dc is also not a valid level.

During the transmission of data, the mark state denotes binary state 1 and the space state denotes binary state 0.

For interface control circuits, the function is on when the voltage is greater than +3 V dc with respect to the signal ground and is off when the voltage is less than –3 V dc with respect to the signal ground. See the following table for EIA 232 signal levels.

Interchange Voltage	**Binary State**	**Signal Condition**	**Interface Control Function**
+ Voltage	0	Space	On
– Voltage	1	Mark	Off

Standards Compliance

The electrical characteristics of the 16–Port asynchronous EIA 232 adapter ports conform to the EIA Standard 232C dated October 1969 and to the EIA Standard 232D dated January 1987.

The adapter ports meet the functional requirements for asynchronous operation (start–stop protocol) as described in the EIA Standard 232C dated October 1969 and in the EIA Standard 232D dated January 1987.

16–Port Asynchronous Adapters EIA 422A Interface Signals

The following EIA 422A interface signals are implemented on each port of the adapter:

Signal	Definition
TxA	Transmit Data
TxB	Transmit Data
RxA	Receive Data
RxB	Receive Data
Sig Gnd	Signal Ground

EIA 422A Signal Voltage Levels

The line driver produces a differential voltage in the range of 2 to 6 volts (measured at the generator interface point). The magnitude of the differential voltage at the receiver must be in the range of 200 millivolts to 6 volts (measured at the load interface point).

Measurements are taken at terminal A (positive lead) with respect to terminal B (negative lead). The following table describes the signal states with respect to voltage levels:

Interchange Voltage	Binary State	Signal Condition
+ Voltage	0	Space
– Voltage	1	Mark

Surge Protection Circuitry

The 16–Port Asynchronous EIA 422A adapter supports indoor cabling up to 1200 m (4000 ft) in length. Cables of such lengths are susceptible to sudden voltage surges due to induced voltages such as indirect lightning strikes. Secondary surge protection circuitry is implemented on the EIA 422A adapter to protect it from these voltage surges. The surge protection circuitry is implemented on the adapter interface data lines.

Fail–Safe Circuitry

Fail–safe circuitry has been added to the input leads of each EIA 422A receiver to prevent fault conditions when the receiver is not connected to a driver (open cable). The fail–safe circuitry sets the receiver to the mark state (binary 1) whenever the receiver is not connected to a driver.

Standards Compliance

The electrical characteristics of the 16–Port asynchronous EIA 422A adapter ports comply with the EIA Standard 422A dated December 1978.

Chapter 5. 64–Port Asynchronous Adapters

The following sections contain information on planning, configuring, and managing the 64–port asynchronous adapter. The 64–port adapter is only supported on Micro Channel uniprocessor machines and has been withdrawn as a feature for AIX Versions 4.3 and later.

64–Port Asynchronous Adapter Subsystem

The 64–Port asynchronous controller subsystem provides attachment for 64 asynchronous communication ports from a single I/O slot in the system unit. The subsystem has connectors to attach up to four external 16–port asynchronous concentrators thus giving a total of 64 asynchronous ports.

A single concentrator provides 16 connections for EIA 232 asynchronous devices. Although application dependent, data rates up to 38.4Kbps per port are supported.

A 7.6 m (25 ft) controller attachment cable is included with each concentrator. Cabling distances of up to 762 m (2500 ft) between controller and concentrator are supported but must be supplied by the customer. EIA 232 cabling of up to 61 m (200 ft) between concentrator and device is also supported.

The 16–Port asynchronous concentrator uses RJ–45 connectors for device attachment. The RJ–45 to DB25 converter cable plugs in to the RJ–45 connector and provides a 25–pin D–shell connector for device attachment.

The 64–port asynchronous adapter hardware is no longer available as a supported product. However, if you purchased one of the models (with a 64–port adapter) specified in the following chart before November 30, 1993, the adapter is still supported.

Supported Models					
100	110	125W		110	
400	430	440	450	460	470
600	610	620	650	660	
800	820	830			

If you upgraded from the models previously specified to one of the models in the following chart, your 64–port adapter is still supported.

Supported Models (as Upgrades)			
400	480	490	490H
600	680	690	690H

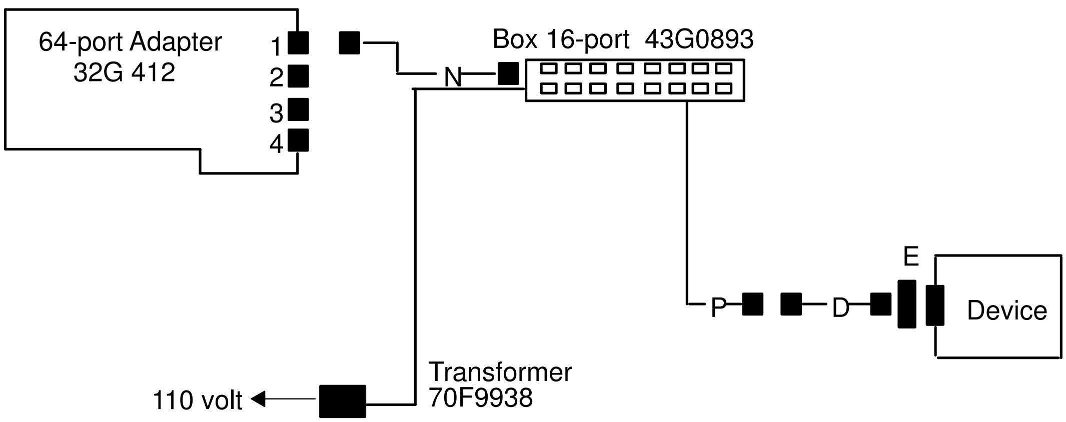
Characteristics

- Standard Micro Channel form factor card.
- Maximum of 38.4Kbps per port (dependent on application and use).
- Up to 2500 feet between controller and concentrator.
- Up to 200 feet from concentrator to device.
- Surge protection circuitry.
- Supports the following interface signals on each port:

  `TxD, RxD, DTR, CTS, RTS, and DCD`
- 8–bit/16–bit Micro Channel slave interface.

Topology

The following figure illustrates the 64–port asynchronous controller with one of four concentrator boxes that can be attached to any one of the controller connectors. The position of the adapter determines the controller connector numbering:



Cables	PN	FC	Length m	ft
D	6323741	#2936	3	10
E	58f2861	#2937	N/A	N/A
N	00G1109	N/A	7.6	25
P	53F3432	#6402	.45	1.5

- Top to bottom 1 through 4, when the card is in the vertical position
- Right to left 1 through 4, when the card is in the horizontal position.

A choice of cables can be attached to any of 16 concentrator box ports. The ports on the concentrator box are labeled 0 through 15 and accept RJ–45 connectors. In order to make the necessary connections to this adapter, the system administrator needs to know the devices and persons assigned to each port.

Listing All Defined 64–Port Asynchronous Adapters

This procedure provides a list of the 64–port asynchronous adapters that are defined in the customized database. A defined device can be either available (configured) or unavailable to the operating system. To obtain a list of defined 64–port adapters, use the System Management Interface Tool (SMIT) to build the **lsdev** command, or issue the **lsdev** command directly from the command line. The list obtained contains the name of the defined 64–port asynchronous adapters, their status (available or defined), location code (which identifies the 64–port asynchronous adapter physical device), and a device description.

Note: The status of a defined and configured device is *available*. The status of a device that is defined but not configured is *defined*.

List 64–Port Asynchronous Adapters Using SMIT

1. To list all defined 64–port asynchronous adapters regardless of whether they are available or not, use the **smit lsddev** fast path.
2. A scrollable list of all defined 64–port asynchronous adapters is displayed.
3. Exit the SMIT interface.

List 64–Port Asynchronous Adapters from the Command Line

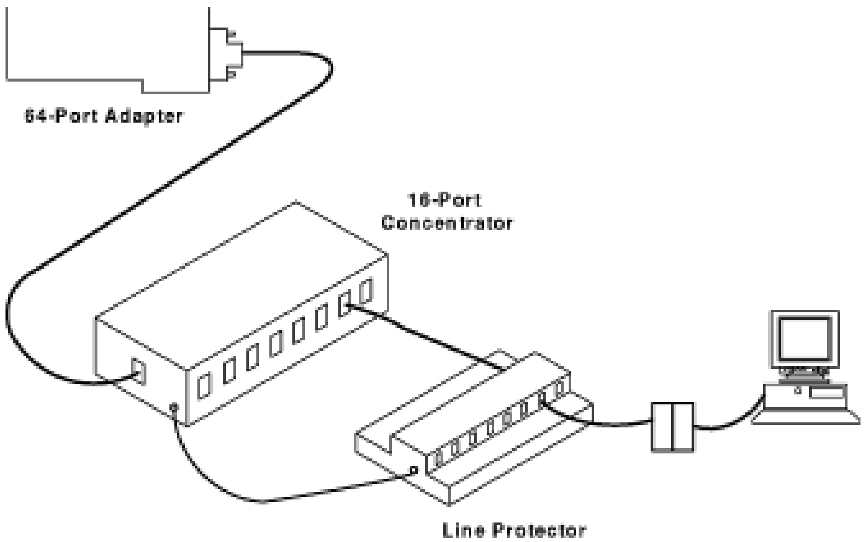
Enter the **lsdev** command at the command line as follows:

- To list all 64–port asynchronous adapters, regardless of whether they are available, enter: `lsdev –Cc adapter –t 64p232 –H`
- To list all 64–port asynchronous adapters that are available to the operating system, enter: `lsdev –Cc adapter –t 64p232 –H –S a`
- To list all 64–port asynchronous adapters that are *not* available to the operating system, enter: `lsdev –Cc adapter –t 64p232 –H –S d`

See the **lsdev** command for a detailed description and optional flags.

Transient Voltage Line Protector

The Transient Voltage Line Protector kit for the 16–port concentrator of the 64–port asynchronous adapter, clips off dataline voltage spikes that come from induced noise in a cabling environment. It is possible, in some circumstances, for voltage spikes to cause unrecoverable data interruptions to the port. In some severe cases, spikes may cause permanent damage to the 64–port hardware.



Transient Voltage Line Protector

The line protectors are effective in protecting the 16–port concentrators from voltage spikes due to thunderstorm activity and other sources of damaging electromagnetic or electrostatic energy.

Attention: Line protectors must not be used in place of outdoor surge protectors when going out of doors or between buildings.

Transient Voltage Line Protectors Terms and Conditions

The following are the terms and conditions necessary for transient voltage line protectors:

- If the concentrators are damaged during thunderstorms, replace the damaged hardware and install line protectors for all concentrators on the system.
- If ports on the concentrators frequently hang and cannot be reset, replace the bad or damaged hardware, if any, and install line protectors for all concentrators on the system.
- If ports on the concentrators hang and can be reset, it is not necessary to replace the concentrator. The addition of a line protector may help prevent further occurrences of the problem if the hang condition was not caused by bad cable connections.
- If the system is located in a hazardous storm area with medium to long (100 ft – 200 ft) cabling distances between the 16–port concentrators and devices.
- If the system is installed in a highly industrial area with long (150 ft – 200 ft) cabling distances between the 16–port concentrators and devices.

Follow the installation instructions and verify that the concentrator power cord ground is connected to a solid AC ground. Contact your hardware service representative for more information concerning transient voltage line protectors.

64–Port Problem Determination and Troubleshooting

Problems with the 64–port system generally fall into three categories:

- Hung ports which can be reset
- Hardware failure with the controller, concentrator, or devices
- Data integrity.

This procedure provides a list of actions and suggestions for debugging 64–port Asynchronous Adapter problems on the system unit. Use them to record data or correct the problem before placing a service call.

1. Before restarting the system with the **shutdown** command or reset key, check for obvious problems such as a disconnected cable or unplugged device.
2. Do not work on the system without proper electro–static discharge (ESD) protection.
3. Do not replace a 16–port concentrator because of a dead port without first having tried to reset the port through a hardware power cycle (shutdown) or software commands. In many cases, replacing the concentrator corrects the problem only because it performs a power cycle which, in this instance, may be the only action necessary.
4. Do not reconnect cabling without first dissipating possible built–up electrical charges.
5. Keep a repair log for the system unit. Look for repeat problems or patterns to problems which may help isolate the original source of trouble. This log will help isolate problems with a particular run, concentrator, or device and is invaluable in determining the root cause in a problem environment.

Known Problems with Asynchronous Adapters

Non–Shielded Wire

Any wiring scheme which runs several hundred feet is subject to atmospheric noise through magnetic induction. Shielding of cable grounds this noise and prevents its induction to signal wires. Non–shielded wiring subjects the system to port hangs, equipment damage, and erroneous data.

Corrective Action

Minimum Transient voltage surge suppression (TVSS) devices should be used to minimize noise and over current situations. A TVSS device should be used at both the device and concentrator end of any port having a higher than normal failure rate. Note that it may become necessary to use TVSS devices on all ports of a concentrator if system environmental problems are serious enough. In most cases, a good contact with ”earth ground” is ample enough to dissipate high voltage spikes.

Maximum Replace the wiring to meet specification.

Outdoor Wiring Run and/or Frequent Lightning Damage

Any wiring run that is located outdoors is *very* subject to atmospheric noise or lightning surges through magnetic induction. A direct lightning hit is not necessary to cause system damage.

Corrective Action

Minimum	Transient voltage surge suppression (TVSS) devices should be used to minimize noise and over current situations. A TVSS device should be used at both the device and concentrator end of any port having an out–door run.
Maximum	All outdoor wiring runs should be inside metal conduit, and the conduit should be grounded.

Concentrator to Device Exceeds 200 ft or 2500 Pico–Farads

Any wiring system that exceeds the rated capacitance is subject to data loss, port hangs, or both.

Corrective Action

Minimum	By moving the concentrator closer (within 200 ft) to its devices, you can take advantage of the maximum length of the controller–to–concentrator cable (2500 ft) and reduce the concentrator–to–device cable length.
Maximum	If the concentrator is already 2500 ft from the controller, then the 200 ft run will need a booster device.

Thunderstorms and ESD

Lightning strikes pose the greatest hazard to communication equipment circuitry. A lightning strike several miles away can induce voltages in the communication lines that are sufficient to destroy driver/receiver circuits.

Corrective Action

Cables should be adequately shielded while keeping the cable length to a minimum. Transient voltage surge suppression (TVSS) devices must be used if connecting cables out–of–doors between buildings. This should be done even if cabling is buried in the ground or in conduit.

Chapter 6. 128–Port Asynchronous Subsystem Overview

The 128–port asynchronous adapter subsystem meets the multiuser requirements for workstations in the open system environment. The subsystem can support up to 128 devices and offers higher speeds to the end user. It is an *intelligent* adapter with the ability to offload line protocol processing from the operating system.

The 128–port product consists of an adapter card, up to eight remote asynchronous node (RAN) units, each supporting 16 devices, for a total of 128 devices per adapter. The 128–port asynchronous adapter card resides in the system unit, and is connected by either EIA 422, direct cabling, EIA 422/EIA 232 synchronous modems, or Data Service Unit/Channel Service Units (DSU/CSUs) to the RAN. The number of 128–port asynchronous adapters that can be installed is dependent on the number of slots available in the system with a hardware limit of seven adapters per bus (an optional I/O bus can provide eight additional slots).

To understand the 128–port asynchronous adapter in more detail, consider:

- 128–Port Asynchronous Adapter, on page 6-1
- 16–Port Remote Asynchronous Node, on page 6-1
- Topology, on page 6-3
- RAN Removal, on page 6-4
- RAN Termination, on page 6-4
- RAN Node Numbers, on page 6-5
- The Number of 128–Port Asynchronous Adapters Per System, on page 6-5

128–Port Asynchronous Adapter

The 128–port asynchronous adapter is an intelligent dual–channel EIA 422 synchronous board for the AIX system. The 128–port asynchronous adapter features the following main components:

- On card microprocessor
- Up to 1 Megabyte of memory
- Two high–speed EIA 422 synchronous lines (or channels) used to communicate with 128–port RAN at data rates of up to 2.458 Mbps. The synchronous lines are also capable of communicating with EIA 422 and EIA 232 synchronous modems.
- Available for Micro Channel, ISA, or PCI bus.

16–Port Remote Asynchronous Node (EIA 232)

The 128–port remote asynchronous node (RAN) is a complete subsystem and contains the following main components:

- 25 MHz 80C186 microprocessor
- 128K of RAM and 16K of EPROM
- Sixteen 16C550 universal asynchronous receiver/transceivers (UARTs) for the 16 EIA 232 asynchronous serial ports
- High–speed synchronous EIA 232 port for communication with the adapter and other RANs.

The RAN receives data packets from the adapter at data rates of up to 1.2 Mbps, and then distributes the data, as appropriate, to the 16 EIA 232 ports. Data received by the EIA 232 ports is similarly divided into packets and sent back to the adapter over the high–speed synchronous line. The EIA 232 ports operate at data rates of up to 57,600 bps.

For local applications, where the work groups are not located more than 300 m (1000 ft) from the system, RANs can be connected to a 128–port asynchronous adapter through 4– or 8–conductor twisted–pair cable (the 8–conductor cable is recommended). With the 8–conductor cable, data is transferred between RANs and the adapter at rates of up to 1.2 Mbps. With 4–wire cable, the maximum synchronous data rate is 460 Kbps.

For more information concerning cable types, lengths, and baud rates, refer to ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7.

16–Port Enhanced Remote Asynchronous Node (EIA 232)

The 128–port remote asynchronous node (RAN) is a complete subsystem and contains the following main components:

- 25 MHz 80C186 microprocessor
- 128K of RAM and 16K of EPROM
- Sixteen 16C550 universal asynchronous receiver/transceivers (UARTs) for the 16 EIA 232 asynchronous serial ports
- High–speed synchronous port for communication with the adapter and other RANs.

The RAN receives data packets from the adapter at data rates of up to 2.458 Mbps, and then distributes the data, as appropriate, to the 16 EIA 232 ports. Data received by the EIA 232 ports is similarly divided into packets and sent back to the adapter over the high–speed synchronous line. The EIA 232 ports operate at data rates of up to 230 Kbps.

Adapter configuration software will autodetect and configure RANs of multiple types on a single line. Line speed will be limited to 1.2 Mbps when connected on the same line as the original 128–port asyncronous node EIA 232.

For local applications, where the work groups are not located more than 300 m (1000 ft) from the system, RANs can be connected to a 128–port asynchronous adapter through 4– or 8–conductor twisted–pair cable (the 8–conductor cable is recommended). With the 8–conductor cable, data is transferred between RANs and the adapter at rates of up to 1.2 Mbps. With 4–wire cable, the maximum synchronous data rate is 460 Kbps.

For more information concerning cable types, lengths, and baud rates, refer to ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7.

16–Port Enhanced Remote Asynchronous Node (EIA 422)

The 128–port remote asynchronous node (RAN) is a complete subsystem and contains the following main components:

- 16 MHz 80C186 microprocessor
- 128K of RAM and 16K of EPROM
- Sixteen 16C550 universal asynchronous receiver/transceivers (UARTs) for the 16 EIA 422 asynchronous serial ports
- High–speed synchronous EIA 232 port for communication with the adapter and other RANs.

The RAN receives data packets from the adapter at data rates of up to 2.458 Mbps, and then distributes the data, as appropriate, to the 16 EIA 422 ports. Data received by the EIA 232 ports is similarly divided into packets and sent back to the adapter over the high–speed synchronous line. The EIA 422 ports operate at data rates of up to 230 Kbps.

Adapter configuration software will autodetect and configure RANs of multiple types on a single line. Line speed will be limited to 1.2 Mbps when connected on the same line as the original 128–port asyncronous node EIA 232.

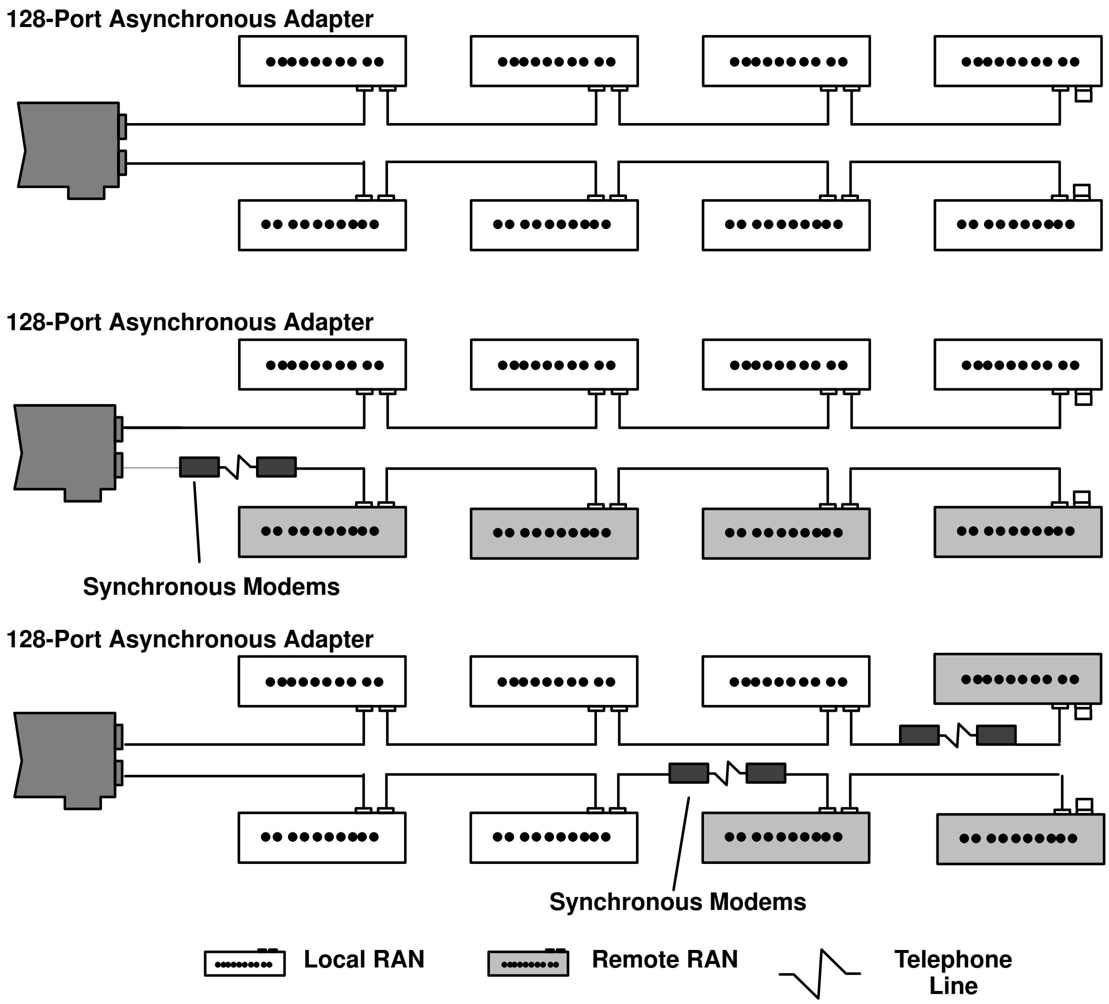
For local applications, where the work groups are not located more than 300 m (1000 ft) from the system, RANs can be connected to a 128–port asynchronous adapter through 4– or 8–conductor twisted–pair cable (the 8–conductor cable is recommended). With the 8–conductor cable, data is transferred between RANs and the adapter at rates of up to 2.458 Mbps. With 4–wire cable, the maximum synchronous data rate is 460 Kbps.

For more information concerning cable types, lengths, and baud rates, refer to ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7.

Topology

Each 128–port asynchronous adapter has two synchronous lines. Up to four RANs can be connected to a single synchronous line in daisy–chain fashion. That is, the first RAN is connected to one of the adapter’s lines, the second RAN is connected to the first RAN, and so on (up to four RANs).

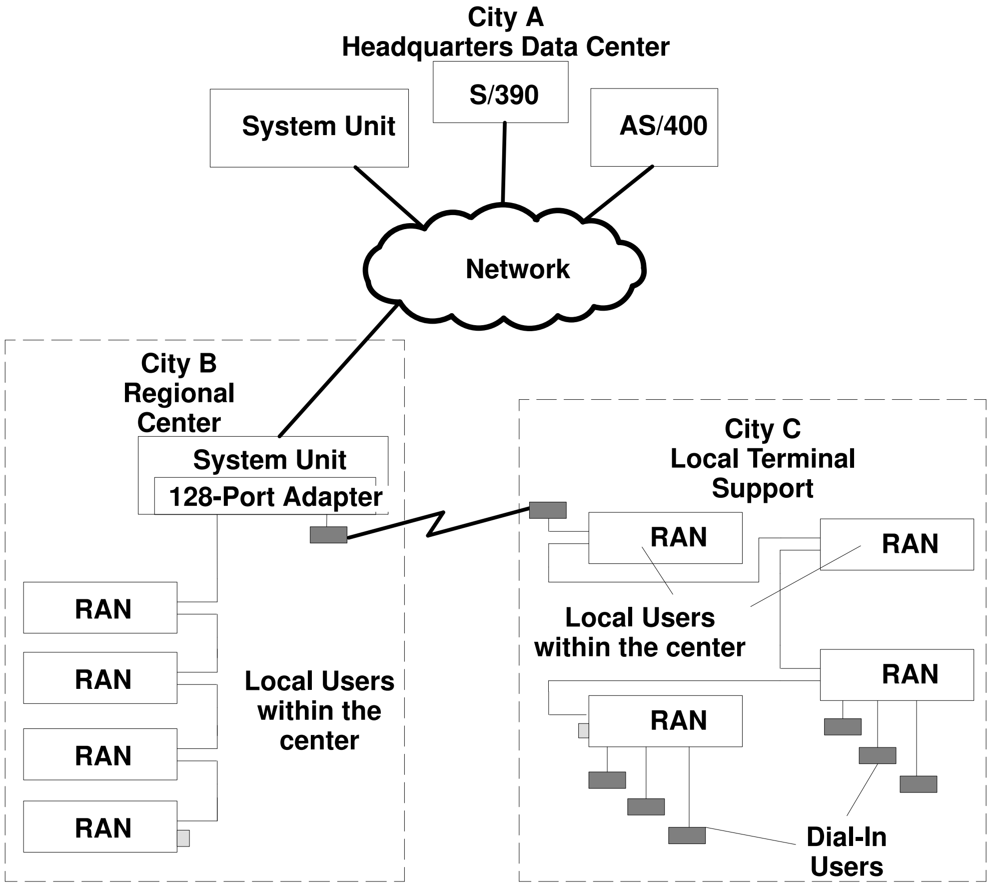
The RANs can be attached to the 128–port adapter with a direct local connection, a synchronous modem connection, or a combination of the two. The following figure shows some of the possible connection types.



Local and Remote Configuration Scenarios

Note: Only a single modem connection is allowed per adapter line.

See the following figure for an example of a scenario in a distributed environment using the 128–port asynchronous adapter.



Distributed Environment Solution

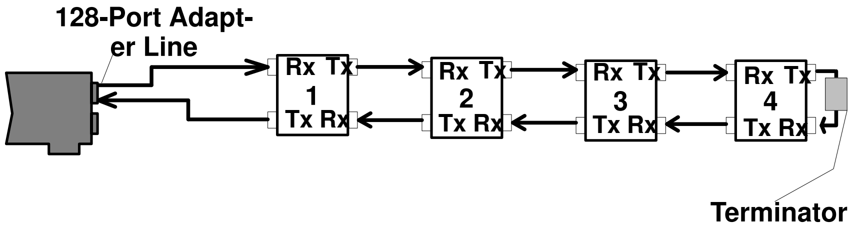
- The system unit in City B can support up to 64 terminal/printers for local users within the center.
- City C is connected through synchronous modems to the system unit in City B.
- The RANs in City C provide support for both local and dial–in users.
- The system unit in the regional center at City B is connected by the backbone network using a networking protocol to the Headquarters Data Center in City A.
- All users in Cities B and C can have access to the Headquarters Data Center.

RAN Removal

RANs are assigned physical node numbers (set by the operator during installation, see ”Setting a RAN Node Number”, on page 6-37). The node number is used by the adapter to route data to and from a specific RAN. If a particular RAN is turned off or removed from the daisy chain, the 16 devices on that RAN become unavailable to the system, but the rest of the system remains unaffected. Since the RAN’s IN and OUT/T ports are of opposite gender, a direct or local RAN can be removed from the middle of a daisy chain by plugging the cables together so that the chain remains unbroken. To remove the last RAN, simply plug the terminator plug into the end of the daisy–chain cable.

RAN Termination

RANs always receive input data via their IN ports and always transmit output data from their OUT/T ports. When multiple RANs are daisy–chained together, data travels in a circular fashion. Thus, if there are four RANs connected to an adapter’s synchronous port, data from the adapter to RAN 4 must pass through RAN 1, 2, and 3 before reaching RAN 4. At the same time, data from RAN 1 to the adapter must travel the full circle through RAN 2, 3, and 4 before being returned to the adapter. Refer to the following figure for an example of the data flow.

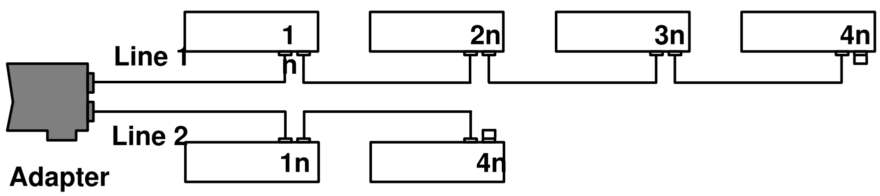


Logical Data Flow in a Daisy-Chain Configuration

To make the loop complete, a terminator plug must be installed on the OUT/T port of the last RAN in the daisy chain. This plug ties all of the OUT/T port's output signals back to their corresponding input signals (TxD to RxD, TxC to RxC, and so on). The RAN's OUT/T port input signals are hard–wired to its IN port output signals, so once any RAN's output data reaches the terminator plug on the last RAN, it is passed back through all of the RANs until it is ultimately received by the adapter. Note that if only one RAN is installed, it is by default the last one and needs to have a terminator plug (IBM PN 43G0926) installed.

RAN Node Numbers

The adapter identifies RANs by their node numbers. Each RAN in a daisy chain must have a unique node number (*1n* through *4n*), which must be set during installation (see " Setting a RAN Node Number", on page 6-37). The node numbers must be assigned in ascending order with the lowest number assigned to the RAN closest to the adapter. You can skip node numbers (to facilitate insertion of additional RANs at a later date), as long as the ascending sequence is maintained. The following figure summarizes the node number assignment.



RAN Node Numbers

Number of 128–Port Asynchronous Adapters per System

The number of adapters that can be installed in a system is dependent on the number of available slots. See "Product Selection Considerations" in Chapter 1 for more information.

Planning for Your 128–Port Asynchronous Adapter

Successfully installing and configuring a 128–port asynchronous adapter and its associated remote asynchronous node (RAN) requires careful planning. The following discusses planning issues for the 128–port asynchronous adapter subsystem.

A planning worksheet, on page A-1 is provided to assist you in gathering information needed for SMIT before beginning the configuration process.

The major areas of configuration for the 128–port asynchronous adapter are:

- RAN configuration
  - Local or direct RAN attachment
  - Remote RAN attachment
- RAN attached device configuration
  - EIA 232 asynchronous terminal configuration
  - EIA 232 asynchronous printer/plotter for a RAN
  - EIA 422 asynchronous terminal configuration printer/plotter

Planning Steps

1. Determine topology for each line based on user requirements and physical locations:
   - How many devices will be supported?
   - What kinds of devices?
   - Are the devices local, remote, or a combination of both?
   - What cables are needed, and can existing cables be used?

   The type and number of devices, their geographical location, and the expected data load associated with the devices will determine the types of cables to be used.

   Refer to ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7 for information on cabling requirements.

   Refer to ”Connecting a RAN to a 128–Port Asynchronous Adapter”, on page 6-28.
2. Select RAN node IDs. Refer to ”Setting a RAN Node Number”, on page 6-37.
3. Determine the values for the SMIT 128–port asynchronous adapter configuration parameters.
   a. Use the ”Line Speed and Line Cable Type Determination Flowchart”, on page 6-31 to select the *line speed* and *line cable type*.
   b. To select the *node comm mode*, follow the instructions in ”Determining a RAN’s Connection Type and SMIT Node Comm Mode Values”, on page 6-32.

128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios

There are three basic wiring modes for connecting remote asynchronous nodes (RAN) to an adapter or to each other:

- 8–wire direct
- 4–wire direct
- 8–wire synchronous modem

Additionally, there are four types of modular plugs that can be used to connect your asynchronous device to the RAN's RJ–45 10–pin jack:

- 4–pin RJ–11 plugs (EIA 232 only)
- 6–pin RJ–11 plugs
- 8–pin RJ–45 plugs
- 10–pin RJ–45 plugs.

This article discusses the following cables and jacks including their pin–outs and uses.

Attachment Methods

The following figure illustrates three general methods for attaching remote asynchronous nodes (RANs) to the 128–port asynchronous adapter.

Attachment Methods and Their Benefits

Attachment Method	Recommended Environment	Benefit
8-wire direct	Moderate to heavy data loads	Maximum performance
4-wire direct	Light data loads*	Reduced cabling cost
Synchronous modems	Light data loads*	Remote location

Note: *Contact your sales representative for information on the 4–wire direct and synchronous modem cables.

Cable Planning

In order to make the necessary connections to the 128–port asynchronous adapter, your setup person needs to know the type of connection and which cables to use. The following cable planning chart shows what cables are used.

Cable Planning

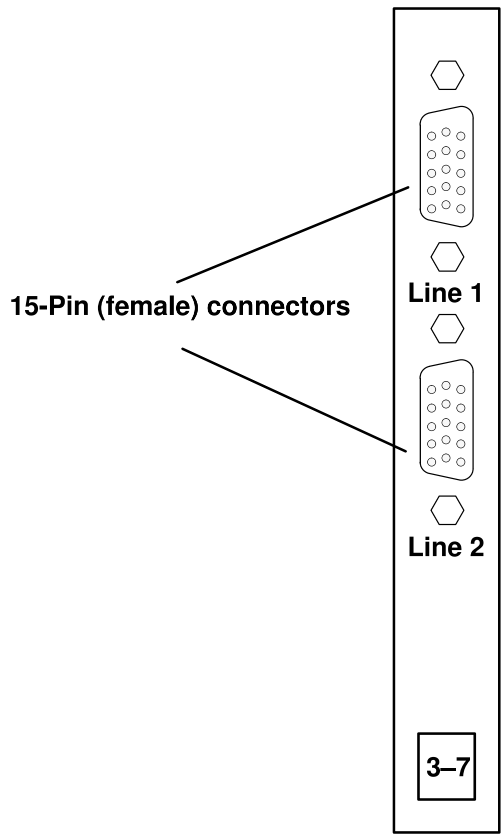
Cable Letter	Name/Description
NC	8-Wire Daisy-Chain EIA-422 cable (9 inch)
NB	8-Wire Daisy-Chain EIA-422 cable (15 ft)
ND	4-Wire Daisy-Chain EIA-422 cable
NE and NF	8-Wire EIA-232 Synchronous Modem cables
NG and NH	8-Wire EIA-422 Synchronous Modem cables
NL	RAN-to-Device EIA-232 cable
NM	RAN-to-Modem EIA-232 cable
NK[2]	RJ-45 to DB-25 Converter cable (4 cables in a kit)
D[2]	EIA-232 Asynchronous cable
E[2]	EIA-232 Printer/Terminal Interposer

Notes:

1. Contact your sales representative for information on cable availability.
2. For additional information on cable planning, see ”128–Port Async Controller to Remote Async Node Cables” in *Adapters Information for Micro Channel Architecture Systems*.

128–Port Asynchronous Adapter Connectors

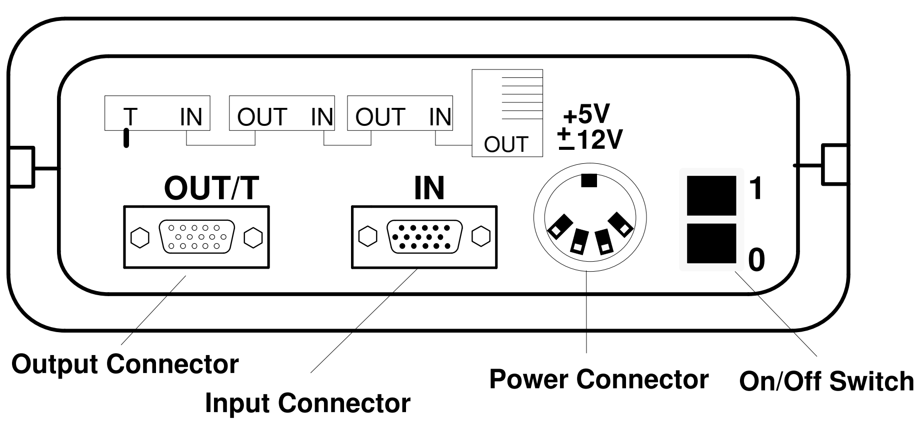
The following figure illustrates the location of the EIA 422 channels or lines of the 128–port asynchronous adapter.



Location of 128-Port Asynchronous Adapter Lines

RAN EIA 422 and Power Connectors

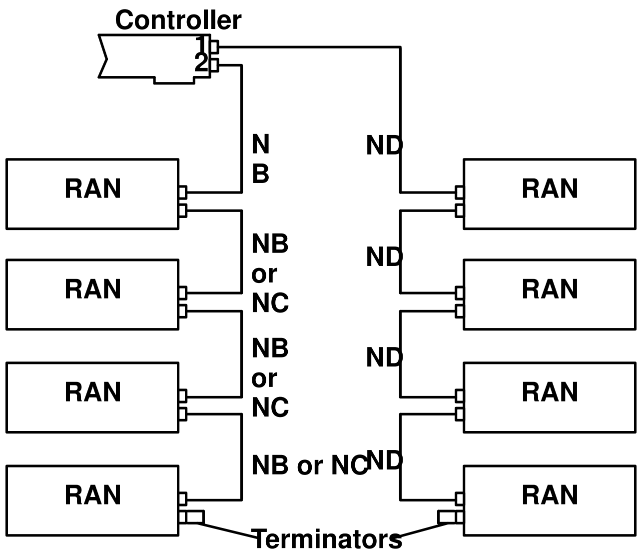
The following figure illustrates the RAN's connectors, power connectors, and its On/Off switch.



RAN and Power Connectors

Local 8–Wire and 4–Wire Direct Cabling Scenario

The following figure shows a typical configuration in which eight RANs are attached to the 128–port asynchronous adapter using both 4–wire and 8–wire cabling.

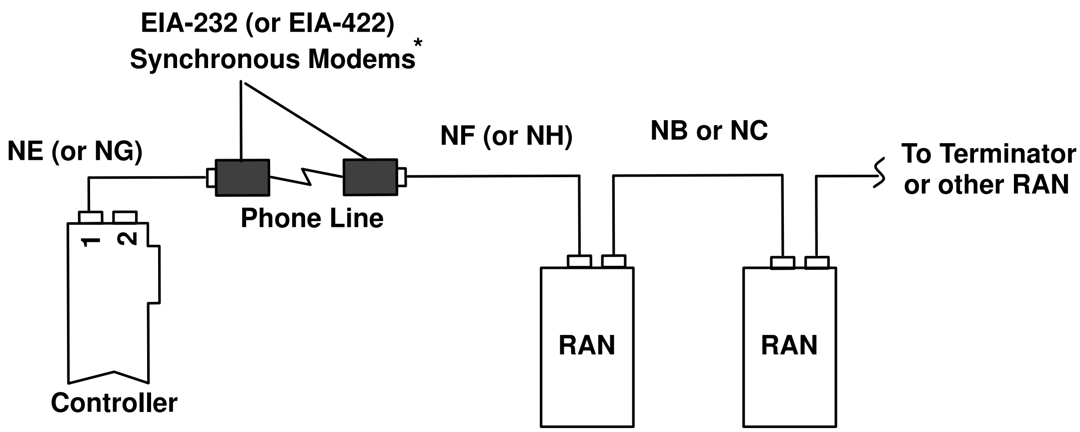


Typical 8-Wire and 4-Wire Topology

Note: 8–wire cabling cannot be used in combination with 4–wire cabling on the same adapter line.

Synchronous Modems Cabling Scenario

The following figure illustrates the use of EIA 232 or EIA 422 synchronous modems in typical 128–port asynchronous adapter configurations. Note that each configuration requires a unique set of customer–supplied cables for modem attachment.



Synchronous Modems Cabling Scenario

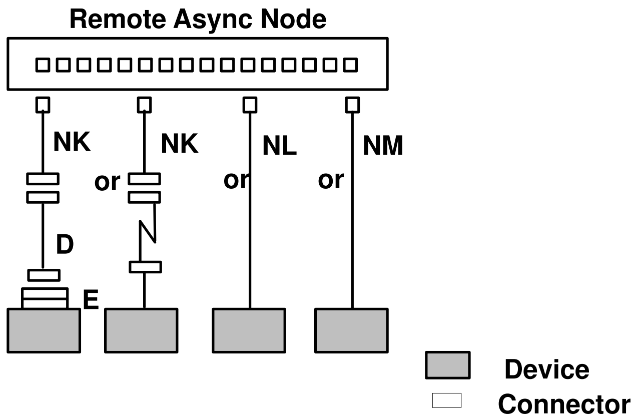
Note: * Cables NE, NG, NF, and NH are not available from IBM.

Any combination of 8–wire cabling and synchronous modems can be used to attach remote asynchronous nodes. However, 4–wire cabling cannot be used in combination with synchronous modems on the same adapter line.

Note: Only one pair of synchronous modems can be used per adapter line.

Device Cabling Scenario

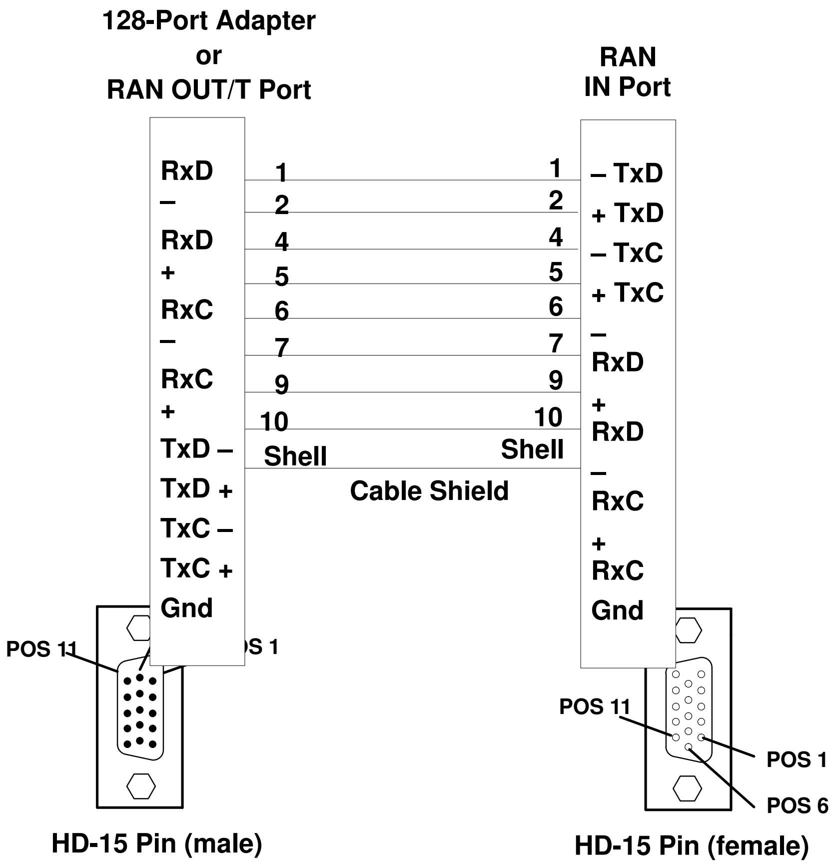
A choice of cables can be attached to any of the 16 RAN ports. These ports are labeled 0 through 15 and accept 4–, 6–, 8–, and 10–pin RJ type connectors. Refer to the following figure for an example of device cabling. The device can be a display, a printer, or a plotter.



Device Cabling Scenario

8–Wire Direct Daisy–Chain Cable (NC and NB) Specifications

8–wire direct is the standard method for connecting RANs to the 128–port adapter. It provides transmit and receive data signals plus discrete clock signals for transmit and receive data. This permits synchronous data rates of up to 2.458 Mbps, which results in the maximum data throughput under moderate to heavy loads. See the following figure for an illustration of the 8–wire direct daisy–chain cable.



8-Wire Direct Wiring (Cable NC or NB)

The maximum length of a daisy chain is dependent on the synchronous data rate used for the 128–port line to which it is connected. The following table lists the maximum ”cumulative” daisy–chain cable lengths for various line speeds or baud rates.

Baud Rate	Maximum Cable Length (24 AWG Twisted Pair, 12 pF/ft)
up to 460K	2000 ft
up to 1.2M	1000 ft
up to 2.4M	300 ft

Note: The information in this table represents the maximum ”recommended” supported configurations and are intended for general guidelines only. Configurations above these recommendations can be used, but be aware that, depending on your particular operating environment, loss of data integrity and possible hardware failures may occur.

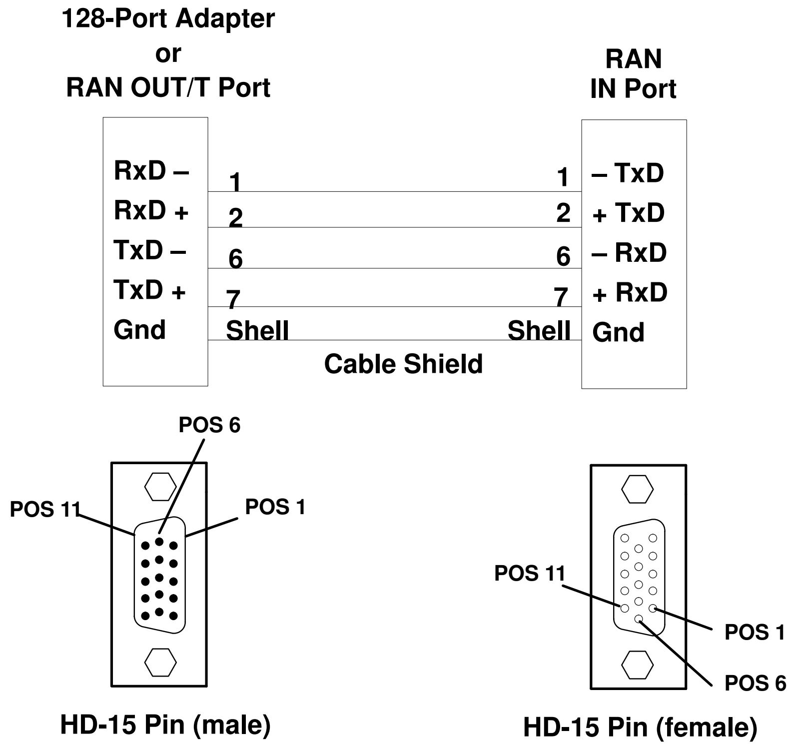
For example, to run a synchronous line at 1.2 Mbps, the total length of all daisy–chain cables for that synchronous line cannot exceed 1000 ft. Therefore, a single RAN could be placed 1000 ft from the host adapter, or four RANs could be spaced at 250–ft intervals and still operate at 1.2 Mbps.

4–Wire Direct Daisy–Chain Cable (ND) Specifications

4–wire direct wiring can be used to connect RANs where longer synchronous cable runs are necessary. While not as fast as 8–wire connections (the maximum data transfer rate is 460 Kbps), this wiring method is more economical and is sufficient in all but the most demanding high–performance applications (terminal users should see no degradation in performance). In the 4–wire direct–wiring mode, the clock signals are encoded with the receive and transmit data signals (non–return–to–zero inverted encoding at 230 Kbps and FM0 encoding at 460 Kbps), so only two twisted pairs are required.

The cable has four conductors, two twisted–pair, and is shielded on the outside. If built to a length of 300 m (1000 ft) or less, conductors should be 28 AWG (stranded wire) with a capacitance rating of 52 pF/m (16 pF/ft) or less (Belden type 9804 or equivalent). For lengths greater than 300 m (1000 ft), conductors should be 24 AWG (stranded wire) with a capacitance rating of 52 pF/m (16 pF/ft) or less (Belden type 9829 or equivalent).

See the following figure for an illustration of the 4–wire direct daisy–chain cable.



4-Wire Direct Wiring (Cable NC or NB)

Note: The 4–wire direct daisy–chain cable is a customer–supplied cable.

The 128–port asynchronous adapter supports two adapter line–baud rates in 4–wire direct–attach mode. The following table shows the maximum allowable adapter line length for each supported baud rate. The adapter line length is the actual cable length from the adapter to the last remote asynchronous node in the adapter line.

Adapter Line Baud Rate	Total Adapter Cable Length	
bps	**m**	**ft**
230000	400	1350
460000	300	1000

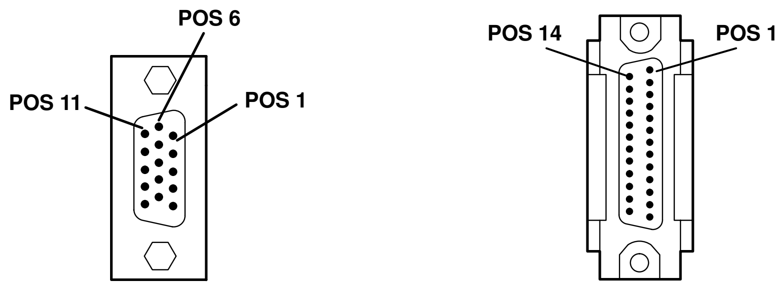
Note: This table assumes no intermediate connectors between remote asynchronous nodes. Each additional connection will decrease the maximum allowable adapter line length by approximately two percent due to increased line capacitance.

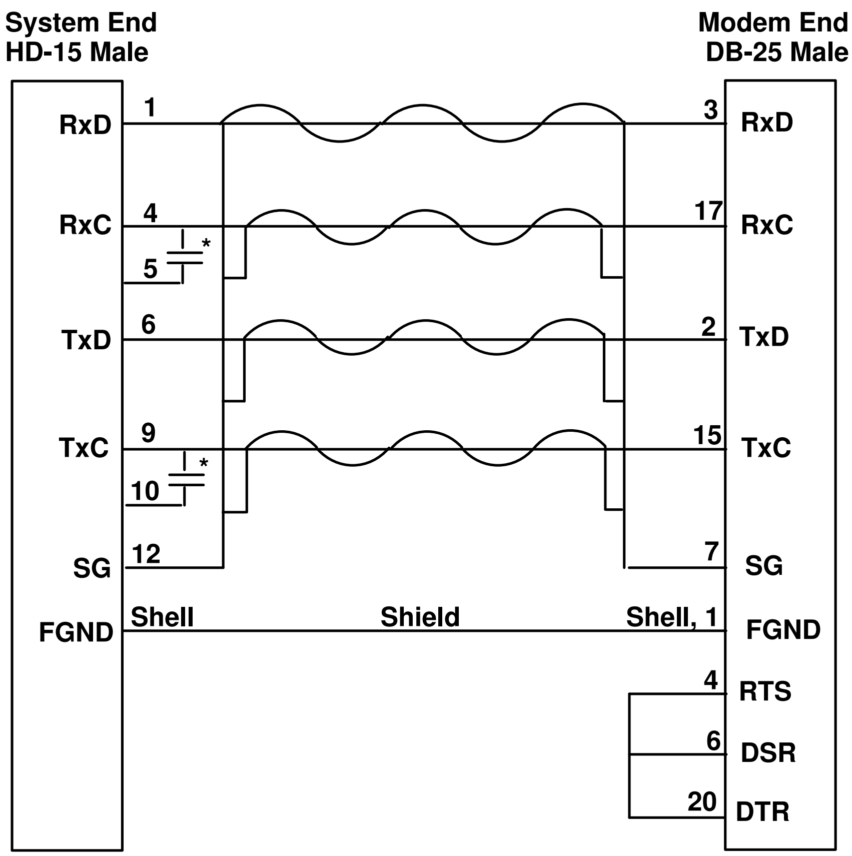
8–Wire Synchronous Modem (NE, NF, NG and NH) Cable Specifications

8–wire synchronous modem wiring allows RANs to be installed in remote locations and connected to the adapter using EIA 232 or EIA 422 synchronous modems. The 128–port asynchronous adapter and the RAN are designed so that the synchronous ports can support either EIA 422 or EIA 232 line levels.

EIA 232 Synchronous Modem Cables NE and NF

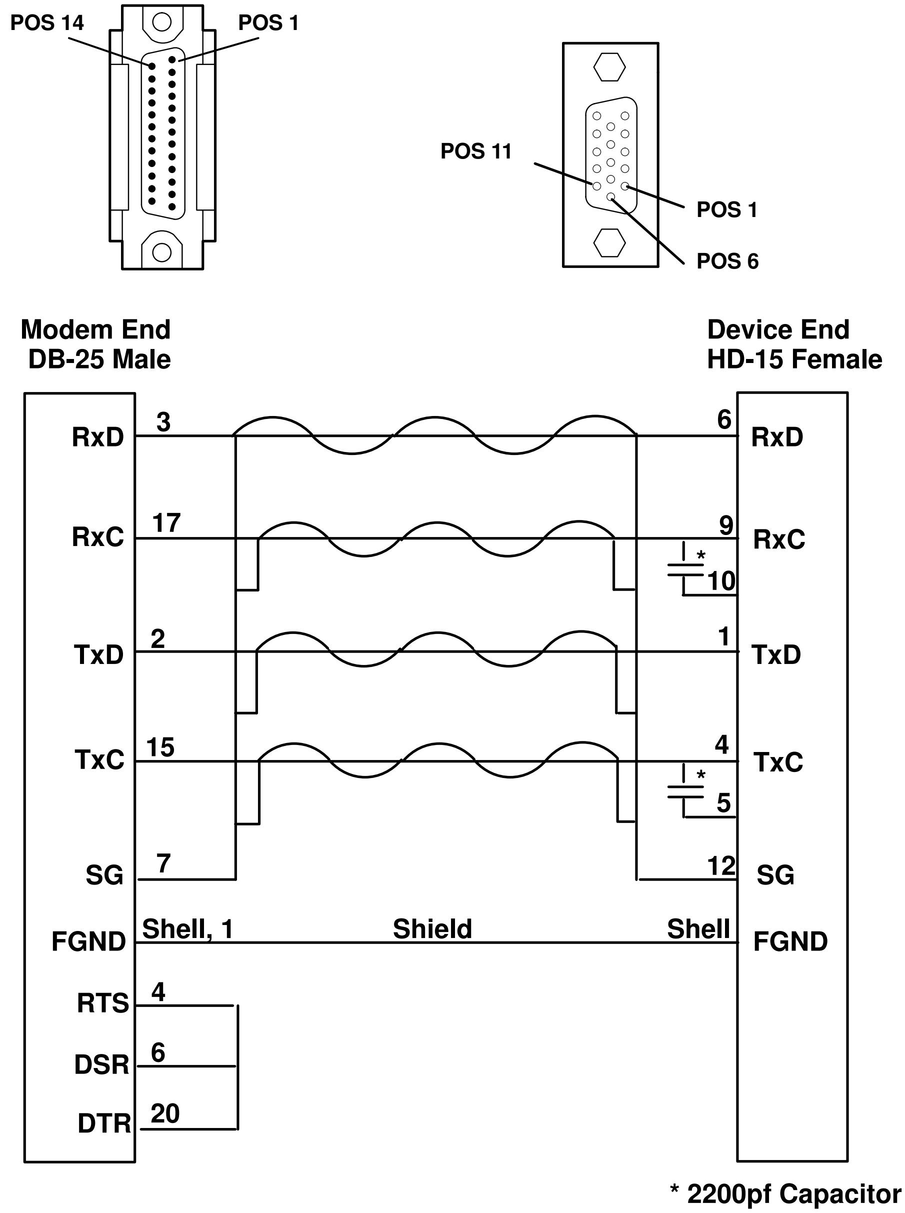
The following figure, Cable NE, shows a 128–Port asynchronous controller EIA 232 modem cable and system.





* 2200pf Capacitor

The following figure, Cable NF, shows a 128–Port asynchronous controller EIA 232 modem cable and device.



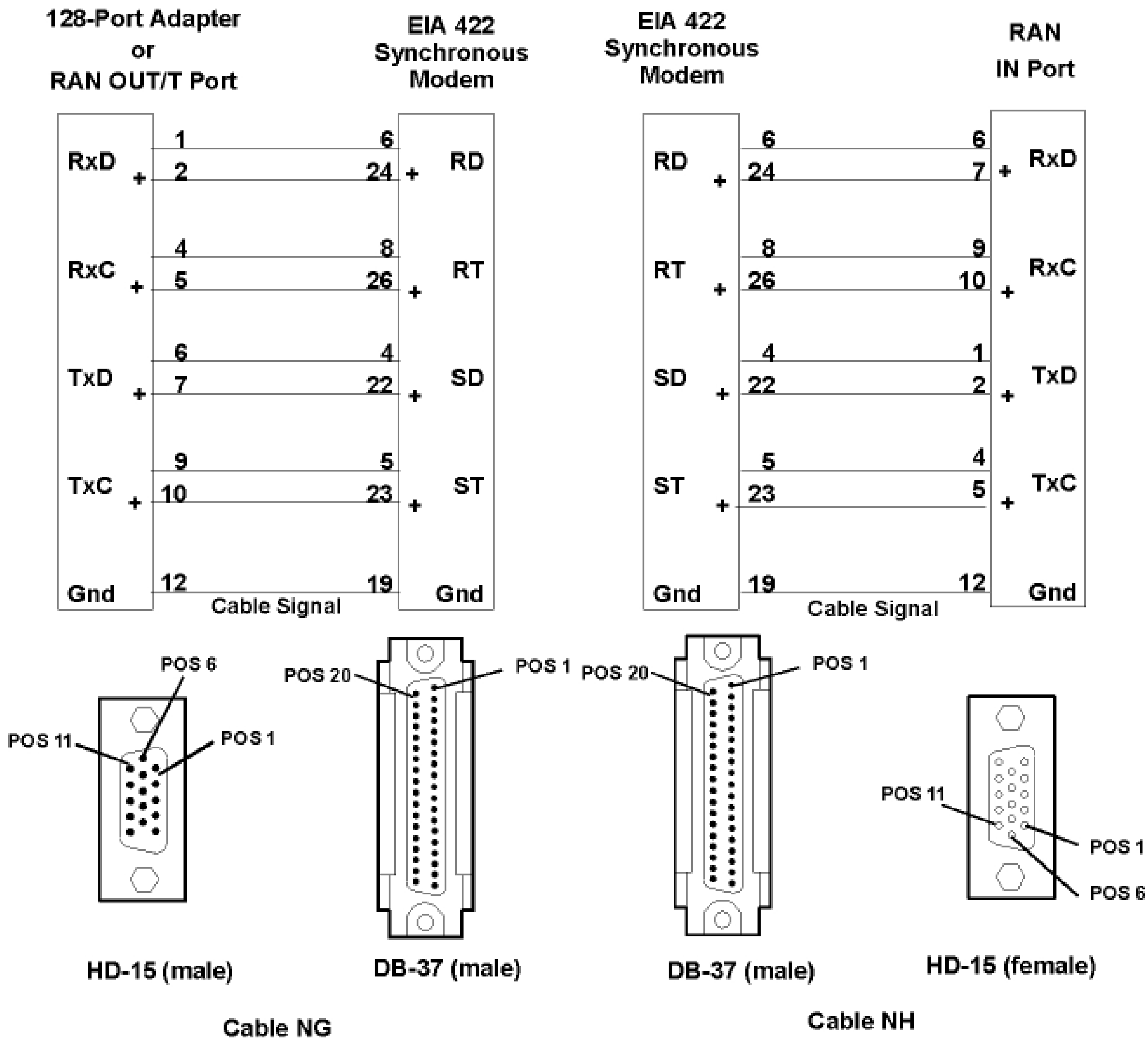
The cable has eight twisted–pair conductors and is shielded on the outside. Cable length can be from 1.8 m (6 ft) to 3.7 m (12 ft). Conductors should be 24 AWG (stranded wire) with a capacitance rating of 41 pF/m (12.5 pF/ft) or less.

The 128–port asynchronous controller supports multiple controller line baud rates in EIA 232 synchronous–modem–attach mode. However, to ensure data integrity, controller line baud rates of 57.6 Kbps or less are recommended.

EIA 422 Synchronous Modem Cables NG and NH

The cable has eight conductors, four twisted–pair, and is shielded on the outside. If built to a length of 300 m (1000 ft) or less, conductors should be 28 AWG (stranded wire) with a capacitance rating of 52 pF/m (16 pF/ft) or less (Belden type 9806 or equivalent). For lengths greater than 300 m (1000 ft), conductors should be 24 AWG (stranded wire) with a capacitance rating of 52 pF/m (16 pF/ft) or less (Belden type 9831 or equivalent).

The following figure shows the 128–port asynchronous adapter supporting multiple adapter line–baud rates in EIA 422 synchronous modem–attach mode. See the Adapter Line Baud Rate table, on page 6-12.



EIA 422 Synchronous Modem Cable (NG and NH)

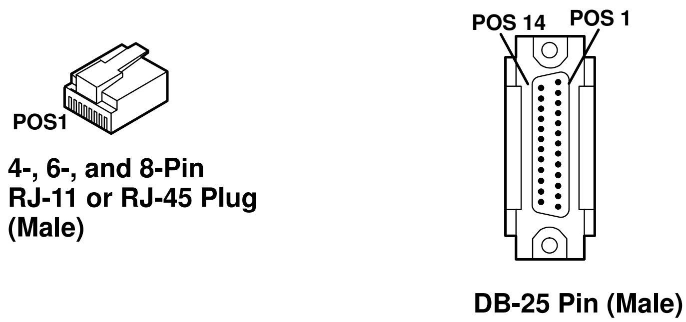
RAN 16–Port Device Cables

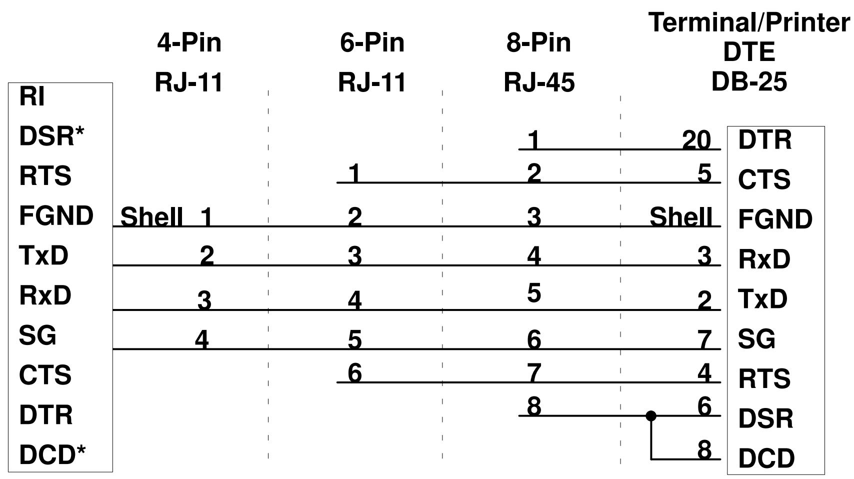
There are four types of modular plugs that can be used with the RAN RJ–45 10–pin jack:

- 4–pin RJ–11 plugs
- 6–pin RJ–11 plugs
- 8–pin RJ–45 plugs
- 10–pin RJ–45 plugs

RAN–to–Device EIA 232 Cable (NL and NM)

The following figure, Cable NL, shows a customer–supplied cable for connecting remote asynchronous node 16–Port EIA 232 to a printer or terminal device.

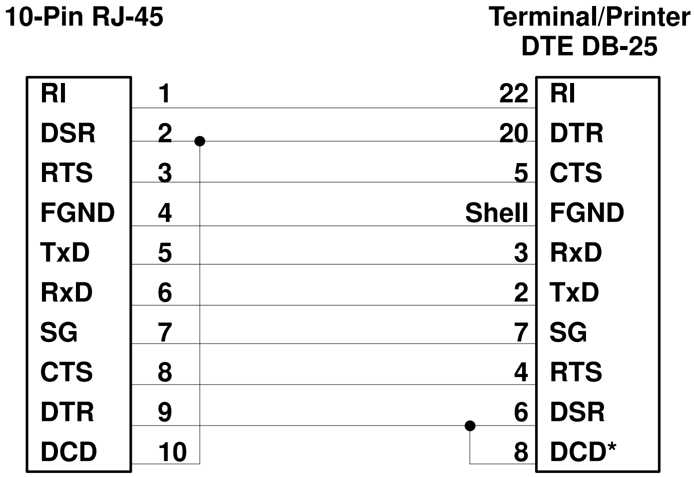




RAN to Printer/Terminal Cable (NL)
for 4- and 6-Pin RJ-11, and 8-Pin RJ-45 Plugs

Cable length can be up to 61 m (200 ft). For 115Kbps the maximum cable length is 80 ft. For 230Kbps the maximum cable length is 40 ft. Use overall foil/braid shielded multiconductor cable with a capacitance rating of 41 pF/m (12.5 pF/ft) or less. Conductors should be 28 AWG (stranded wire). For lengths less than 61 m (200 ft), higher capacitance cable can be used, as long as the total capacitance (including intermediate connectors and cables) does not exceed 2500 pF.

The following diagram illustrates cable NL using a 10–pin RJ–45 plug.

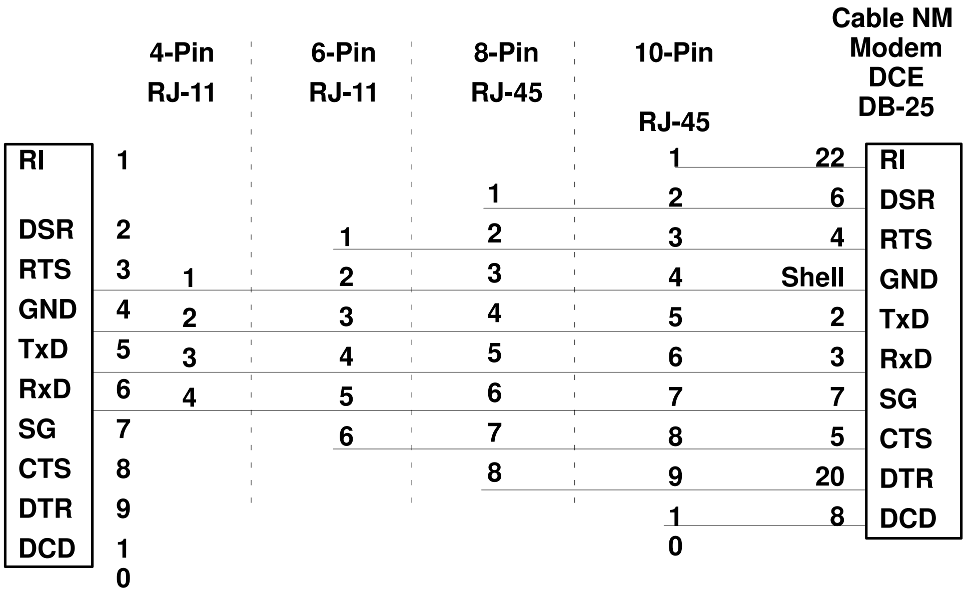


**RAN to Printer/Terminal Cable (NL)
for 10-Pin RJ-45 Plug**

Attention: The receivers and drivers used in most asynchronous communications devices are sensitive to electrostatic discharge (ESD). To reduce the possibility of exposure to ESD, observe the following cabling practices when building or using device cables for attachment to the RAN 16–Port EIA 232:

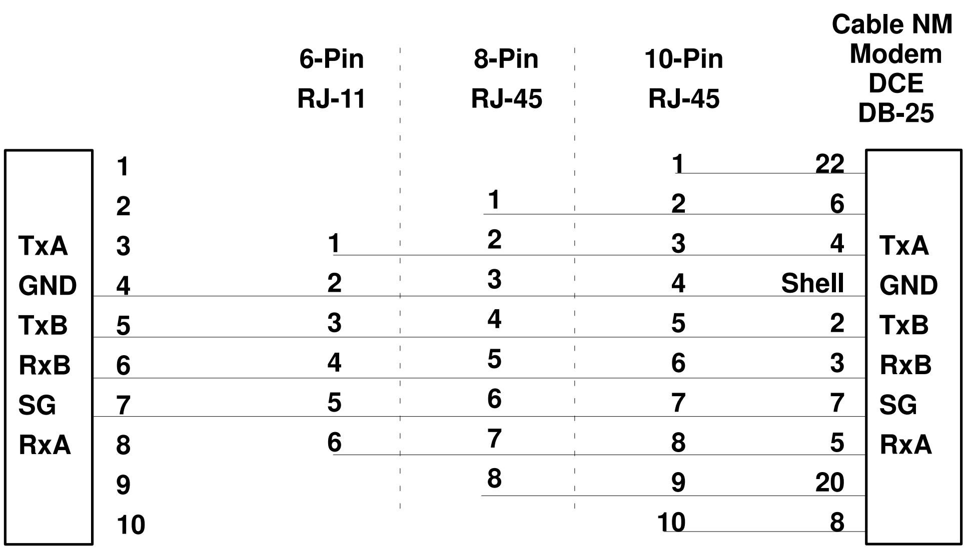
1. Do not build a cable that has exposed conductors, leads, or pins that could be touched by someone not protected against ESD. Avoid the use of punchdown blocks and patch panels which have exposed terminator/pins. If you use intermediate connectors or cables, be sure to discharge them to ground before plugging them into equipment.
2. Do not run any cables outdoors without having proper transient voltage suppression devices installed.
3. Do not route cables near or around items such as power transformers, high–power switching devices, and refrigeration units.
4. Use shielded cables. All wires should be terminated, not floating. The shield should be connected to shield ground at the remote asynchronous node.

Cable NM is used to connect modems to the RAN's RJ–45 connectors. See the following diagram for an illustration of the cable NM and EIA 232 using 4– and 6–pin RJ–11 and 8– and 10–pin RJ–45 plugs.



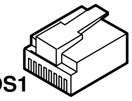
RAN to Modem Cable (NM)
for 4- and 6-Pin RJ-11, and 8- and 10-Pin RJ-45
Plugs

The following diagram shows cable NM and EIA 422 using 6–pin RJ–11 and 8– and 10–pin RJ–45 plugs.

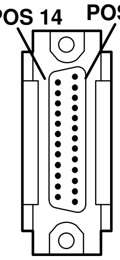


RAN to Modem Cable (NM)
for EIA 422 Plugs
(4-Pin Connector not Supported)

The following diagram illustrates the plugs used with cables NL and NM.



4-, 6-, and 8-Pin R-J11 or RJ-45 Plug (Male)



DB-25 Pin (Male)

Notes:

1. These cable assemblies are shielded.
2. These cable assemblies and the 64–port RJ–45 to DB–25 converter cable (FC 6402) are not interchangeable.

Cable length can be up to 61 m (200 ft). Use overall foil/braid shielded multiconductor cable with a capacitance rating of 41 pF/m (12.5 pF/ft) or less. Conductors should be 28 AWG (stranded wire). For lengths less than 61 m (200 ft), higher capacitance cable can be used, as long as the total capacitance (including intermediate connectors and cables) does not exceed 2500 pF.

4–, 6–, 8– and 10–Pin Plug Considerations (EIA 232)

The 8– and 10–pin RJ–45 connections are suitable for both terminal and printer attachment. The 4– and 6–pin RJ–11 connections are not recommended for printer attachment.

10–Pin RJ–45 Plugs

The 10–pin RJ–45 plug carries all eight of the EIA 232 signals supported by RANs, plus the two ground lines, Signal Ground (SG) and Chassis Ground (GND). The 10–pin configuration includes the modem control lines Ring Indicator (RI) and Data Carrier Detect (DCD).

8–Pin RJ–45 Plugs

The 8–pin RJ–45 plug supports all of the EIA 232 signals except the modem control lines RI and DCD. It is ideal for use with terminals and printers that require full hardware handshaking, both Data Set Ready (DSR) and Clear To Send (CTS) must be satisfied for data transmission to occur.

> **Note:** The physical location of DCD and DSR may be swapped. The operating system permits software ”rewiring” of the RJ–45 connectors so that DCD is available in 8–pin configurations. When connecting to Data Terminal Equipment (DTE) devices (such as terminals and printers), you can either use the standard configuration shown above, or the ALTPIN configuration (see ”RJ–45 Connection to Modem Considerations”, on page 6-21).

6–Pin RJ–11 Plugs

6–pin RJ–11 plugs can be used in hardware handshaking situations that require only RTS and CTS to be available. This cable is not recommended for printer attachment.

4–Pin RJ–11 Plugs

The 4–pin RJ–11 plug can be used in situations requiring software flow control (XON/XOFF) but no hardware handshaking. No hardware handshake lines are available with this configuration. This is useful for the popular ”three–wire” connection (TxD, RxD, and SG) used for terminals that support XON/XOFF handshaking. The 4–pin plug is not supported for printers that must have CTS connected to work properly.

When using the 4–pin plug with terminals, DCD sensing must be disabled in the software. This can be done with SMIT, or with the command:

```
chdev –l ttyXX –aforcedcd=enable
```

RJ–45 Connection to Modem Considerations (EIA 232)

Modems are generally equipped with DB–25 connectors, and the NE cable is suitable for this use. Since the modem control lines RI and DCD are on pins 1 and 10, respectively, of the RAN RJ–45 jack, the 10–pin RJ–45 plug is ideal for modems.

Due to the wide availability of RJ–45 8–pin cable, the AIX operating system software incorporates an optional feature called ALTPIN, which swaps the logical functions of DSR with DCD. When ALTPIN is enabled, DCD becomes available on pin 1 of an 8–pin RJ–45 connector (equivalent to pin 2 of a 10–pin connector).

The operating system does not require DSR in modem–control applications, and since almost all of today's modems have auto–answering capability, the RI signal is generally unnecessary.

Cable length can be up to 61 m (200 ft). Use overall foil/braid shielded multiconductor cable with a capacitance rating of 41 pF/m (12.5 pF/ft) or less. Conductors should be 28 AWG (stranded wire). For lengths less than 61 m (200 ft), higher capacitance cable can be used, as long as the total capacitance (including intermediate connectors and cables) does not exceed 2500 pF.

6–, 8–, and 10– Pin Considerations (EIA 422)

The *forcedcd* option must be set to enable. The 422 RAN defaults to *forcedcd=enable*. The 4–pin RJ–11 connection is not supported. Hardware flow control is not supported for EIA 422 connections. XON/XOFF flow control should be used.

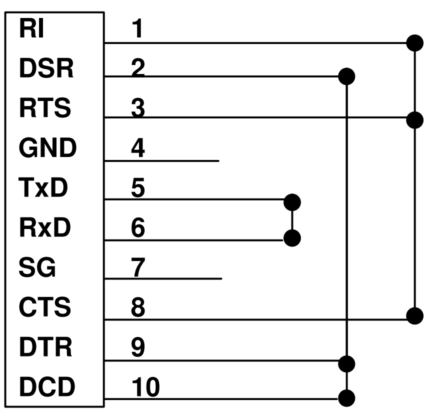
RAN Diagnostics Loopback Plug (EIA 232 only)

The loopback plug is used during the RAN diagnostic asynchronous external test to enable an asynchronous port to transmit and receive data. This plug is not supported for EIA 422.

The loopback plug consists of a *single* 10–pin RJ–45 plug wired as follows:

- Pin 3 connected to pins 1 and 8 (RTS to RI and CTS)
- Pin 5 connected to pin 6 (TxD to RxD)
- Pin 9 connected to pins 2 and 10 (DTR to DSR and DCD).

Refer to the following figure for an illustration of the wiring of the RJ–45 loopback plug.



10-Pin RJ-45
Loopback Plug

Configuring a Defined 128–Port Asynchronous Adapter

This procedure allows you to make a defined 128–port asynchronous controller available for use by the operating system. Use the System Management Interface Tool (SMIT) to build the **mkdev** command, or issue the **mkdev** command directly from the command line.

Prerequisites

- A 128–port asynchronous controller must be defined to the system.
- You must have root user authority.

Configuring a Defined 128–Port Asynchronous Adapter Using SMIT

1. Use the **smit ttyadapters** fast path to access the **128–Port Async Adapters** screen.
2. Select the adapter type you want to make available. The system scans for the device and configures it.
3. Select **Done** or **Cancel**.

Configure a Defined 128–Port Asynchronous Adapter from the Command Line

Enter the **mkdev** command at the command line, specifying the logical name of the already defined 128–port asynchronous adapter in the **–l** *Name* flag. For example, to make a defined MCA 128–port asynchronous adapter `cxma#` available, enter:

```
mkdev -l cxma#
```

To make a defined 128–port ISA adapter `cxia#` available, enter:

```
mkdev -l cxia#
```

To make a defined 128–port PCI adapter `cxpa#` available, enter:

```
mkdev -l cxpa#
```

See the **mkdev** command for a detailed description and optional flags.

Configuring the 128–Port Asynchronous ISA Adapter

Note: ISA adapters are supported only on AIX Version 4.1 systems (or later).

This procedure describes how to use SMIT to configure a 128–port asynchronous EIA 232 ISA adapter.

Prerequisites

- The **devices.isa.cxia128** software package is installed for the ISA 128–port adapter.
- All RAN connections are local to the system. There is no remote RAN attached using a modem.

Procedure

1. Use the **smit mkdev_isa** fast path.
2. Select **cxia** from the list of adapters displayed on the screen.
3. Select the appropriate bus and press Enter.
4. Select **Bus I/O Address** and set the address to the address of the adapter (set by DIP switches on the adapter). Refer to the *128 Port Asynchronous ISA Adapter Installation Guide* for more information about DIP switches. The remainder of the adapter configuration is done automatically when the system displays **cxia Available**.
5. Exit the SMIT interface.

Listing All Defined 128–Port Asynchronous Adapters

This procedure provides a list of the 128–port asynchronous adapters that are defined in the customized database. A defined device can be either available (defined and configured) or unavailable (defined only) to the operating system. To obtain a list of defined 128–port asynchronous adapters, use the System Manager Interface Tool (SMIT) to build the **lsdev** command, or issue the **lsdev** command directly from the command line. The list obtained contains the name of each defined 128–port asynchronous adapter, its status (*available* or *defined*), its location code (which identifies the 128–port asynchronous adapter physical device), and its device description.

Note: The status of a defined and configured device is *available*. The status of a device that is defined but not configured is *defined*.

List Defined Micro Channel 128–Port Asynchronous Adapters Using SMIT

To list all defined 128–port asynchronous adapters regardless of whether they are available, proceed as follows:

1. Use the **smit lsd128psync** fast path. The system scans for the information and displays it.
2. Exit the SMIT interface.

List Defined 128–Port Asynchronous Adapters from the Command Line

Enter the **lsdev** command at the command line as follows:

- To list all defined 128–port asynchronous adapters regardless of whether they are available, enter:

```
lsdev –C –c adapter –t 128psync –H
```

- To list all defined 128–port asynchronous adapters regardless of whether they are available, enter:

```
lsdev –C –c adapter –t cxia128
```

- To list all defined 128–port asynchronous adapters that are available to the operating system, enter:

```
lsdev –C –c adapter –t 128psync –H –S a
```

- To list all defined 128–port asynchronous adapters that are available to the operating system, enter:

```
lsdev –C –c adapter –t 128psync
```

- To list all defined MCA 128–port asynchronous adapters that are *not* available to the operating system, enter:

```
lsdev –C –c adapter –t 128psync –H –S d
```

- To list all defined PCI 128–port asynchronous adapters that are *not* available to the operating system, enter:

```
lsdev –C –c adapter –t 4f111b00
```

See the **lsdev** command for a detailed description and optional flags.

Changing/Showing Characteristics of a 128–Port Asynchronous Adapter

This article describes the general process of changing or showing the characteristics of a 128–port asynchronous adapter. For a detailed discussion of configuring the 128–port asynchronous adapter and RAN, refer to ”Configuring a RAN with SMIT”, on page 6-30.

Prerequisites

- A 128–port asynchronous adapter must be defined and available to the system.
- You must have root user authority.

Change or Show the Characteristics of a 128–Port Asynchronous Adapter Using SMIT

1. Use the **smit ttyadapters** fast path to access the **128–Port Async Adapters** screen.
2. Select the appropriate adapter (**cxma** for Micro Channel, **cxia** for ISA, or **cxpa** for PCI) to change or see the characteristics.
3. In the displayed dialog fields, supply the values or accept the default values.
4. When you have finished, select **Do**.

Note: To make the remote asynchronous nodes detectable by the adapter, you must power–cycle the nodes affected by the configuration changes and select the **Configure Devices Added After IPL** option from the SMIT Devices menu.

Show the Characteristics of a 128–Port Asynchronous Adapter from the Command Line

To display the characteristics of a defined and available 128–port asynchronous adapter, enter the **lsattr** command at the command line, specifying the **–l** *Name* and **–E** flags. For example, to display the characteristics of a 128–port asynchronous adapter, enter:

```
lsdev –Cc adapter
```

Find the name of the defined adapter (for example, `cxma0`) in the displayed output from the **lsdev** command, and enter:

```
lsattr –El cxia0

lsattr –El cxma0

lsattr –El cxpa0
```

See the **lsattr** and **lsdev** commands for a detailed description and optional flags.

Change the Characteristics of a 128–Port Asynchronous Adapter from the Command Line

To change the characteristics of a defined and available128–port asynchronous adapter, enter the **chdev** command at the command line, specifying the **–l** *Name* flag. In addition, specify (multiple instances of) the **–a** *Attribute=Value* flag to change attribute values, or specify the **–f** *File* flag to retrieve attribute values from a file, or specify a combination of both.

For example, to change a defined 128–port Micro Channel asynchronous adapter named `cxma0` to use an 8–wire cable on adapter line 1, enter:

```
chdev –l cxma0 –a line1_cabletype=8
```

To change a 128–port ISA adapter named `cxia1` to use an 8–wire cable on adapter line 1, enter:

```
chdev –l cxia1 –a line1_cabletype=8
```

For an 128–port PCI adapter named `cxpa0` to use an 8–wire cable on adapter line 1, enter:

```
chdev -l cxpa0 -a line1_cabletype=8
```

Specify the **–P** flag if the configuration changes should be applied only to the database. This allows the database to be changed for a 128–port asynchronous adapter that is in use, so that changes take effect the next time the system is started.

See the **chdev** command for a detailed description and other optional flags.

Consult the following list to find out the *Attribute* name for attributes that can be changed in the **–a** *Attribute=Value* flag for the characteristic to change. For example, specify `-a commode_21c2=direct` to change the communication mode of node 2 for the line 2 cable to direct communication transmission. For further information, see ”Configuring a RAN with SMIT”, on page 6-30.

Characteristic	**Attribute Name**
Line 1 speed	line1_speed
Line 1 cable type	line1_cabletype
Line 1 cable type node 1 comm mode	commode_11c1
Line 1 cable type node 2 comm mode	commode_11c2
Line 1 cable type node 3 comm mode	commode_11c3
Line 1 cable type node 4 comm mode	commode_11c4
Line 2 speed	line2_speed
Line 2 cable type	line2_cabletype
Line 2 cable type node 1 comm mode	commode_21c1
Line 2 cable type node 2 comm mode	commode_21c2
Line 2 cable type node 3 comm mode	commode_21c3
Line 2 cable type node 4 comm mode	commode_21c4

Removing the 128–Port Asynchronous Adapter from the Command Line

1. To get information specific to the 128–port asynchronous adapter, enter:

   ```
   lsdev -C | grep cx
   ```

   The output of this command may be similar to the following:

   ```
   cxia0    Available   01-02    IBM 128-Port Async, EIA-232 (ISA)
   ```

   This output indicates that adapter `cxia0` is in slot 2.

2. To remove a 128–port asynchronous adapter and its associated RAN, printer devices, and TTYs, enter:

   ```
   rmdev -l cxia0 -R
   ```

   Running this command maintains the adapter definition in the configuration database.

   To remove a 128–port asynchronous adapter and delete the device definition from the Configuration database, enter:

   ```
   rmdev -l cxia -dR
   ```

   This command also removes the associated printer devices, RANs, and TTYs.

Connecting a RAN to a 128–Port Asynchronous Adapter

This procedure allows you to connect a remote asynchronous node (RAN) to a 128–port asynchronous adapter. The connection between the 128–port adapter and the RAN can be as follows:

- Direct attachment
- Remote attachment using synchronous modems (EIA 232 or EIA 422)

Prerequisites

1. A 128–port asynchronous adapter must be defined and available to the system.
2. The power switch on the RAN must be in the Off (O) position.

Connect First Local RAN Directly to 128–Port Asynchronous Adapter

1. Connect the male end of an IBM PN 43G0937, or identical cable to line 1 of the 128–port adapter.

   See ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7 for cable wiring information.
2. Connect the female end of the cable to the RAN port marked IN.
3. Install the terminator plug on the OUT/T port of the RAN.
4. Plug the AC connector of the power supply (US IBM PN 40H3611, or PN 93H7091 for the enhanced RANs) into a standard grounded wall outlet. Plug the DIN connector into the receptacle labeled +5V/+–12V on the RAN.
5. Turn the RAN's power switch to the On (|) position. The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence, and the seven–segment LED display will display `P1` to indicate that the POST sequence has completed successfully.

   The display will show `AC` to indicate that it is online or active and has received instructions from the adapter.
6. Set the node number. Refer to ”Setting a RAN Node Number”, on page 6-37 for instructions.
7. Reconfigure the adapter by running *cfgmgr –l cxia0*, *cxma0*, or *cxpa0*.

Connect Remote RAN to 128–Port Asynchronous Adapter via Modems

1. Connect the local EIA 232 or EIA 422 synchronous modem to one of the adapter synchronous lines using an EIA 232 or EIA 422 synchronous modem cable.
2. Connect remote EIA 232 or EIA 422 synchronous modem to the RAN port marked IN using the EIA 232 or EIA 422 synchronous modem cable.

   See ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7 for cable wiring information.
3. Install the terminator plug on the OUT/T port of the RAN.
4. Plug the AC connector of the power supply (IBM PN 40H3611, or PN 93H7091 for the enhanced RANs) into a standard grounded wall outlet. Plug the DIN connector into the receptacle labeled +5V/+–12V on the RAN.
5. Turn the RAN's power switch to the On (|) position. The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence, and the seven–segment LED display will eventually display `P1` to indicate that the POST sequence passed.

6. Set the node number. Refer to ”Setting a RAN Node Number”, on page 6-37 for instructions.
7. Reconfigure the adapter by running *cfgmgr –l cxia0*, *cxma0*, or *cxpa0*.

Add Subsequent RAN to a System (Daisy Chain)

Refer to ”128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios”, on page 6-7 for further information.

Note: Each 128–port asynchronous adapter line is limited to no more than four daisy–chained RANs.

1. Using a daisy–chain cable, connect the male end of the cable to the port marked OUT/T on an existing RAN.
2. Connect the female end of the cable to the IN port on the new RAN.
3. If the new RAN is being placed in the middle of a daisy chain, connect the male end of another daisy–chain cable to the OUT/T of new RAN.
4. Connect the female end of the cable to the IN port to the next RAN.
5. Ensure that a terminator plug is on the OUT/T port of the last RAN in the chain.
6. Plug the AC connector of the power supply (IBM PN 40H3611, or PN 93H7091 for the enhanced RANs) into a standard grounded wall outlet. Plug the DIN connector into the receptacle labeled +5V/+–12V on the RAN.
7. Turn the RAN’s power switch to the On (|) position. The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence, and the seven–segment LED display will display `P1` to indicate that the POST sequence passed.
8. Set the node number. Refer to ”Setting a RAN Node Number”, on page 6-37 for instructions.
9. Reconfigure the adapter by running *cfgmgr –l cxia0*, *cxma0*, or *cxpa0*.

Note: The new RAN can be added in the middle or at the end of the daisy chain. However, the RAN must be placed so that it can be assigned a free node number. Each RAN must have a unique node number (*1n–4n*) assigned in ascending order with the lowest number assigned closest to the adapter.

Configuring a RAN with SMIT

The 128–port asynchronous adapter and RAN are automatically configured during the boot process. With locally or directly attached RANs using IBM cabling, it is not necessary to change any of the default parameters supplied by the AIX operating system for the 128–port asynchronous adapter or RAN.

Because the 128–port subsystem is configured during the boot process, administrators should verify that all RANs are correctly set up and cabled before turning on the power of the AIX system. RAN node numbers should be set in an ascending order with the lowest node number being nearest the 128–port asynchronous adapter card.

This section discusses the configuration of the 128–port asynchronous controller for:

- Local or direct RAN Attachment, on page 6-30
- Remote RAN Attachment Using Modems or DSU/CSU, on page 6-30
- Determining Line Speed and Line Cable Type Values for a RAN, on page 6-31
- Determining a RAN's Connection Type and SMIT Node Comm Mode Values, on page 6-32
- Example Node Comm Mode Parameter Determination, on page 6-32
- Sample 128–port Asynchronous Adapter Configuration, on page 6-33

Prerequisites

1. A 128–port asynchronous adapter must be installed, defined, and available.
2. At least one RAN must be connected. See "Connecting a RAN to a 128–Port Asynchronous Adapter", on page 6-28.
3. Set the RAN's node ID. Refer to "Setting a RAN Node Number, on page 6-37.
4. Fill out the "128–Port Asynchronous Adapter Planning Worksheets", on page A-1.

Configure a Local or Direct RAN Attachment

1. Use the **smit ttyadapters** fast path to access the **128–Port Async Adapter** screen.
2. Select the appropriate Micro Channel 128–port adapter (for example, *cxma0*, *cxma1*, *cxma2*), ISA 128–port adapter (for example, *cxia0*, *cxia1*, *cxia2*), or the PCI port adapter *cxpa0* to make the change.
3. In the displayed dialog fields, supply the values or accept the default values.
4. Select **Do** to implement the new configuration.

Configure a Remote RAN Attachment Using Modem or DSU/CSU

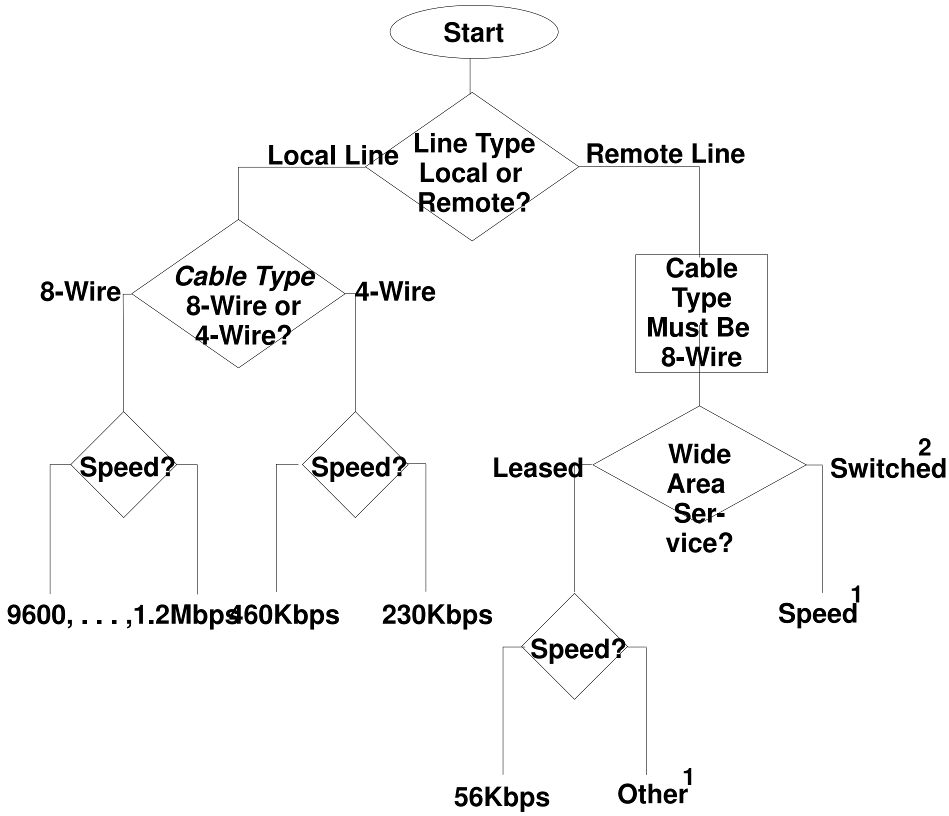
1. Use the **smit ttyadapters** fast path.
2. Select the appropriate 128–port asynchronous adapter from the list displayed on the screen.
3. To view a list of possible values for each field, move the cursor inside the field and select the **List** option.
4. Change the Comm Mode fields to the appropriate values based on the interface type of your modem or DSU/CSU (data service unit/channel service unit). Refer to "Determining a RAN's Connection Type and SMIT Node Comm Mode Values (AIX Version 4)", on page 6-32.

Note: Only one pair of modems or DSU/CSU is allowed per adapter line.

5. Select **Do** to implement the new configuration.

Determining Line Speed and Line Cable Type Values for a RAN

A line from the adapter is considered a remote line whenever there is a modem or DSU/CSU used to connect any pair of RANs within that line. The following flowchart can help you determine the values for the SMIT line speed and cable type parameters.



Line Speed and Cable Type Determination Flowchart

Notes:

1. Set the SMIT line speed parameter based on the line speed provided by the carrier service.
2. For initial configuration, the switched line must be manually established and a connection made between the two modems or DSU/CSU in order to bring up the remote RAN.

Determining a RAN's Connection Type and SMIT Node Comm Mode Values (AIX Version 3.2)

For a line that consists of 4 RANs, the value for the Node Comm Mode field is defined as shown in the following table.

Node Comm Mode Parameter Selection	
Node Comm Mode	**Description**
`direct`	Selected when no modem is used.
`232_modem`	Selected when an EIA 232 modem is used to communicate to the RAN at a remote location.
`422_modem`	Selected when an EIA 422 modem is used to communicate to the RAN at a remote location.

For a line that consists of fewer than 4 RANs, the Node Comm Mode field for a virtual RAN is defined in the following table. A virtual RAN is one that does not physically exist. The sum of the number of physical RANs and virtual RANs is four. For example, if two RANs with node numbers of 1n and 2n are connected to an adapter, 3n and 4n are considered virtual RAN node numbers.

Node Comm Mode for Virtual RAN			
Last Physical Ran Node Number	**Virtual RAN Node Number**	**Node Comm Mode Value for Virtual Node when the Last Physical RAN is Connected with:**	
		EIA 422 Modem	**EIA 232 Modem**
1n	2n 3n 4n	422_modem direct direct	232_modem direct direct
2n	3n 4n	422_modem direct	232_modem direct
3n	4n	422_modem	232_modem

The table can be read as follows:

If the last physical RAN is connected to line 1 (or line 2) and its node number is node 1n, then the virtual RAN node number values are:

2n = `422_modem` (or `232_modem`)

3n = `direct`

4n = `direct`

Determining a RAN's Connection Type and SMIT Node Comm Mode Values (AIX Version 4)

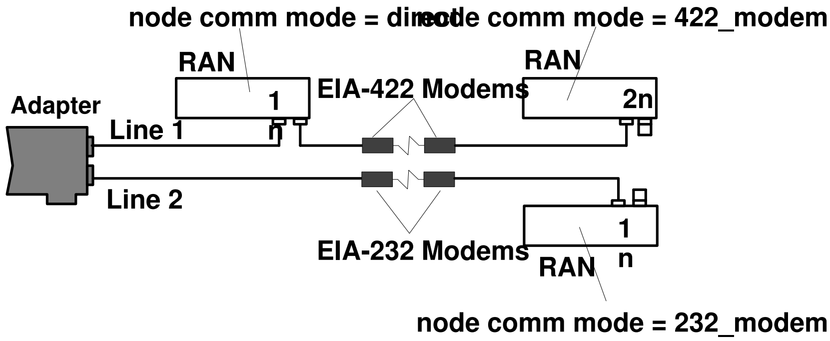
For a line that consists of 4 RANs, the value for the Node Comm Mode field is defined as shown in the following table.

Node Comm Mode Parameter Selection	
Node Comm Mode	**Description**
`direct`	Selected when no modem is used.
`none`	Selected when the designated node is not used or does not exist.
`232_modem`	Selected when an EIA 232 modem is used to communicate to the RAN at a remote location.
`422_modem`	Selected when an EIA 422 modem is used to communicate to the RAN at a remote location.

Example Node Comm Mode Parameter Determination

The following two examples summarize the Node Comm Mode field for the virtual nodes.

See the following figure for an example of three RANs. The first is directly connected to the asynchronous adapter line 1; the second is connected to the first with an EIA 422 modem, and the third is connected to the asynchronous adapter line 2 with an EIA 232 modem.



Example Topology for RAN Node Comm Mode Determination

The values of the Node Comm Mode field for this example are:

Example Node Comm Mode Parameter Determination			
Line	**Node**	**Node Comm Version 3.2**	**Mode Value Version 4**
1	1n	`direct`	`direct`
	2n	`422_modem`	`422_modem`
	3n	`422_modem*`	`none`
	4n	`direct*`	`none`
2	1n	`232_modem`	`232_modem`
	2n	`232_modem*`	`none`
	3n	`direct*`	`none`
	4n	`direct*`	`none`
* – Denotes virtual node definition			

Sample 128–Port Asynchronous Adapter Configurations

This section discusses three sample 128–port asynchronous adapter configurations and their corresponding SMIT configurations.

Example A:

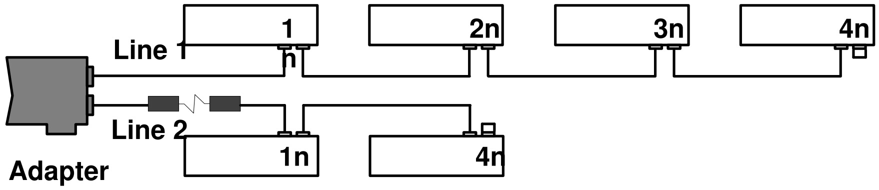
Line 1:

- Has four local– or direct–attached RANs with node numbers 1n through 4n.
- Direct attachment at 1.2 Mbps.

Line 2:

- Has no local RANs and two remote RANs with node numbers 1n and 4n.
- Communication is over a leased–line (speed 14.4 Kbps) using EIA 232 synchronous modems.

See the following figure for a diagram of example A.



Example A: 128-Port Asynchronous Adapter Configuration

The SMIT fields for the above scenario are shown in the following figure.

```
Line 1 Speed                          [1.2M]
Line 1 Cable Type                     [8]
                                   Version 3.2  Version 4
        Node 1 Comm Mode           [direct]     [direct]
        Node 2 Comm Mode           [direct]     [direct]
        Node 3 Comm Mode           [direct]     [direct]
        Node 4 Comm Mode           [direct]     [direct]

Line 2 Speed                          [14.4k]
Line 2 Cable Type                     [8]
                                   Version 3.2  Version 4
        Node 1 Comm Mode           [232_modem]  [232_modem]
        Node 2 Comm Mode           [direct]     [none]
        Node 3 Comm Mode           [direct]     [none]
        Node 4 Comm Mode           [direct]     [none]
```

Example B:

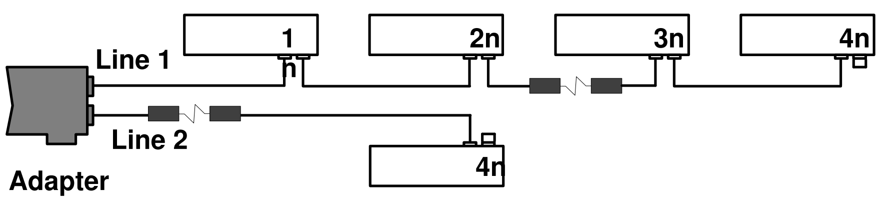
Line 1:

- Has two local and two remote RANs
  - Local RAN node numbers are 1n and 2n.
  - Remote RAN node numbers are 3n and 4n.
- Communication is over a leased–line (speed 56 Kbps) using EIA 232 synchronous modems.

Line 2:

- Has no local RANs and one remote RAN with a node number of 4n.
- Communication is over a leased–line (speed 56 Kbps) using EIA 232 synchronous modems.

See the following figure for a diagram of example B.



Example B: 128-Port Asynchronous Adapter Configuration

The SMIT fields for the above scenario are shown in the following figure.

```
Line 1 Speed                       [57.6K]
Line 1 Cable Type                  [8]
                                 Version 3.2    Version 4
    Node 1 Comm Mode             [direct]       [direct]
    Node 2 Comm Mode             [direct]       [direct]
    Node 3 Comm Mode             [232_modem]    [232_modem]
    Node 4 Comm Mode             [direct]       [direct]

Line 2 Speed                       [57.6k]
Line 2 Cable Type                  [8]
                                 Version 3.2    Version 4
    Node 1 Comm Mode             [232_modem]    [none]
    Node 2 Comm Mode             [direct]       [none]
    Node 3 Comm Mode             [direct]       [none]
    Node 4 Comm Mode             [direct]       [232_modem]
```

Example C:

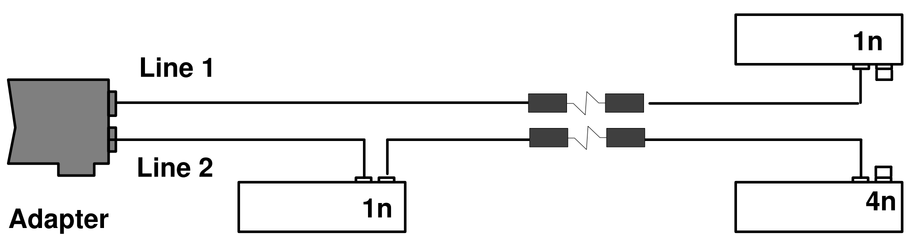
Line 1:

- Has no local RANs and one remote RAN with a node number of 1n.
- Communication is over a leased–line (speed 56 Kbps) using EIA 232 synchronous modems.

Line 2:

- Has one local RAN and one remote RAN.
  - Local RAN node number is 1n.
  - Remote RAN node number is 4n.
- Communication is over a leased–line (speed 56 Kbps) using EIA 232 synchronous modems.

See the following figure for a diagram of example C.



Example C: 128-Port Asynchronous Adapter Configuration

The SMIT fields for the above scenario are shown in the following figure.

```
Line 1 Speed                          [57.6K]
Line 1 Cable Type                     [8]
                                    Version 3.2  Version 4
    Node 1 Comm Mode                [232_modem]  [232_modem]
    Node 2 Comm Mode                [232_modem]  [none]
    Node 3 Comm Mode                [direct]     [none]
    Node 4 Comm Mode                [direct]     [none]

Line 2 Speed                          [57.6k]
Line 2 Cable Type                     [8]
                                    Version 3.2  Version 4
    Node 1 Comm Mode                [direct]     [direct]
    Node 2 Comm Mode                [232_modem]  [none]
    Node 3 Comm Mode                [232_modem]  [none]
    Node 4 Comm Mode                [direct]     [232_modem]
```

Connect Remote RAN to 128–Port Asynchronous Adapter Using Modems

1. Connect the local EIA 232 or EIA 422 synchronous modem to one of the adapter synchronous lines using an EIA 232 or EIA 422 synchronous modem cable.
2. Connect the remote EIA 232 or EIA 422 synchronous modem to the RAN port marked IN using the EIA 232 or EIA 422 synchronous modem cable. See 128–Port Asynchronous Adapter Cable Specifications and Cabling Scenarios, on page 6-7 for cable wiring information.
3. Install the terminator plug on the OUT/T port of the RAN.
4. Plug the AC connector of the power supply (IBM PN 40H3611 for the standard RAN or PN 93H7091 for the enhanced RAN) into a standard grounded wall outlet. Plug the DIN connector into the receptacle labeled +5V/+–12V on the RAN.
5. Turn the RAN's power switch to the on position (|). The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence. The seven–segment LED display should eventually display `P1` to indicate that the POST sequence passed.
6. Set the node number. Refer to Setting a RAN Node Number, on page 6-37 for instructions.
7. Reconfigure the adapter by running *cfgmgr –l cxia0, cxma0,* or *cxpa0.*

Setting a RAN Node Number

The adapter identifies RANs by their node numbers. Each RAN in a daisy chain must have a unique node number (*1n* through *4n*), which must be set during installation. The node numbers must be assigned in ascending order with the lowest number assigned to the RAN closest to the 128–port asynchronous adapter. For direct connect configuration, it is permissible to skip node numbers (to facilitate insertion of additional RANs at a later date), as long as the ascending sequence is maintained.

Set the RAN Node Number

1. Turn the RAN on and wait for the power–on self–test (POST) to complete.
2. When `P1` is displayed on the front panel seven–segment LED display, press the Left Arrow button once. The current node number will be displayed, for example, `1n` for node 1.

Note: Refer to RAN Front Panel and Display Modes, on page 6-38 for an illustration of the RAN front panel.

3. Press the Right Arrow button to advance the node number through the eight possible settings (*1n* through *8n*). *Only 1n* through *4n* are supported, *5n* through *8n* can be displayed but are *not* supported node numbers.
4. When the desired node number is displayed, press the Left Arrow button again to select the number. The display should now read `Pn` (indicating a pass condition). If there was an error, the display will read `En`.

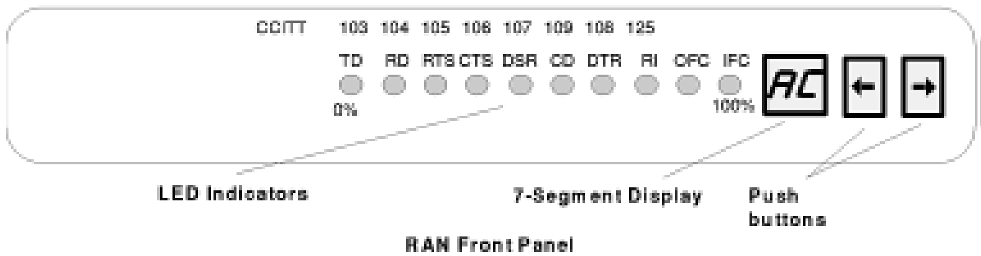
   In the case of duplicate node numbers, the RAN farthest from the host adapter will display `En`, instead of `AC`, when the system is started.

RAN Front Panel and Display Modes

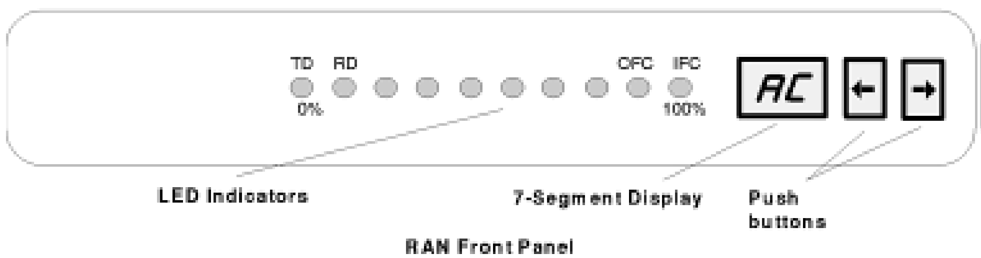
The remote asynchronous node (RAN) is equipped with:

- Ten LED indicators
- Two–digit, seven–segment LED display
- Two push buttons

The following figure shows the front panel of the EIA 232 RAN.



The following figure shows the front panel of the EIA 422 RAN.



The LED indicators can be used to reflect the activity of each of the EIA 232 lines and flow control status for a given line. They can also be set to act as a bar graph to show CPU utilization and the activity level of the EIA 422 synchronous line.

The RAN front panel display has several different display modes as indicated by the two–digit, seven–segment display. Pushing the Right or Left arrow push buttons will cycle the display sequentially through the modes.

The following table describes the RAN display modes.

RAN Display Modes		
Mode	**Mode Name**	**Description**
P1	POST Complete	`P1` appears on the seven–segment display. Power–on self–test is complete, relays are open waiting for connection.
P2	Ping Packet Receive	`P2` appears on the seven–segment display. Indicates that the operating system successfully transmitted a ping packet to RAN. The ping packet contains configuration information used by the RAN (for example baud rate, type of interface).
P3	Transmit Configuratio n Packet	`P3` is *not* displayed on the seven–segment display. The RAN transmits a packet that contains information about the RAN's physical characteristics. The operating system uses this information to determine which download image to send to the RAN. The RAN does not receive confirmation that the operating system has received the packet.

P4	Image Receive	`P4` appears on the seven–segment display. Download image is being received from the host. The RAN will normally stay at `P4` for a length of time, depending on the synchronous baud rate being used.
AC	Activity	`AC` appears on the seven–segment display. The 10 LEDs turn on sequentially from left to right. The speed of this ”chase light” display increases with the overall activity level of the RAN.
00–15	Line Monitor	`00–15` appears on the seven–segment display. Modes `00` through `15` correspond to channels 0 through 15. Press the right or left push buttons until the desired channel number appears in the seven–segment display. The LEDs act as line monitor for the selected channel. The first eight LED indicators show the activity of each of the eight EIA 232 signals (TD, RD, RTS, CTS, DSR, DCD, DTR, and RI). The last two LED indicators show when output flow control (OFC) and input flow control (IFC) are active.
En	Error Node	`En` appears on the seven–segment display. Indicates that a valid ping packet was received but the node number in EEPROM is incorrect.
PC	Packet Count	`PC` appears on the seven–segment display. The 10 LEDs show a binary representation of the total number of packets transmitted or received. Pressing both push buttons simultaneously resets the count to 0.
EC	Error Count	`EC` appears on the seven–segment display. The 10 LEDs show a binary representation of the total number of errors counted in the data. Pressing both push buttons simultaneously resets the count to 0.
		When the EC counter is rapidly increasing, this is often an indicator that: 1. The line quality between RANs and/or the adapter is poor. 2. The line between RANs and/or the adapter is open.
PU	Processor Utilization	`PU` appears on the seven–segment display. The 10 LEDs become a bar graph indicating the percentage (0–100%) of the time the RAN microprocessor is being used.
LU	Line Utilization	`LU` appears on the seven–segment display. The 10 LEDs become a bar graph indicating the percentage (0–100%) of the time that the synchronous communications line is being used.
1n, 2n,...,8n *	Node Number	The seven–segment display shows the node number of the RAN.
Ed	POST Error Diagnostics	RAN Error. RAN is Defective. RAN will stay isolated from host.

Note: * Only node numbers *1n* through *4n* are valid. Node numbers *5n* through *8n* are not supported.

Removing or Replacing a RAN

This procedure allows you to remove or replace a remote asynchronous node (RAN) from your system.

If a particular RAN is turned off or removed from the daisy chain, the 16 ports on that RAN become unavailable to the system, leaving the rest of the system unaffected.

Because the RANs IN and OUT/T ports are of opposite gender, a RAN can be removed from the middle of a daisy chain by plugging the cables together so that the chain remains unbroken.

Prerequisites

1. Make sure the power switch on all RANs is in the Off (O) position.
2. Unplug the AC connector of the power supply from the wall outlet. Unplug the DIN connector from the receptacle on the RAN.

Remove a RAN

1. Disconnect the cables from the RAN.
2. Reconnect other RANs (if any) to complete the daisy chain.
3. Ensure that the terminator plugs on the OUT/T port of the last RAN.
4. Reconfigure the adapter using *cfgmgr –l* (*adapter name*).

Replace a RAN

1. Disconnect the cables from the RAN to be removed.
2. Connect the new RAN. Refer to ”Connecting a RAN to a 128–Port Asynchronous Adapter”, on page 6-28.
3. Ensure that the terminator plug is on the OUT/T port of the last RAN.
4. Turn the RAN’s power switch to the On (|) position. The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence, and the seven–segment LED display will display `P1` to indicate that the POST sequence has completed successfully.
5. Set the node number of the new RAN. Refer to ”Setting a RAN Node Number”, on page 6-37.
6. Reconfigure the adapter using *cfgmgr –l* (*adapter name*).

Configuring an Asynchronous Terminal Connected to a RAN

This procedure allows you to define and configure a tty device connected to a 128–port asynchronous adapter RAN.

Prerequisites

1. You must have root user authority.
2. A 128–port asynchronous adapter must be installed, defined, and available.
3. At least one RAN must be connected. See ”Connecting a RAN to a 128–Port Asynchronous Adapter”, on page 6-28.
4. Set the RAN node ID. Refer to ”Setting a RAN Node Number”, on page 6-37.

Note: Correct cabling must be used so the asynchronous terminal operates correctly on the remote asynchronous node. RJ–45 10–pin cable is recommended to support the full EIA 232 specifications.

Procedure

1. Use the **smit tty** fast path to access the **TTY** menu.
2. Select **Add a TTY**.
3. Select **tty rs232 Asynchronous Terminal** or EIA 422 as appropriate for the RAN type.
4. Make a selection from the available RANs displayed on the screen. If no RANs are displayed or if they are shown in a **defined** state, check the RAN configuration, cabling, and setup again.

   For this example, the following selection was made:

   ```
   sa4 Available 00-03-21 16-Port RAN EIA 232 for 128-Port Adapter
   ```
5. In the displayed dialog fields, you can add or change the field values as needed.
6. When you have finished, select **Do**.

Configuring a Printer or Plotter Connected to a RAN

This procedure allows you to define and configure a printer or plotter attached to a 128–port asynchronous adapter RAN.

Prerequisites

1. You must have root user authority.
2. A 128–port asynchronous adapter must be installed, defined, and available.
3. At least one RAN must be connected. See ”Connecting a RAN to a 128–Port Asynchronous Adapter”, on page 6-28.
4. Set the RAN’s node ID. Refer to ”Setting a RAN Node Number”, on page 6-37.

Note: To work properly with the Print Spooler Subsystem, serial printers and plotters should be attached to the RAN using 8– or 10–pin RJ–45 cabling. See ”RAN 16–Port Device Cables”, on page 6-16 for proper cabling information.

Procedure

1. To add a printer/plotter device to a 128–port RAN, use the **smit pdp** fast path to access the **Printer/Plotter Devices** menu.
2. Select **Add a Printer/Plotter**.
3. Select the appropriate printer device from the list of printer and plotter types shown on the screen and press the Enter key. For this example, the following selection was made:

   ```
   osp     Other serial printer
   ```
4. Select **rs232** or RS–422, as appropriate for the RAN type.
5. Make a selection from the available RANs displayed on the screen. If no RANs are displayed or if they are shown in a **defined** state, check the RAN configuration, cabling, and setup again. For this example, the following selection was made:

   ```
   sa4 Available 00-03-21 16-Port RAN EIA-232 for 128-Port Adapter
   ```
6. In the displayed dialog fields, you can add or change the desired printer/plotter device attributes.
7. When you have finished, select **Do**.

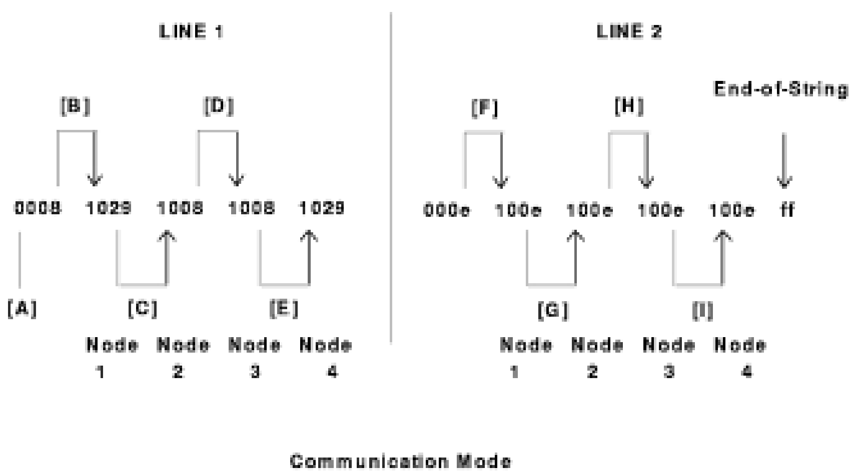
The 128–Port Configuration String

The configuration string of the 128–port asynchronous adapter is a multi–byte, hexadecimal character field that represents the communication modes used between the adapter and the RAN. The values shown in this string should be examined whenever the connection modes are in question.

In the following example, the **lsattr** command is used to view the current 128–port adapter settings.

```
#lsattr –Elcxma0

bus_io_addr              0x108              Bus memory address       False
bus_mem_addr             0x380000           Bus I/O address          False
config              00081029100810081029000e100e100e100e100eff N/A   True
line1_speed              57.6K              Line Speed               True
line1_cabletype          8                  Cable Type               True
commode_l1c1             direct             Comm Mode                True
commode_l1c2             232_modem          Comm Mode                True
commode_l1c3             direct             Comm Mode                True
commode_l1c4             direct             Comm Mode                True
line2_speed              1.2M               Line Speed               True
line2_cabletype          8                  Cable Type               True
commode_l2c1             direct             Comm Mode                True
commode_l2c2             direct             Comm Mode                True
commode_l2c3             direct             Comm Mode                True
commode_l2c4             direct             Comm Mode                True
```

In the following figure, the first byte [A] of the config string starts with 00. The next byte [B] 08H shows the communication mode from LINE 1 of the adapter to the first 16 (10H) ports of RAN node 1 as being configured with a line speed of 57.6K internal clocking. See the figure,



Configuration String Data.

Mode	Bit Rate	Clocking
00 (00H)	115K	8-wire, internal clock
01 (01H)	230K	4-wire, self clocked
02 (02H)	460K	4-wire, self clocked
03 (03H)	2400	8-wire, internal clock
04 (04H)	4800	8-wire, internal clock
05 (05H)	9600	8-wire, internal clock
06 (06H)	19.2 K	8-wire, internal clock
07 (07H)	38.4 K	8-wire, internal clock
08 (08H)	57.6 K	8-wire, internal clock
09 (09H)	76.8 K	8-wire, internal clock
10 (0AH)	115 K	8-wire, internal clock
11 (0BH)	230 K	8-wire, internal clock
12 (0CH)	460 K	8-wire, internal clock
13 (0DH)	920 K	8-wire, internal clock
14 (0EH)	1.2 M	8-wire, internal clock
15 (0FH)	2400	8-wire, external clock (EIA 422)
16 (10H)	4800	8-wire, external clock (EIA 422)
17 (11H)	9600	8-wire, external clock (EIA 422)
18 (12H)	19.2 K	8-wire, external clock (EIA 422)
19 (13H)	38.4 K	8-wire, external clock (EIA 422)
20 (14H)	57.6 K	8-wire, external clock (EIA 422)
21 (15H)	76.8 K	8-wire, external clock (EIA 422)
22 (16H)	115 K	8-wire, external clock (EIA 422)
23 (17H)	230 K	8-wire, external clock (EIA 422)
24 (18H)	460 K	8-wire, external clock (EIA 422)
25 (19H)	920 K	8-wire, external clock (EIA 422)
26 (1AH)	1.2 M	8-wire, external clock (EIA 422)
33 (21H)	14000	8-wire, external clock (EIA 422)
35 (23H)	2400	8-wire, external clock (EIA 232)
36 (24H)	4800	8-wire, external clock (EIA 232)
37 (25H)	9600	8-wire, external clock (EIA 232)
38 (26H)	14000	8-wire, external clock (EIA 232)
39 (27H)	19.2 K	8-wire, external clock (EIA 232)
40 (28H)	38.4 K	8-wire, external clock (EIA 232)
41 (29H)	57.6 K	8-wire, external clock (EIA 232)
42 (2AH)	64000	8-wire, external clock (EIA 232)
43 (2BH)	76.8 K	8-wire, external clock (EIA 232)

Configuration String Data

The communication mode [C] from RAN node 1 to RAN node 2 is 29H (57.6K, external EIA 232 clocking). The communication mode [D] from RAN node 2 to RAN node 3 is 08H (57.6K, internal clocking). The communication mode [E] from RAN node 3 to RAN node 4 is 08H (57.6K, internal clocking).

LINE 2 entries of the config string also start with 00 from the adapter. The communication mode shown as [F] from the adapter to the first 16 (10H) ports of RAN node 1 is 0EH (1.2M, internal clocking). [G], [H], and [I] entries contain the same information as [F]. A final entry of **ff** indicates the end of this string.

Running the Remote Asynchronous Node Diagnostics

This procedure allows you to run diagnostics on your remote asynchronous node (RAN) in one of two modes:

Front Panel Mode	Diagnostics are run from the front panel of the RAN.
Video Mode	Diagnostics are run from a terminal attached to the RAN.

The diagnostics routines can test the following aspects of the RAN:

- Front panel
- Memory
- Memory/direct memory access (DMA)
- Asynchronous internal
- Asynchronous external
- Sync EIA 422/DMA
- Sync EIA 232/DMA
- EEPROM
- Watchdog timer

Run Diagnostics from the Front Panel

1. Disconnect the RAN from the adapter and other RANs before running diagnostic tests. Make sure the RAN is terminated.

   Note: Refer to ”Removing or Replacing a RAN”, on page 6-40

2. Turn the power switch to the Off (O) position.
3. Press and hold the Right Arrow button while turning the power switch to the On (|) position.
4. The front panel display window shows `P0`. Then, starting with the leftmost LED and proceeding to the right, all LEDs must turn on one at a time until all LEDs are on. All the LEDs then turn off. At this point, release the Right Arrow button. The number 1 with a blinking ”.” in the lower–right corner of the window will be displayed. The `1` refers to TEST 1, the blinking ”.” in the lower–right corner of the window indicates that TEST 1 is currently selected.
5. Use the right arrow button to cycle through the possible test numbers 1–9.
6. When the desired test number is displayed in the window, press the Left Arrow button to start the test. A solid ”.” will display in the lower–left corner of the window to indicate the test is currently running.
7. To stop a test, press the Left Arrow button once. The status of the test will display in the window. A `Pn` indicates the test passed successfully. An `Fn` indicates a failure of test *n*.
8. To exit diagnostic test mode and return to the initialized state (P1), run TEST 9, the Watchdog timer test. This test ends by resetting the RAN and running the POST sequence.

Run Diagnostics from a Terminal

1. Disconnect the RAN from the host adapter and other RANs before running diagnostic tests. Make sure the RAN is terminated properly using wrap plugs.
2. Turn the RAN off by switching the power switch to the Off (O) position.
3. Connect a serial terminal to either port 0 or port 15. Set the terminal's communications parameters to VT100 emulation, 9600, 8 data bits, no parity, and 1 stop bit.
4. Turn the RAN's power switch to the On (|) position. The lights on the front panel should flash as the RAN executes its power–on self–test (POST) sequence, and the seven–segment LED display will display `P1` to indicate that the POST sequence passed.
5. While `P1` is displayed on the front panel, press the letter V (either uppercase or lowercase) on your terminal.

   See this illustration for an example of the Diagnostics Menu displayed on the screen of the serial terminal.

```
Diagnostics                              EPROM Version: 0226
 Test Name                 Passes              ERRORS

A) ALL TESTS.
1) Front panel test.
2) Memory test.
3) Memory/DMA test.
4) Asynchronous internal test.
5) Asynchronous external test.
6) Sync EIA 422/DMA test.
7) Sync EIA 232/DMA test.
8) EEPROM test.
9) Watchdog timer test.
— — — — — — — — — — — — — — — — — — — — — — — — — — — —
Operator Command:
<Y>/<N> - Stop on 1st error = Y
<B>ell - Bell rings for each error = ON
<D>ump - Dumps 256 bytes of the selected memory
<ESC>ape - Stops the test in progress.
<S>et - Sets the node number in EEPROM.
<T>oggle - External/Internal mode for EIA 422 & EIA 232 tests = INTERNAL.

Enter the test selection or command:
```

RAN Serial Terminal Diagnostics Menu

Note: Some terminals (including the Wyse 60) will not transmit data if their clear to send (CTS) line is LOW. They will only transmit if CTS is HIGH or floating. When using a fully configured null modem cable, the terminal's CTS line is connected to the RAN's request to send (RTS) line, which is held LOW by the RAN until `AC` is displayed. This can prevent the terminal from communicating with the RAN when attempting to run diagnostics.

   If you are using such a terminal, use a cable that does not have CTS connected at the terminal's end.

6. Enter test selection or command.

Diagnostics Menu Options Descriptions

1–9 Runs individual tests as shown on the menu. Refer to "Diagnostic Test Descriptions", on page 6-48 for further information.

A Runs all tests except test 9 (Watchdog timer test). The screen will display the number of passes run for each test. The test will halt if an error is detected if `Stop on 1st` is set to Y.

N Runs the test continuously, keeping count of the errors detected. This must be selected prior to running the test.

Esc key Cancels test execution at any time during execution. The current pass of the test will be completed and control will be returned to the terminal.

B Toggles the bell option. When set to `ON`, the terminal beeps each time an error is encountered.

D Allows you to dump 256 bytes of the RAN's memory to the screen beginning with a specified address (you will be prompted for the starting segment address).

S Allows you to view or change the RAN's node number. The node number is written in three different EEPROM locations to ensure correctness of the number.

T Toggles the loopback mode for the EIA 422 and EIA 232 tests (tests 6 and 7) between INTERNAL and EXTERNAL. When set to INTERNAL, the signals are looped back internally by the relays in the RAN. When set to EXTERNAL, a standard 8–wire daisy–chain cable must be installed between the RAN's IN port and its OUT/T port.

Diagnostic Test Descriptions

TEST 1: Front Panel Test This test alternately turns on and off at one–second intervals all 10 LED indicators and all segments (plus the decimal point) of both seven–segment displays to verify proper operation of the front panel indicators. Since the indicators are write–only, the operator must visually verify success or failure of the test.

TEST 2: Memory Test Each pass of this test performs a pattern test and an address tag test of DRAM. The byte pattern is incremented for each pass and is displayed on LEDs 0–7. The pattern is written to 32KB, beginning at 08000H. The pattern is written again to 32KB beginning at 10000H. The two 32KB blocks are compared to determine the pass or fail status of the test.

The address tag test writes 32KB beginning at address 08000H. Address 08000H will be written with 0H, address 08001h will be written with 1h, etc. This same tag pattern will be written beginning at address 10000H. The two 32KB blocks are compared to determine pass or fail status.

TEST 3: Memory/DMA Test The DMA tests DMA transfers from one memory location to another. This test may be run without being connected to the RAN. The byte pattern is incremented for each pass and is displayed on LEDs 0–7. The pattern is written to 32KB beginning at 08000H. DMA0 is used to move the 32 KB from 08000H to 10000H. When the move is complete, DMA0 interrupts the CPU and the two 32KB blocks are compared to determine pass or fail status. If the DMA transfer is not completed within 2 seconds, a time–out error causes the test to fail.

TEST 4: Asynchronous Internal Test

This test checks the asynchronous ports hardware. The test puts the port in local loopback mode. The four output signals (DTR, RTS, OUT1, and OUT2) are looped back within the UART chip to the four input signal lines (CTS, DSR, DCD, and RI). The following steps are completed:

- The output and input signals are checked for high and low conditions.
- The ports are initialized to 9600 baud, 8 data bits, 1 stop bit, and no parity.
- Data is transmitted and received at the same port.
- Received data is compared to the transmitted data.
- An interrupt is generated and checked for each port.

This test functions differently depending on whether it is run in video mode or front panel mode.

- **Front Panel Mode**

  In front panel mode, all 16 asynchronous ports of the RAN are checked in each pass. To begin the test, press the Left Arrow button while the number `4` is displayed in the seven–segment LED display. The left–hand digit of the seven–segment display shows a pass count. The first five LEDs (TD, RD, RTS, CTS, and DSR) give a binary representation of the number of the port under test (TD=LSB; DSR=MSB). The test will be repeated until a failure occurs, or until the Left Arrow button is pressed. At this point either `P4` (pass) or `F4` (fail) will be displayed. Press the Left Arrow button now to restart the test or the Right Arrow to advance to TEST 5.

- **Video Mode**

  In video mode, 15 asynchronous ports (1–15 if the terminal is connected to Port 0, or 0–14 if the terminal is connected to Port 15) are tested in each pass. In addition to the number of passes and errors, the display shows the number of each port under test, and its pass/fail status. A failure will also result in a beep from the terminal.

TEST 5: Asynchronous External Test

This test is used to check the asynchronous ports and their associated driver and receiver circuits. Loopback plugs are required. See ”RAN Diagnostics Loopback Plug”, on page 0 for more information. These plugs enable one asynchronous port to transmit and receive data. The following steps are completed:

- The two control signals (DTR and RTS) are used to test the four input signals (DSR, DCD, CTS, and RI) on the same port.
- The port is tested for transmit and receive data:
  - DTR is looped back to DCD and DSR.
  - RTS is looped back to CTS and RI.
  - The port is initialized to 9600 baud, 8 data bits, 1 stop bit, and no parity.
  - Data is transmitted out of the port and received by the same port.
  - Received data is compared to the transmitted data.

This test functions differently depending on whether it is run in video mode or front panel mode.

- **Front Panel Mode**

  This mode checks one port at a time. Only one loopback plug is needed. Pressing the Left Arrow button once will display the current port number to be tested. Pressing the Left Arrow button repeatedly causes the port number to be incremented through all 16 ports. The test will be repeated until a failure occurs or until the Left Arrow button is pressed again. At this point, either `P5` (pass) or `F5` (fail) will be displayed. Once the test is halted, pressing the Left Arrow button again increments the port number as before. To advance to TEST 6, press the Right Arrow button while `P5` or `F5` is shown.

- **Video Mode**

  In video mode, 15 asynchronous ports are tested in each pass. For repeated testing, 15 loopback plugs are necessary. In addition to the number of passes and errors, the display shows the number of each port under test, and its pass/fail status (a failure will beep the terminal).

  To test a single port in video mode, plug the loopback plug in the port to be tested and set `Stop on first error` to `YES`. If the port passes, 14 errors will be counted (for 14 ports with no loopback plugs). If the port fails, 15 errors will be reported.

For repeated testing of a single port, front panel mode is recommended.

TEST 6: Sync EIA 422/DMA Test

This test uses DMA0, DMA1, and the SCC 8530 sync port to transmit, receive, and verify data using the EIA–422 circuitry. The byte pattern is incremented for each pass and is displayed on LEDs 0–7. The pattern is written to a block of memory beginning at address 08000H.

DMA0 (Rx) and the sync port are used to move the block of data from 08000H to 10000H. The test checks three modes of synchronous data transmission: 8–wire, 1.2 Mbps (32K–byte blocks); 4–wire NRZI, 230 Kbps (8K–byte blocks); and 4–wire FMO, 460 Kbps (16–byte blocks).

When the move is complete, the sync port generates an interrupt and the two blocks are compared to determine pass or fail status. If the DMA transfer using the sync port is not completed within 2 seconds, a time–out error causes the test to fail.

- **Front Panel Mode**

  While `6` is shown in the seven–segment LED display, press the Left Arrow button to begin the test. The left–hand digit will now show the number of sync port receive interrupts and the right–hand digit displays the number of transmit interrupts. This test always runs in internal mode (*no* loopback plug is required) and checks the port in three modes of synchronous transmission. The test halts when an error is detected (`F6` will be displayed) or when the Left Arrow button is pressed again (`P6` will be displayed). At this point pressing the Left Arrow button again restarts the test; pressing the Right Arrow button advances to TEST 7.

- **Video Mode**

  This test can be run with internal or external loopback mode. The current mode is indicated at the end of the `<T>oggle` menu option line. To select between internal and external mode, press T. In internal mode, the port is tested in three modes of synchronous transmission. In external mode, only one transmission method is checked (8–wire, 1.2 Mbps, 32 Kbyte blocks). External mode requires a standard 8–wire daisy–chain cable to be installed between the concentrator's IN port and its OUT/T port.

TEST 7: Sync EIA 232/DMA Test

This test uses DMA0, DMA1, and the SCC 8530 sync port to transmit, receive, and verify data using the EIA–232 circuitry. The byte pattern is incremented for each pass and is displayed on LEDs 0–7. The pattern is written to 512 bytes beginning at address 08000H. DMA1 (Tx), DMA0 (Rx), and the sync port are used to move the 512 bytes of data from 08000H to 10000H at an SDLC rate of 19200 baud. When the move is complete, the sync port generates an interrupt, and the two 512–byte blocks are compared to determine pass or fail status.

If the DMA transfer using the sync port is not completed within 2 seconds, a time–out error causes the test to fail. An audible relay click will be heard upon entering and exiting this test. In video mode, this test can be run with either internal or external loopback by pressing T on the terminal. The external mode requires a standard 8–wire daisy–chain cable to be installed between the RAN's IN port and OUT/T port.

TEST 8: EEPROM Test

This test generates a checksum of the EPROM contents and compares it with the checksum stored in the EEPROM. If the checksums match, the test passes. This does not write to the EEPROM (EEPROM write operations only occur when a new node number is set).

TEST 9: Watchdog Timer Test

This test checks out the watchdog timer. This is a hardware feature that is used to ensure system reliability. When the watchdog timer is loaded and enabled, the timer begins counting down. It is up to the system to keep reloading the timer to prevent it from expiring. If the timer expires, the hardware forces the CPU into a reset state and the POST sequence is started (exactly as if the concentrator is turned off and then on again). Normal test execution allows the timer to expire and force POST execution. The test fails after 1 second if the timer has not expired.

Monitoring the 128–Port Asynchronous Subsystem (AIX Version 4.1 Systems or Later)

The mon–cxma program provides the following capabilities:

- Displays adapter line speeds and wire schemes.
- Displays RAN status, node numbers, and attachment topology.
- Realtime monitoring of synchronous packets transmitted to and received from the RAN.
- Shows whether synchronous packets are being received from the RAN.
- Displays port EIA 232 signal activity through a simulated RAN front panel.
- Displays port input, output, and control modes.
- Displays port and RAN input and output flow control activity.
- Displays RAN status condition as active (AC) or down (DN).
- Allows loopback testing of individual ports on attached RAN.
- Displays 128–port adapter VPD (Vital Product Data).
- Allows screen contents to be dumped to a log file.

Prerequisite Information

- Root user authority is required.
- AIX V4.1 or later is required (verify using the **oslevel** command).
- 128–port asynchronous adapter software must be installed on the system.
- At least one128–port asynchronous adapter must be installed, configured, and available on the system.

Running the Monitor from SMIT

Use the following procedure to execute the mon–cxma program from SMIT.

Procedure

1. Use the **smit ttyadapters** fast path to access the **128–Port Asynchronous Adapter** menu.
2. Select **Monitor Async Adapters.**
3. Select the appropriate 128–port asynchronous adapter from the SMIT panel. In this example, three asynchronous adapters are installed in the system; one of which is located on a different bus.

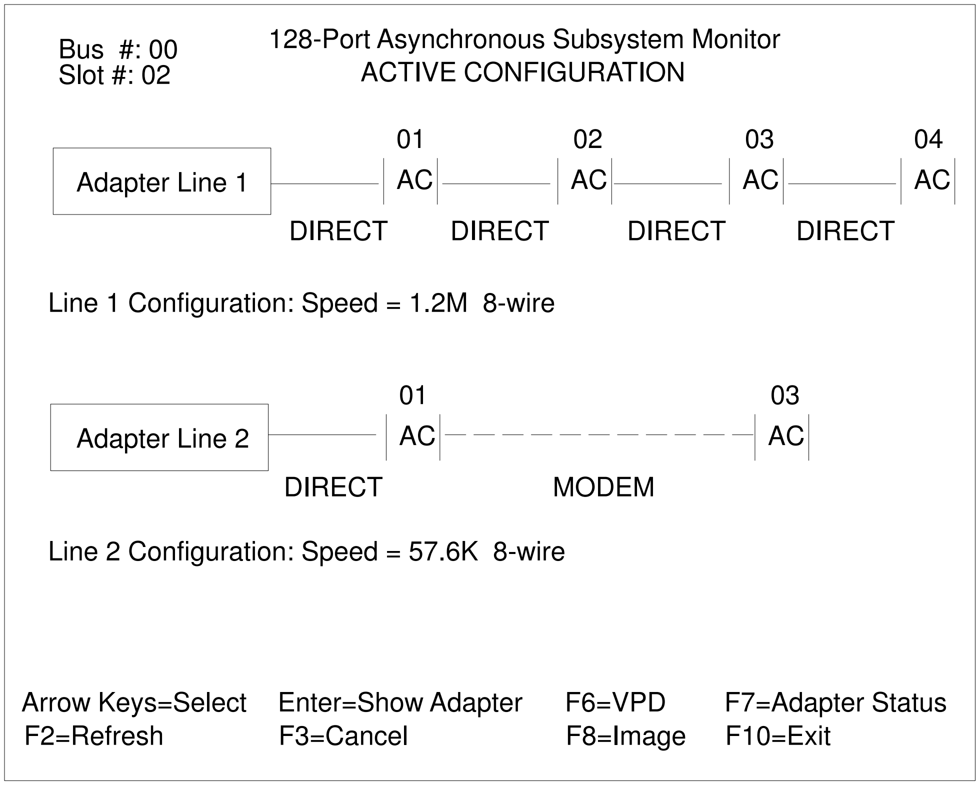
```
                128-Port Asynchronous Subsystem Monitor



        Bus 0, Slot 2 Asynchronous adapter
        Bus 0, Slot 7 Asynchronous adapter
        Bus 1, Slot 8 Asynchronous adapter








     Up/Down Arrow Keys        Enter=Do          F10=Exit
     F2=Refresh                F3=Cancel
```

4. For this example, select **Bus 0, Slot 2 Asynchronous adapter**. This selection takes you to the **ACTIVE CONFIGURATION** screen.
5. The **ACTIVE CONFIGURATION** screen is used as the main panel for the monitor program and displays the following 128–port subsystem information:
   - Bus and slot numbers.
   - Line topology.
   - RAN node numbers.
   - Current RAN status.

   You can also view Vital Product Data (VPD) and adapter status from this panel by pressing the appropriate function key (F6 or F7). A screen image can be saved to the monitor log file, **/tmp/mon–cxma.log**, by pressing the F8 (Image) key.

   Vital Product Data (VPD) is a collection of device information such as part numbers, serial numbers, and engineering change levels. This information is used by service personnel to determine compatibility of hardware components. The data is collected from the 128–port asynchronous subsystem and automatically added to the system configuration. Service representatives may find it necessary to view this information before making upgrade or replacement decisions concerning the 128–port subsystem.

   The following figure illustrates the main panel for the monitor program.



On this screen, RANs are represented by [AC] if active, or (–) if inactive (turned off, defective). In this example, `Adapter Line 1` has four directly attached RANs which are daisy–chained directly off the 128–port adapter. They are all active and assigned node numbers sequentially from `01` to `04`.

`Adapter Line 2` has two RANs attached. The first RAN, `node 01`, is directly attached to the 128–port adapter while the second, `node 03`, is connected to the first using EIA 232 modems.

You can move between the Adapter Lines or RANs by using the up and down arrow keys. If you press Enter on an Adapter Line, the **ADAPTER CONFIGURATION** screen for that adapter is displayed.

The **ADAPTER CONFIGURATION** screen shows the adapter's line configuration as well as the configuration string stored in memory during system startup. This string is used during initialization to determine line speeds and connection types between the 128–port adapter and RAN. It is useful for diagnosing configuration problems.

6. To view a port's activity, use the arrow keys to select the desired, active RAN (represented by [AC]) and press the Enter key. The software displays a graphic representation of the RAN front panel as shown in the figure below:

```
Bus #: 00          128-Port Asynchronous Subsystem Monitor
Slot #: 01                 Status for Device /dev/tty1

  Synchronous Packets Transmitted: /
  Synchronous Packets Received:      /


              Line 1, RAN 1, Port 0

   TD   RD   RTS   CTS   DSR   CD   DTR   RI   OFC   IFC   AC
   —    —                —     —    —

    Input Modes  :BRKINT:IXANY:IXOFF:IXONA:
  Output Modes  :XCASE:ONLCR:TAB2:BS1:
 Control Modes  :38.4K Baud:8 Bits:1 Stop Bits:No Parity:


Left Arrow Key=Next Port   Right Arrow Key=Previous Port  F12=Loopback Test
F2=Refresh             F3=Cancel           F8=Image              F10=Exit
```

This screen is updated at half second intervals and shows the current status of the selected port.

The box in the center of the screen shows the state of the EIA 232 signals TD, RD, CTS, DSR, CD, DTR, and RI, plus input and output flow control status (IFC and OFC) for the selected port.

An inactive signal is indicated by a dash (–) under its name. An active signal is indicated by a block under its name. If data is currently being transmitted or received for the selected port, a blinking "X" will appear under the signal name (TD or RD, respectively).

The status of the RAN is represented at the right–hand end of the box with [AC] if the connection to the host adapter is good.

Beneath the line status box are the input, output and control modes for that port. These are terminal attributes with descriptions.

INPUT MODES

IGNBRK Ignore Break.

BRKINT Interrupt on Break.

IGNPAR Ignore parity errors.

PARMRK Mark parity errors.

INPCK Input parity check.

ISTRIP Strip input characters.

ITOSS Toss IXANY characters.

IXON Enable start/stop output.

IXANY Restart output on any character.

IXOFF Enable start/stop input.

IXONA	Enable start/stop output Auxiliary.
OUTPUT MODES	
XCASE	Canonical upper/lower display.
OLCUC	Map lower case to upper.
ONCLR	Map NL to CR/NL.
OCRNL	Map CR to NL.
ONOCR	No CR output at column 0.
ONLRET	NL performs CR function.
OFILL	Use fill characters for delay.
OFDEL	Fill character id DEL; otherwise NUL.
NLDLY	NL delay is selected.
CR1	Carriage Return delay type 1.
CR2	Carriage Return delay type 2.
CR3	Carriage Return delay type 3.
TAB1	Tab delay type 1.
TAB2	Tab delay type 2.
TAB3	Tab delay type 3.
BS1	Backspace delay.
VT1	Vertical tab delay.
FF1	Form feed delay.
CONTROL MODES	
Baud Rate	50,75,110,134,150,200,300,600,1200,1800,2400,4800,9600,9200,38400. If fastbaud is enabled, baud rates of 50, 75, 110, and 200 are translated to 57600, 75600, 115200, and 230000, respectively.
Char Bits	5, 6, 7, 8 or none.
Stop Bits	2, 1 or none.
Parity	Enabled, odd or none.

fastbaud	Fast baud rates.
	To examine the fastbaud setting, use the command:
	`/usr/lbin/tty/stty-cxma –a tty1`
	Fastbaud is disabled if it is preceded by a minus sign. To enable fastbaud, use the command:
	`/usr/lbin/tty/stty-cxma fastbaud tty1`
	The baud rate of the tty device must be 50, 75, 110, or 200 for fastbaud to take effect.

Up–arrow	Increment RAN number. If the currently selected RAN is the last one on the synchronous line, then the number wraps around to first RAN.
Down–arrow	Decrement RAN number. If the currently selected RAN is the first one on the synchronous line, then the number wraps around to the last RAN on the line.
Right–arrow,l	Increment port number. If the currently selected port is the last one on the RAN, then the port number wraps around to the first port.

Left–arrow,h Decrement port number. If the currently selected port is #1, then the port number wraps around to the last port on the RAN.

0–9, a–f Fastpath port selection. Press 0–9 for ports 0–9, respectively, and a–f for ports 10–15, respectively.

F12 Run loop back test on the selected port. The loop back test transmits 128 bytes and attempts to read them back through the same port. A loop back plug is required.

F8 Dumps the current screen contents to the log file. The default log filename is **/tmp/dpalog**. A different file may be specified by using the **–l** *LogFile* command line option. Refer to the **mon–cxma** command for a description. If the Print key does not map correctly, use ^P (Ctrl–P) instead.

F3 Return to the previous screen.

F10 Quit the monitor program.

7. To run a loop back diagnostic test from the **Status for Device** screen, press the F12 key. A loop back plug (IBM part number 43G0928) must be installed in the port to be tested. The loop back test consists of five phases:

PHASE 1 Verify that the port is not currently in use. All signals must be low. If a port is busy, a message similar to the following is displayed:

```
Port is Busy  :DTR:CD:DSR:RTS
```

PHASE 2 128 bytes of test data (the letter A is used) are put into the transmit buffer. If the write operation fails, a message similar to the following is displayed:

```
Loop Back Test Failure #1
```

After the write operation is complete, the FEP/OS is notified that data is available for transmission. At this time, the 128 bytes of data is transmitted. If the transmission fails, the following message is displayed:

```
Loop Back Test Failure #2
```

PHASE 3 At this point, the subsystem should have received 128 bytes of data. If not, the following message is displayed:

```
Loop Back Test Failure #3
Data Not Transferred to RX.
```

PHASE 4 128 bytes of data are read from the receive buffer. If the read operation fails, the following message is displayed:

```
Loop Back Test Failure #4
```

PHASE 5 The data that was transmitted is compared with the data that has been read from the receive buffer. If the comparison fails, the following message is displayed:

```
Loop Back Test Failure #5
```

If all test phases pass, the following message is displayed:

```
Loop Back Test Passed.
```

8. Press the F3 key to return to the **Status for Device** screen after the test is completed.

9. You can then press the F3 key to return to the **ACTIVE CONFIGURATION** screen and continue in the monitoring program or exit the SMIT interface.

Configuring Terminal–Attached Printers

Many of the ASCII terminals used today have an auxiliary serial or parallel port on which to connect a printer device. Connections of this type offer administrators a means of sharing valuable computer resources and increasing user productivity and efficiency by moving the printers as close as possible to the users. This section describes the steps necessary to configure this type of terminal–attached printer in an AIX Version 4 environment.

Hardware Requirements

- Bull 8– or 128–port adapter
- ASCII display (a Bull BQ306 display is used for this hardware)
- Serial printer
- EIA 232 serial cabling and gender changers

Prerequisites

1. The 128–port adapter is properly configured and operational.
2. The 3151 display is attached to the 128–port adapter and is operational.
3. The ASCII display is attached to the 128–port adapter and is operational.
4. Printer software for terminal–attached printing is installed.

Setting Up the Hardware

Before adding the printer to your AIX Version 4 system, perform the following steps:

1. Connect the serial printer to the auxiliary port of your terminal using an EIA 232 modem cable (Bull cable D). A null–modem cable is NOT needed on the AUX port of a Bull BQ306.
2. Make note of the following settings for your printer: line speed, word length (or bits–per–character), parity (odd, even, no, space, mark), and the number of stop bits.

Configuring the Auxiliary Port

On the Bull BQ306 terminal, perform the following steps:

1. Turn on your terminal simultaneously pressing the Ctrl and Setup keys. The SETUP menu is then displayed on the BQ306 screen.
2. Use the Send key to move between menu screens until the **KEYBOARD/PRINTER** option is displayed. Fill in the **PRINTER** options with the printer settings you noted previously.
3. Press the Send key until the FUNCTION screen is displayed. Select the **Save** option and press the Space Bar to save the configuration.

Adding the Print Queue

At this point, you have physically attached the printer to the terminal and configured its auxiliary port with the correct printer settings. The following procedures describe the creation of a local print queue on the host that will access the terminal–attached printer device.

1. Log on as root or as a member of the printq admin group.
2. Use the **smit mkpq** fast path to access the **Add a Print Queue** menu.

Note: The screen content may vary depending on the printer software installed on the host.

3. Select the **ascii** option, which takes you to the **Printer Type** screen.

4. Select the appropriate printer manufacturer or Other. For this example, we selected `IBM`. An additional **Printer Type** screen is displayed.
5. Select the appropriate printer model. For this example, we selected `ibm2380-2`. The **TTY Name** screen is displayed.
6. Select the tty for the terminal that has your printer attached. For this example, we selected `tty0`. The **Add a Print Queue** screen is then displayed.
7. Enter a descriptive name for the terminal–attached print queue (for example, `tty0asc`) in the **Name of new PRINT QUEUE to add** field, and press the Return key. The print queue will NOT be created.

Testing the Printer

To verify that the printer is functioning properly, issue an **lpstat** command to list all available print queues on the system. The following output is displayed:

```
# lpstat
Queue    Dev    Status   Job Files   User   PP %   Blks   Cp  Rnk
-----    ---    ------   ---------   ----   ----   ----   --  ---
4019g1   lp0    READY
4019ps   lp1    READY
tty0asc  tty0   READY
```

Use the following **enq** command to send an ASCII file to the printer:

```
enq –P tty0asc /etc/qconfig
```

The file should be printed without altering the display or functionality of the terminal.

Chapter 7. 8–Port Asynchronous ISA/PCI Adapter

The 8–port asynchronous ISA and PCI adapters are multi–channel, intelligent, serial communications features available for PowerPC computers.

The ISA adapters contain 128K of dual–ported high–speed Random Access Memory (RAM) used for program code and data buffering. The asynchronous ports are run by a 32–bit 16 MHz IDT 3041 processor that supports throughput speeds of 115Kbps.

The 3041 processor and dual–ported RAM help to offload much of the character processing from the system. Large blocks of data are transferred directly to the adapter, and then sent out on serial ports one character at a time.

The dual–ported RAM is accessible for read and write operations by both the adapter and the computer. The computer sees the dual ported RAM as its own memory and accesses it using the same high–speed memory referencing commands it uses for internal memory.

The 8–port EIA 232 ISA adapter supports EIA 232 devices only. This adapter requires the device package `devices.isa.cxia` to be installed on the system. This adapter is supported on AIX Version 4.1 and later.

The 8–port EIA 232/422 ISA adapter supports EIA 232 and EIA 422 devices. Both device types may be configured in any combination on a per–port basis. This adapter requires the device package `devices.isa.pc8s` to be installed on the system. This adapter is supported on AIX Version 4.2 and later.

The 8–port EIA 232/422 PCI adapter supports EIA 232 and EIA 422 devices. Both device types may be configured in any combination on a per port basis. This adapter requires the device package `devices.pci.4f111100` to be installed on the system. This adapter is supported on AIX Version 4.2 and later.

All of the above packages require the `devices.common.IBM.cx` package.

Installing 8–Port Adapters

ISA adapters cannot be autodetected by AIX and must be manually installed. PCI adapters are autodetected and installed by AIX, but can also be added later.

The following sections contain detailed information about the 8–Port asynchronous ISA/PCI adapters:

- Configuring the Adapter, on page 7-2
- Performance Tuning, on page 7-3
- Setting Terminal Options with stty–cxma, on page 3–16

Configuring the Adapter

ISA

1. To configure the 8–Port asynchronous EIA 232/EIA 422 ISA adapters, use the **smit mkdev_isa** fast path to access the **Add an ISA Adapter** screen.
2. Select **pcxr** (for the 8–port EIA 232 adapter) or **pc8s** (for the 8–port EIA 232/EIA 422 adapter) and press Enter.
3. Select the appropriate bus and press Enter.
4. In the Bus I/O Address field, set the address to the address of the adapter (set by DIP switches on the adapter). For additional information on DIP switches. The remainder of the adapter configuration is done automatically when the system displays **saX Available**.
5. When you have finished, select **Do**.

PCI

Run **cfgmgr** to configure the PCI 8–port adapter. There are no configurable attributes for this adapter.

Performance Tuning

The device driver software is configured to give the best performance under the widest variety of conditions. Performance under certain conditions can be improved through the use of tunable parameters. As with most tunable parameters, increasing performance in one area decreases performance in other areas. The software supports a number of tunable parameters that may be useful under special conditions. These parameters are tunable on a per–port basis and can be set with **stty–cxma**, **chdev**, or SMIT.

EDELAY is a tunable parameter used to determine the number of milliseconds of delay between the time the first character arrives after a period of no characters and notification of its arrival to the host. This is also referred to as the wakeup rate between the host software Front End Process Operating System (FEPOS) and the host device driver. This has the advantage of reducing host overhead by allowing the host to process larger blocks of incoming data. Larger EDELAY values result in more characters being sent in a given time period. This will reduce host processor utilization and character response time and increase overall system throughput. Smaller EDELAY values result in fewer characters being sent in a given time period. This increases character response time and increases host processor utilization. The default value for EDELAY is 100.

This is a good value for normal tty activity including typing and UUCP. For applications receiving continuous input at high speeds, increasing EDELAY results in lower host overhead and an increase in overall system throughput. A value of 250 is reasonable.

Note: For more terminal options, see ”Setting Terminal Options with stty–cxma”, on page 3–16.

Chapter 8. Serial Communications Network Server Overview

This chapter contains overview information for the 7318 Serial Communications Network Server. This chapter also includes specific information to set up and attach terminals, printers, and modems to a 7318 Model P10, guidelines for cabling your network, and specific information on installing and configuring the 7318 Model P10.

Refer to the *7318 Network Terminal Accelerator Guide and Reference* for information specific to the 7318 Model S20.

The IBM 7318 Serial Communications Network Server is a multiprotocol communications server that connects serial and parallel ports to an Ethernet. There are two models of the 7318, the Model P10 and S20. Both models have the same physical connectivity and appearance; however, the model S20 supports standard TCP/IP protocols in addition to the complete functionality of the Model P10.

The 7318 Model P10 and S20 provide for attachment of 16 serial asynchronous devices and two parallel printers to one or more systems utilizing standard Ethernet wiring and an Ethernet interface. The 7318 Model P10 is unique when compared to other Ethernet ”terminal servers” in that the attached devices appear to the systems as native devices attached through a multiport adapter.

This capability gives the user all the flexibility of Ethernet terminal servers with all the advantages of a multiport adapter. For instance, terminal users can connect directly to one or more systems on the network without being dependent on one system. Systems can also share devices attached to the 7318 Model P10. These connectivity advantages are provided while at the same time providing the same configurability and control as a native serial and parallel port.

The 7318 Models P10 and S20 connect to the system over Ethernet wiring. For systems that have integrated Ethernet, the 7318 has an advantage over multiport boards in that it consumes no adapter card slots. Model P10 ports are attached to systems using network cabling. The network is made transparent because the devices appear to be locally attached tty or LP devices.

The 7318 Model S20 ports can be configured to operate the same as a Model P10 port or can be used to take full advantage of a network environment. The Model S20 supports standard Transmission Control Protocol/Internet Protocol (TCP/IP) networking protocols, telnet and rlogin, which can be used to communicate to any similarly capable systems on the network. The Model S20 also supports 3270 terminal emulation for communication with applications like CICS/6000 or for communication with large systems. The 7318 Model S20 can also provide remote dial–in users with network access, using Serial Line Interface Protocol (SLIP), compressed SLIP (CSLIP), or Point–to–Point Protocol (PPP) serial line networking protocols. Dial–in users are able to run TCP/IP applications similar to other users locally attached to the network.

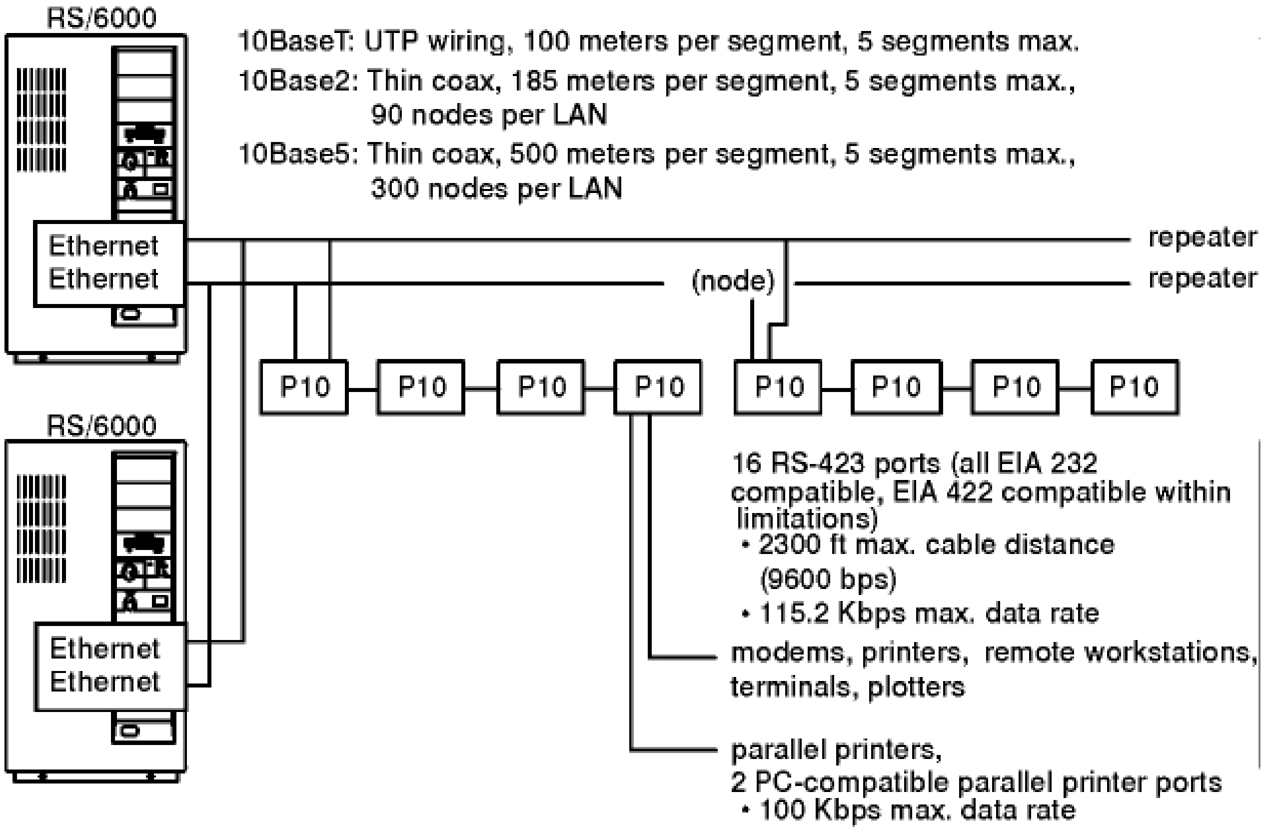
Both the Model P10 and the Model S20 have network security support. The Model P10 port or the Model S20 port configured as a P10 port can only communicate to systems configured for that 7318 port. Users have the same network access as a user attached to the native serial port on a system. The Model S20 can be configured for local or remote password security or Kerberos V.5 password authentication. Kerberos V.5 uses the Data Encryption Standard (DES) encryption algorithm to prevent passwords from being sent across the network. The Model S20 also has the capability to encrypt all the data associated with an rlogin application.

The Model S20 has the capability to take full advantage of the Network Terminal Accelerator features. The combination of the 7318 Model S20 and Network Terminal Accelerator eliminates the networking overhead of TCP/IP, the **telnetd** daemon, and the **rlogind** daemon. This combination enables the system to support more users or more applications than before.

Refer to the *7318 Network Terminal Accelerator Guide and Reference* for more information on the 7318 Model S20.

7318 Model P10

The 7318 Model P10 Ethernet Attached Async figure provides a more complete overview of the 7318 Model P10.



7318 Model P10 Ethernet Attached Asynchronous

Ethernet Wiring

Ethernet cabling provides flexible wiring options including unshielded twisted pair and two types of coax. Distances can be extended using up to four repeaters for a maximum of five segments.

Attachment to the system requires an Ethernet interface. For many systems, this will not require a system adapter card slot.

Up to four 7318s may be daisy–chained using only one Ethernet connection (node).

Ethernet wiring can be used for purposes other than attaching 7318s. The 7318 Model P10 utilizes Novell's SPX/IPX protocols, which are interoperable with other Ethernet protocols.

The 7318 supports 10 Mbps Ethernet, 10BaseT, 10Base2, and 10Base5.

7318 Serial and Parallel Port Capabilities

Up to 16 EIA 422 devices are supported with limitations. The 16 serial ports support data rates up to 115.2 Kbps with certain limitations. All 16 ports are EIA 232D and RS–423 compatible. The maximum cable distance supported for an EIA 232D device is 200 ft, while the maximum cable distance supported for an RS–423 or EIA 422 device is 2300 ft at 9600bps. Two PC–compatible parallel ports are also provided for faster printing with burst data rates up to 100 Kbps.

7318 Multi–System Capability

Systems can be configured to share 7318s, 7318 ports and devices, or terminal users attached to the 7318 can login to multiple systems simultaneously using multiple sessions.

Standby Network Capability

The 7318 can be attached to a standby network for availability purposes. When properly configured, the 7318 will detect a network failure and switch to its standby Ethernet interface.

7318 Hardware and Software

The following sections list the features of the 7318 hardware and software, illustrate some possible network configurations, and outline the functions of the front and rear panels.

- Hardware Overview
- Sample 7318 Configuration
- Model P10 Software Overview
- Contents of the Package
- Attachment Requirements
- Software Prerequisites
- Front Panel
- Back Panel
- Base Panel

Hardware Overview

The 7318 is a multiprotocol communications server providing serial and parallel connectivity to Ethernet networks. Highlights of the P10 and S20 models include:

- Sixteen asynchronous serial ports to connect terminals, printers, modems, and other EIA 232, EIA 422, or RS–423 devices.
- Serial data rates up to 115.2 Kbps on all 16 ports.
- Two parallel printer ports with data rates up to 100 Kbps.
- 10BaseT and Adapter Unit Interface (AUI) Ethernet interfaces.
- 10BaseT hub port for direct host connection.
- Daisy–chain ports for attaching up to four 7318s to one Ethernet connection.
- Structured wiring support that minimizes cabling costs.
- Desktop, rack, or wall–mounting.
- Built–in surge protection.
- Common wiring for serial and 10BaseT Ethernet ports.
- Buddy mode, providing full modem control by configuring an odd and even port together.
- Connectivity accessories.

Ethernet

The 7318 attaches to a network or host system, using a 10BaseT port or an AUI Ethernet port. The 10BaseT connections include both a hub port and a host system port. The hub port permits you to use twisted–pair cable to directly attach the 7318 to a host, bypassing the need for a separate hub.

Asynchronous Ports

The front panel of the 7318 provides 16 easily accessible serial ports. All 16 ports can operate simultaneously at data rates up to 115.2 Kbps. The serial ports support EIA 232, RS–423, and EIA 422 devices. RS–423 and EIA 422 devices allow greater cabling distances (up to 2300 ft at 9600 baud). The 7318 is also EIA 422 interoperable with restrictions.

Limitations

1. The 7318 does not support multidrop RS–423 or EIA 422.

2. Up to 16 EIA 422 devices are supported if a 100–ohm current–limiting resistor is placed in series on the TD signal at the device end of the cable. This resistor is required to prevent the 7318 drivers from premature failure. This current–limiting resistor is not required if only the odd–numbered ports are used for EIA 422 devices.
3. Serial input data with a character size of 5 bits is reliable at a maximum data rate of 38.4 Kbps. Serial input data with a character size of 6 or 7 bits is reliable at a maximum data rate of 57.6 Kbps. Serial input data with a character size of 8 bits is reliable to a maximum data rate of 76.8 Kbps. Reliable operation with a serial input data rate of 115.2 Kbps requires SLIP/TCP for the Model P10 and TCP/IP protocols using SLIP, CSLIP, or PPP for the Model S20.
4. Other limitations on EIA 422 interoperability include distance and data rate limited to Electronics Industries Association (EIA) RS–423 specifications.

Parallel Ports

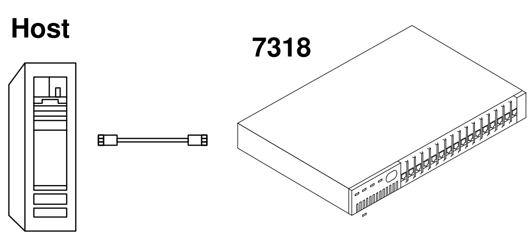
The 7318 provides two parallel ports for network printing. These ports use standard DB–25 connectors, making it possible to attach most industry–standard printers to your network.

Sample 7318 Configuration

The following figures depict five different configurations, each illustrating some aspect of 7318 connectivity. The 7318s may be either Model P10s or Model S20s.

Configuration One

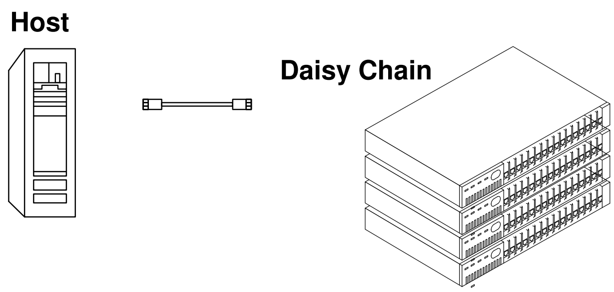
The Single 7318 figure depicts a direct–connect 7318. In this case, the RJ–45 Ethernet cable would attach to the T–X port on the back of the 7318 and to the 10BaseT Ethernet port on the host. This is the simplest configuration and does not require an Ethernet transceiver or an Ethernet hub. However, no other devices can share the Ethernet in this configuration.



Single 7318

Configuration Two

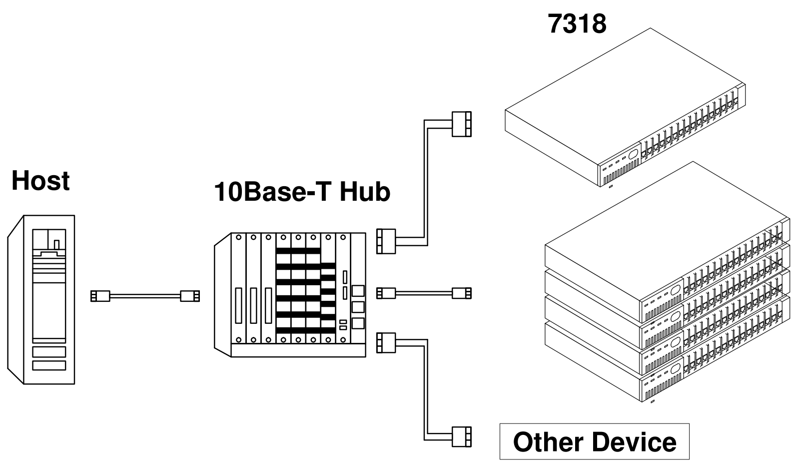
The Daisy–Chained 7318s figure depicts a direct–connect stack of 7318s daisy–chained to each other using the I and O ports. In this case, the RJ–45 Ethernet cable would attach to the T–X port on the back of the first (top) 7318 and to the 10BaseT Ethernet port on the host. This configuration is like the configuration shown in the Single 7318 figure, except that up to 64 asynchronous ports can be attached on a single Ethernet connection. Each 7318 still maintains its own Ethernet address and acts independently. No other device types can share the Ethernet in this configuration.



Daisy-Chained 7318s

Configuration Three

The 7318s with 10BaseT Hub figure depicts a single 7318 and four daisy–chained 7318s connected to a 10BaseT hub. In this case, the RJ–45 Ethernet cable attaches to the T port on the back of the single 7318 and on the first (top) daisy–chained 7318. Using a 10BaseT hub allows the full flexibility of Ethernet 10BaseT wiring.

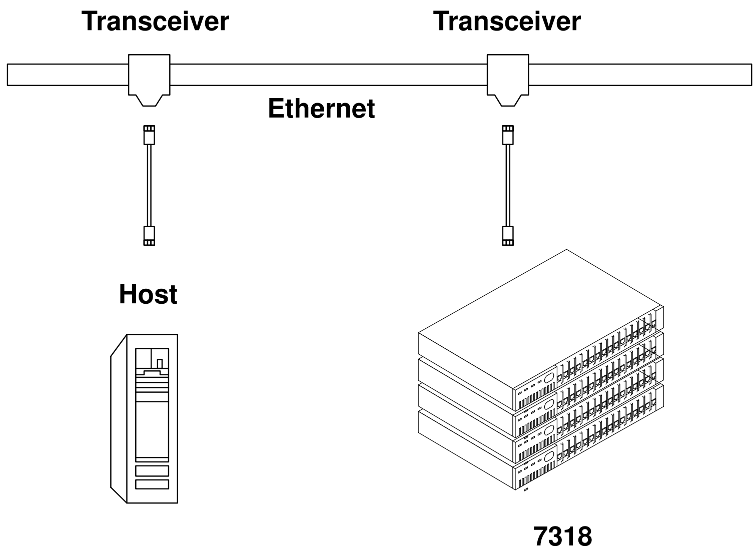


7318s with 10Base-T Hub

Note: The four daisy–chained units are addressed and act as if they are four separately connected units.

Configuration Four

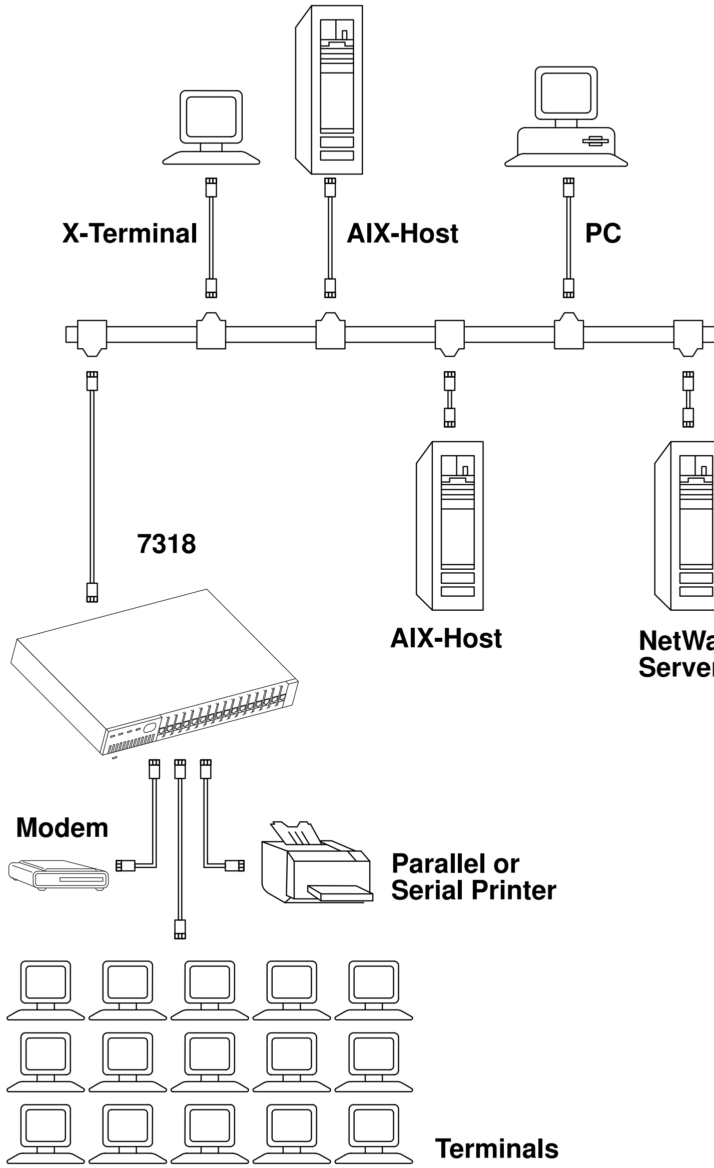
The Daisy–Chained 7318s Connected to Ethernet figure depicts four daisy–chained 7318s connected to an Ethernet network. In this case, the cable connects to the AUI port or the T port on the back of the 7318. Only one transceiver is necessary when 7318s are daisy–chained. With 10Base2 or 10Base5 Ethernet, longer distances can be used. Again, this arrangement allows for full flexibility in network layout and implementation.



Daisy-Chained 7318s Connected to Ethernet

Configuration Five

The Network Resource for Sharing Parallel and Serial Devices figure depicts a 10Base2 or 10Base5 network with several systems and devices attached to the Ethernet. The devices attached to the 7318 are resources available to the network and are configured and used by the various network hosts.



Network Resource for Sharing Parallel and Serial Devices

Model P10 Software Overview

The protocol software package running on the 7318 enables users to connect terminals, printers, and modems to an AIX host. The software features are as follows:

- The 7318 can emulate /**dev**/**tty***X* and /**dev**/**lp***X* serial ports and /**dev**/**lp***X* parallel ports and the associated AIX driver software.
- Using multiple 7318s, users can provide an unlimited number of ports to one or more AIX hosts.
- The software supports most character processing on the 7318, freeing host resources for other processes.

- The software supports multiple sessions on a terminal, enabling a user to rapidly switch between applications.
- The software supports terminal access to any network host running the protocol software.
- The software supports transparent printing using a terminal's auxiliary port, eliminating the need for a separate printer connection.
- The 7318 can switch to a backup Ethernet following a network failure.

With the P10 package, all 16 terminals and the parallel printers appear to be attached directly to an AIX host (or hosts). The P10 package requires minimal setup and configuration and requires fewer CPU resources than the standard TCP/IP protocols running on the host.

Contents of the Package

The 7318 package includes:

- An AC power cord
- Two RJ–45 serial wrap plugs
- One AUI wrap plug
- One 10BaseT wrap plug
- One Parallel wrap cable.

Attachment Requirements

The 7318 must be attached to your network in one of three ways:

- Directly to a 10BaseT Ethernet host adapter
- To an Ethernet transceiver, or medium attachment unit (MAU)
- To a 10BaseT Ethernet hub.

Software Prerequisites

For AIX Version 3.2 systems, you must install Version 3.2.5. The optionally installable component of AIX, **bosnet**, must be installed, in addition to the licensed program.

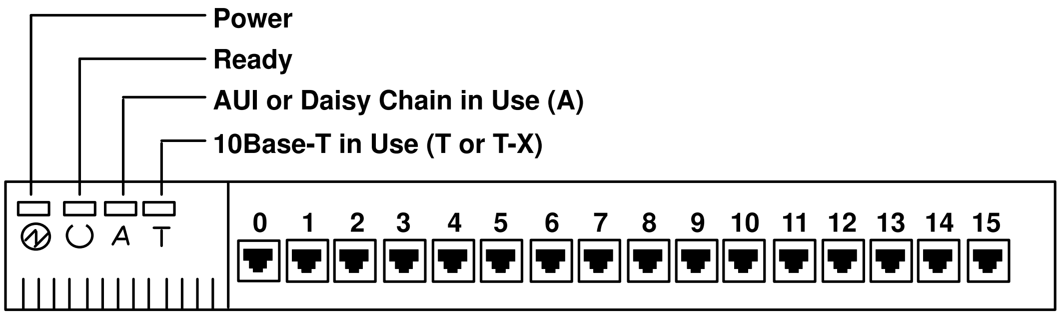
Note: If Netware V3.11x will be concurrently operating on a system with the Async Terminal Server–Accelerator/6000 licensed programming product, it is strongly recommended that the most current PTF for Netware V3.11 be installed. If this is not done, client users may likely experience a degradation in network performance.

For AIX Version 4.1 systems, you must install AIX Version 4.1.4 and the **devices.7318.p10** package. (No separate licensed program is required.)

For AIX Version 4.2 systems, you must install AIX Version 4.2 (or later) and the **ipx.rte 4.2.0.0** package.

Front Panel

The front panel of the 7318 contains four indicator lights to show the status of the 7318 and 16 serial ports for interfacing with serial devices. The Front Panel figure illustrates the layout of the front panel.



Front Panel

The front panel components and their functions are:

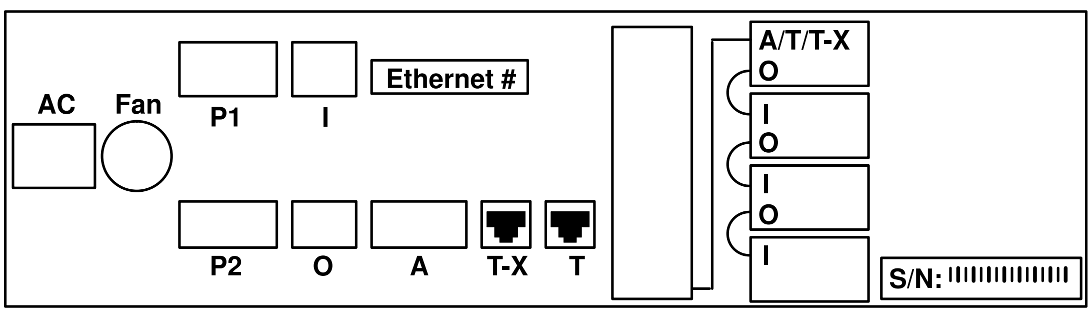
Serial ports Connect EIA 232, EIA 422, or RS–423 serial devices to the 7318. These 16 RS–423 serial ports are numbered from left to right, 0 to 15, and are described more fully in ”Connector Pinouts”, on page B-1.

Status lights Show the status of the 7318. From left to right, the lights correspond to Power, Ready, AUI or Daisy Chain in Use (A), and 10BaseT in Use (T).

Refer to ”Powering Up the 7318”, on page 8-17 for additional information about the four status lights.

Back Panel

The back panel contains the AC socket, fan, parallel ports, Ethernet AUI port, daisy–chain port, and 10BaseT ports. The Back Panel figure illustrates the layout of the back panel.



Back Panel

The back panel components and their functions are:

Universal AC Socket (IEC–320) Provides the connection point for the power cord supplied with the 7318.

Fan Provides ventilation to the internal components of the 7318. When installing your 7318, do not block the fan or the front vents.

Parallel ports Connect parallel printers. These two industry–standard parallel ports, P1 and P2, are compatible with those on a PC. For pinout information, refer to ”Connector Pinouts”, on page B-1.

Daisy–chain ports Connect one 7318 to another. These two ports, I and O, allow connection of up to four 7318s, using a single Ethernet transceiver or 10BaseT connection.

Ethernet number Specifies the 7318's Ethernet hardware address. This address applies to all of the Ethernet ports. Administrators should make a note of this address because they will need to refer to it during the installation process.

Ethernet AUI port Connects to an Ethernet transceiver. Use this port and/or one of the 10BaseT ports to connect the 7318 to a network. For pinout information, refer to ”Connector Pinouts”, on page B-1.

10BaseT ports Specify two 10BaseT RJ–45 ports. Both ports (labeled T–X and T) connect to the same interface, but the wiring for each port is different. The T–X port is wired as a hub connection, allowing for direct connection to a host. The T port is wired as a client connection. Use either of these ports or the AUI port to connect the 7318 to a network. For pinout information, refer to ”Connector Pinouts”, on page B-1.

Daisy–chain diagram Illustrates the correct configuration for daisy–chaining four 7318s to a host system.

Serial number Specifies the 7318's serial number.

Base Panel

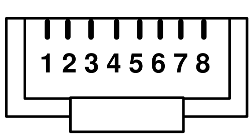
There is a product summary label affixed to the base panel on the bottom of the 7318. The Product Summary Label figure illustrates the information contained in the product summary.

Rear-panel connectors.

Input Power	**100-240VAC 50-660 Hz 0.75 Max**
P1	**Parallel Interface 1**
P2	**Parallel Interface 2**
T	**10Base-T**
T-X	**10Base-T Hub**
A	**Ethernet AUI Interface**
I	**Daisy Chain Input Port**
O	**Daisy Chain Output Port**

Front-panel connectors 0-15:

Pin	CCITT V.24	Designator	Circuit Name (*denotes output)
1	109	DCD	Data Carrier Detect
2	108/2	DTR	* Data Terminal Ready
3	105	RTS	* Request to Send
4	104	RD	Received Data
5	102 (See Note)	RD Reference	Signal Ground
6	106	CTS	Clear to Send
7	102 (See Note)	TD Reference	Signal Ground
8	103	TD	* Transmitted Data



RJ-45 Jack Pinout

Note:
For EIA 232D or CCITT V.28, connect pin 5 to 7.

Product Summary Label

Additional details on the connector and cable pinouts can be found in ”Connector Pinouts”, on page B-1.

Serial Communications Network Server (7318) Installation

This section contains installation information for hardware and software. When you have completed the installation procedures, refer to ”Configuring a 7318 Model P10”, on page 8-20.

Refer to Chapter 2, ”Serial Communications Network Server (7318) Installation,” in the *7318 Network Terminal Accelerator Guide and Reference* for information specific to the 7318 Model S20.

Electrical Safety

Observe the following safety instructions any time you are connecting or disconnecting devices attached to the workstation.

DANGER: An electrical outlet that is not correctly wired could place hazardous voltage on metal parts of the system or the devices that attach to the system. It is the responsibility of the customer to ensure that the outlet is correctly wired and grounded to prevent an electrical shock.

Before installing or removing signal cables, ensure that the power cables for the system unit and all attached devices are unplugged.

When adding or removing any additional devices to or from the system, ensure that the power cables for those devices are unplugged before the signal cables are connected. If possible, disconnect all power cables from the existing system before you add a device.

Use one hand, when possible, to connect or disconnect signal cables to prevent a possible shock from touching two surfaces with different electrical potentials.

During an electrical storm, do not connect cables for display stations, printers, telephones, or station protectors for communications lines.

CAUTION:This product is equipped with a three–wire power cable and plug for the user’s safety. Use this power cable in conjunction with a properly grounded electrical outlet to avoid electrical shock.

DANGER: To prevent electrical shock hazard, disconnect the power cable from the electrical outlet before relocating the system.

7318 Hardware Installation

To install the 7318 hardware, follow these four basic steps:

1. Mount the 7318 on a desktop, wall, or standard 19–inch Electronics Industries Association (EIA) rack.
2. Attach network cables to the 7318.
3. Attach cables to the devices.
4. Start the 7318 by plugging the power cord into an AC outlet.

Desktop Mounting

The 7318 can be positioned on any surface, provided you do not block the air flow from the front and rear of the unit.

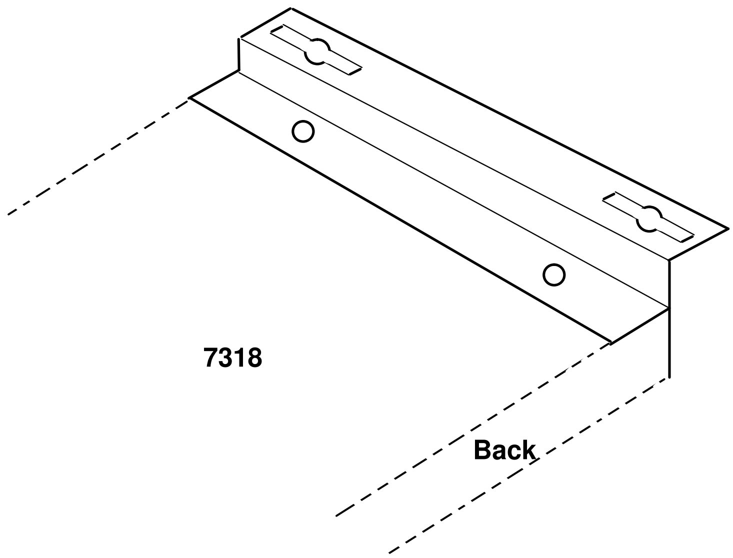
Wall Mounting

Wall mounting works well for a single 7318. When mounting multiple 7318s, install a shelf on the wall and place the 7318s on the shelf or use an EIA rack.

To mount the 7318 on a wall:

1. Have ready the optional wall–mounting brackets (Feature Code 7900).

2. Turn the unit upside down.
3. Remove the four screws on the bottom of the 7318 that attach the cover. Do not remove the cover.
4. Place the wall–mount brackets so the holes line up with the holes on the 7318 as illustrated in the Wall–Mount Bracket figure.



Wall-Mount Bracket

5. Reinstall the screws through the bracket into the 7318.
6. Hold the 7318 up to the wall and mark the location of the mounting holes on the wall. Cabling will be easier if the front and back of the 7318 are vertical rather than horizontal, but this is not required.
7. Drill holes in the wall and insert the screws provided with the wall–mount brackets, leaving about 10 mm (1/4–inch) between the screwhead and the wall.
8. Place the 7318 keyholes over the screws and let the 7318 slide into the keyhole slots. Tighten the screws.

Rack Mounting

Use rack mounting to install one or more 7318s in a single location. Mount the 7318 in the rack, using the optional rack–mounting brackets, or place the 7318s on a standard rack shelf.

To mount the 7318 in an EIA rack:

1. Have ready the optional rack–mounting brackets (Feature Code 7908).
2. Turn the unit upside down and remove the four screws that hold the cover of the 7318, but do not remove the cover.
3. Place the rack–mount bracket so that the holes line up with the holes on the 7318.
4. Reinstall the four screws.
5. Slide the 7318 into the rack.
6. Attach the 7318 to the rack with the screws provided with the rack–mounting brackets.

Attaching Network Cables

The following information briefly describes the choices for attaching to a network. You can choose between attaching the 7318 to a host adapter, a hub, or an external Ethernet transceiver.The following cables are available for network attachment:

- RJ–45 to RJ–45 cable (Feature Code 7901)
- Daisy–chain cable (Feature Code 7909)

AUI Port

The AUI port connects the 7318 to an external transceiver (Part Number 02G7437 for Thin or Part Number 02G7431 for 10BaseT) for the Ethernet network. You can use AUI or 10BaseT but only one can be active at a time (unless in a high–availability [HA] environment). The AUI port supports a directly attached transceiver or a cable to a remote transceiver. AUI transceiver cables are specially designed for this application and normally come with the transceiver.

If you are in an HA environment, attach cables to both the AUI and T ports, one to each Ethernet network.

Note: Only one Ethernet network is used at a time.

See ”7318 High–Availability Environment for the Model P10”, on page 8-47 for more information.

T Port

You can use standard 10BaseT unshielded twisted–pair (UTP) cable to connect to a 10BaseT hub. There are no performance advantages to using a particular Ethernet port on the 7318.

When attaching to a 10BaseT hub, connect the cable from the hub to the T port on the 7318. Use the T port as the second Ethernet connection in an HA environment.

T–X Port

As with the T port, you can use standard 10BaseT unshielded twisted–pair (UTP) cable to connect to a 10BaseT host adapter. There are no performance advantages to using a particular Ethernet port on the 7318.

Unlike the T port, when attaching directly to a 10BaseT host adapter, connect the cable from the host adapter to the T–X port on the 7318. The following procedure explains this simple installation process. In this instance, other network hosts share the 7318 only through the attached host.

To install a 7318 directly to a host:

1. Install an Ethernet host adapter in the host computer. For this simplified installation, you must have a 10BaseT connector on the host adapter, or a transceiver (MAU) with a 10BaseT connector.
2. Use an RJ–45 cable to connect the RJ–45 port on the host adapter to the T–X port on the back panel of the 7318.

Note: If you need a longer cable than the 10 foot RJ–45 to RJ–45 cable (Feature Code 7901), be sure to use a cable designed for 10BaseT.

Attaching Multiple 7318s

If you are planning to connect more than one 7318 to the network, you can either connect the 7318s individually or use daisy–chain connectors. Using the daisy–chaining feature reduces the number of hub or host adapter connectors required for the additional 7318s.

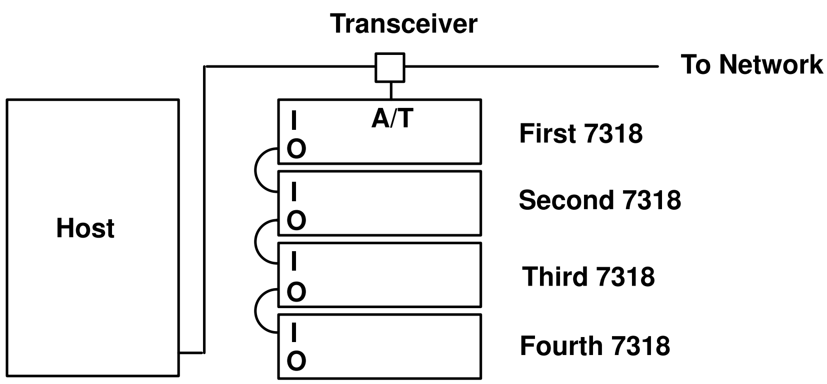
Using the 7318 I (In) and O (Out) ports and daisy–chain cable, you can attach up to four 7318s to the same Ethernet connection. This reduces the need for separate transceivers and additional 10BaseT hub ports. Daisy–chained 7318s each maintain their own Ethernet and IP addresses, load, and operate independently. The P10s and S20s may be mixed in a single daisy chain.

Use the daisy–chain cable (Feature Code 7909) to connect one 7318 to another. You must remove power to all 7318s in the daisy–chain group before installing a daisy–chain cable. If you plug in a daisy–chain cable to an operating unit, the Ethernet interface will not operate properly. You must power cycle the unit, because the 7318s recognize their daisy–chain configuration only when they power up.

The 7318 that is attached to the Ethernet is called the ”first” in the chain. Only the first 7318 is connected to the Ethernet; the rest are connected through the daisy–chain connectors.

The additional 7318s are attached in order; that is, the second is attached to the first, the third to the second, and the fourth to the third. The last 7318 in the line is connected to only one other. Use the O (Out) connector of the higher 7318 to connect to the I (In) connector of the lower 7318.

The Daisy–Chained 7318s figure illustrates the cabling relationship among four daisy–chained 7318s as well as their relationship to the host.



Daisy-Chained 7318s

While the daisy–chain feature is easy to use, you must observe a few rules when chaining 7318s:

- Ensure that only the first 7318 of the group is attached to the Ethernet. The daisy–chained group will not operate reliably if there is more than one Ethernet connection.
- Do not use video graphics adapter (VGA) extension cables even though the connectors are the same. VGA extension cables are unreliable for connecting one 7318 to another.
- Daisy–chained units can only be placed as far apart as the supplied cable allows.
- Do not remove 7318s from the middle or end of a chain while the chain is connected to the Ethernet. To remove a daisy–chained 7318 (for example, for repair), perform the following steps:
  a. Power off the first 7318 by unplugging it.
  b. Power off the others in the chain by unplugging them.
  c. Remove the daisy–chain cables from the chosen 7318.
  d. Physically remove the 7318 from the daisy–chain stack.
  e. Reconnect the daisy–chain cables on the remaining machines.
  f. Power on the 7318s by plugging them in, starting with the first.
- Each 7318 in a chain has a different Ethernet address. This can affect hub management and security software for the network. For example, some hubs only allow packets on a single–hub connection to come from a single Ethernet address. Daisy–chained 7318s would clearly not work with such a hub. Refer to hub management and security software administration manuals if problems relating to multiple Ethernet addresses arise after adding daisy–chained 7318s.
- All 7318s in a daisy–chain must have power for any of the daisy–chained units to communicate.

Powering Up the 7318

The 7318 does not provide a power on/off switch. You must use the AC plug to power on and power off the 7318.

Plug in the 7318 to ensure that the server is functional. The lights on the front will turn on or stay off in different configurations depending on the state of the 7318, the download, and the network.

The operation of these lights is as follows:

1. All four lights cycle on, then off in the light test.
2. The power light stays on after the light test, indicating that the 7318 has AC power.
3. The ready light stays off until after the power–on self–test runs, then blinks until the configuration software downloads to the 7318.
4. If the AUI interface or daisy chain is selected and operational, and the transceiver supports the SQE heartbeat, the (A) light turns on.
5. If the 10BaseT interface is selected and has link integrity, the (T) light turns on.

   Note: Ethernet transceivers must support SQE heartbeat for the AUI light to function properly. If your transceiver does not support SQE heartbeat, the AUI light will not light.
6. The ready light stays on solid once the software download is complete. If the light stays off, there is a problem with the 7318 hardware.

   Note: Daisy–chained 7318s only display the AUI port light, even if the first 7318 is connected using the 10BaseT (T or T–X) port.

Attaching Additional Cables

”Connector Pinouts”, on page B-1 describes the pinouts for all cables. The following cables are available for attaching devices:

- Extended EIA 232D modem control cable (Feature Code 7902)
- Macintosh DIN–8 cable (Feature Code 7906)
- RJ–45 to RJ–45 cable (Feature Code 7901)
- RJ–45 to MMJ cable (Feature Code 7907)

Each cable measures 10 ft in length.

Use the 7318 planning chart, on page 8-25 to assist you in planning your cable configuration. There is a blank chart for your use following the example.

Installing 7318 Software for Version 3.2

To install the protocol software and configure the system and devices, use the System Management Interface Tool (SMIT). The following information describes the installation and system–configuration process for Version 3.2.

Prerequisites

1. You must have root user authority.
2. If you have an Version 3.2 system, you must have Version 3.2.5 installed.
3. The optionally installable component of AIX, **bosnet**, must be installed.

   Notes:

   a. If Netware V3.11x will be concurrently operating with the Async Terminal Server–Accelerator/6000 licensed programming product, it is strongly recommended that the most current program temporary fix (PTF) for NetWare V3.11 be installed. If this is not done, client users may likely experience degradation in network performance.

b. Software prerequisites might be installed automatically through the following procedure.

Procedure

The following steps will help you install the software:

1. Use the **smit install_latest** fast path to access the **Install Software Products at Latest Available Level** menu.
2. At the INPUT device/directory for software field, select the device that has the software tape drive device.
3. Select the software to install. Change the COMMIT software? field to **no**. Change the SAVE replaced files? field to **yes**. The Automatically install PREREQUISITE software? field defaults to **yes**.

   The following describes the contents of the package to install.

p10	SMIT menus, **cnsview**, p10 bootfile, p10 default configuration file.

4. Select **Do** to begin loading the software.
5. When the software is finished loading, exit the SMIT interface.

Version 3.2 Packages

`cns.p10.obj`
Install for Model P10 support

`cns.s20.obj`
Install for Model S20 support

`cns.s20export.obj`

Installing 7318 Software for Version 4.1

The following information describes the installation and system–configuration process for AIX Version 4.1.

Prerequisites

1. You must have root user authority.
2. If you have an AIX Version 4.1 system, you must have AIX Version 4.1.4 installed.

Procedure

1. Use the **smit install_latest** fast path to access the **Install New Software Products at Latest Level** menu.
2. At the INPUT device/directory for software field, select the device name or directory that has the installation images.
3. At the SOFTWARE to install field, select **devices.7318.p10**. Change the COMMIT software updates? field to **no**. Change the SAVE replaced files? field to **yes**.
4. Select **Do** to start the installation.
5. When the software is finished loading, exit the SMIT interface.

Version 4.1 Packages

`devices.7318.com`
Common support, not explicitly installed

`devices.7318.p10`
Install for Model P10 support

Installing 7318 Software for AIX Version 4.2 (or later)

The following information describes the installation and system–configuration process for AIX Version 4.2 (and later).

Prerequisites

1. You must have root user authority.
2. If you have an AIX Version 4.2 system, you must have AIX Version 4.2 (or later) installed.
3. The **ipx.rte 4.2.0.0** package must be installed.

Procedure

1. Use the **smit install_latest** fast path to access the **Install and Update from LATEST Available Software** menu.
2. At the INPUT device/directory for software field, select the device name or directory that has the installation images (for example, **/dev/cd0**).
3. At the SOFTWARE to install field, select **bos.cns**. Change the COMMIT software updates? field to **no**. Change the SAVE replaced files? field to **yes**. Change the AUTOMATICALLY install requisite software? field to **yes**. Change the EXTEND file systems if space needed? field to **yes**.
4. Select **Do** to start the installation.
5. When the software is finished loading, exit the SMIT interface.
6. Reboot the system.

AIX Version 4.2 (and later) Packages

`bos.cns`
Install for 7318 Model P10 and S20.

`bos.cns.s20des`
Install for 7318 Model S20 with DES data encryption.

7318 Model P10 SPX/IPX Addressing

The 7318 Model P10 uses the SPX/IPX protocols for communication with the system. SPX/IPX use a different addressing method than IP addressing. Similar to IP addressing, each node on a network has a unique address. However, with IPX, each network segment has its unique identifier also.The system is given a node address in the **NPSConfig** file located in the **/etc/netware** directory. The `internal_network` field in this file specifies the node address for the system. The complete address of the system will be the `internal_network` number followed by a colon and the number 000000000001. For example:

```
00000005:000000000001
```

The **NPSConfig** file is also used to define the network segment identifier to the system. If there are multiple network adapters in the system, there will be a network segment identifier for each adapter card. This network segment identifier is defined by the lan_x_network field in the **NPSConfig** file, where `x` is the number associated with the adapter card stanza in the file.

Typically, the default network segment identifier is 00000002, and the **NPSConfig** file entry for lan_x_network is:

```
lan_1_network=00000002
```

The 7318 automatically defines its SPX/IPX address to be the network segment identifier plus its Ethernet MAC address. The MAC address is found on the back panel of the 7318. For example:

`00000002:00406e000321`

Medium access control (MAC) addresses are unique to each 7318, so each 7318 on a network has a unique address. The 7318 detects the network segment identifier during the boot process using Service Advertising Protocol (SAP) broadcast packets that are sent by SPX/IPX nodes on a network.

During the SMIT configuration process for adding a 7318, you must complete two fields that define to the software the 7318's SPX/IPX address. Therefore, you need to know the network segment identifier for the segment the 7318 is attached to and its MAC address.

For a 7318 attached to a network segment shared with the system, the network segment identifier can be found in the **NPSConfig** file (lan_x_network). For a 7318 attached to a remote Ethernet, the information must be obtained from the router or another SPX/IPX node on that network.

In the remote Ethernet configuration, the ComNetServer Network Address field in SMIT must be set to the network segment identifier and not the lan_x_network number in the system's **NPSConfig** file.

Configuring a 7318 Model P10

After installing the software, add the 7318 devices to your AIX system. Repeat these steps for each 7318 you add to the network. For example, if you are adding two 7318s, you must run these steps twice, indicating a different 7318 (by Ethernet address) each time.

Prerequisites

1. Fill out the 7318 P10 Installation Worksheet, on page 8-24 to record relevant data before using SMIT to add and configure the 7318.
2. The Ethernet interface must be configured and running. Run the **smit chinet** command, select the interface in question, and when the dialog displays, note the Current STATE field. If it is `up`, the interface is running. If it is `down` or `detach`, change the network state to `up`. Refer to "TCP/IP Network Interfaces" in *AIX 4.3 System Management Guide: Communications and Networks* for information on automatic configuration of network interfaces, how to list network interfaces, and how to configure network interfaces.
3. You must have root user authority.

Procedure

The following steps define an SPX/IPX network address, an Ethernet hardware address, an interface type, and a boot file name. Depending on your choice, network ttys and virtual ttys can be generated as well.

1. Use the **smit ts7318_cs_add_hdr** fast path to access the **Add a ComNetServer** menu.
2. In the New ComNetServer Number field, enter the number for this server. For example, if this is the first server you are adding, the number is **01**. Select the **List** option to see the next available number (maximum of 99).
3. In the Autogenerated TTY Devices field, select the **List** option and select one of the following:
   - **ttys only** – generates all 16 tty devices for this 7318, one for each 7318 serial port.
   - **ttys and virtual ttys** – generates all 16 tty devices, as well as all 6 virtual ttys for each 7318 serial port. Refer to "Enabling Multiple Sessions", on page 8-33 for more information on virtual ttys.
   - **none** – does not generate any devices.

   Notes:

   a. Generating ttys and virtual ttys from this dialog is much faster than generating them one at a time later.

b. You must select **ttys and virtual ttys** for a port to use transparent print. Refer to ”Configuring a Transparent Printer for a Model P10”, on page 8-36.

4. In the Interface Type field, select **EIA 232** or **EIA 422**.
5. From the 7318 P10 Installation Worksheet, enter the network address (found in the **/etc/netware/NPSConfig** file on the `lan_X_network` line) in the ComNetServer Network Address field.
6. From the 7318 P10 Installation Worksheet, enter the Ethernet hardware address in the ComNetServer Ethernet Address field.
7. If you want to ensure that the present host is *not* the load host, blank out the ComNetServer Bootfile field. The default load image file name should be **/usr/lib/cns/cns–p10**.
8. Select **Do** to add the 7318. You will see many messages as SMIT creates the devices for this 7318.
9. Exit (F10) the SMIT interface.
10. Use the **ts7318_cs_mnu** fast path to access the **ComNetServer Server Configuration** menu. Proceed to the procedure described in ”Enabling ttys”, on page 8-21 to enable login prompts for each tty that you generated.

Generating a Single 7318 tty Device

The following information details steps to create a tty device for a specific session and port on a 7318.

Prerequisites

1. You must have root user authority.
2. The 7318 software must be installed. Refer to ”Installing 7318 Software for AIX Version 3.2” , on page 8-17, ”Installing 7318 Software for AIX Version 4.1”, on page 8-18, or ”Installing 7318 Software for AIX Version 4.2 (or later)”, on page 8-19 for instructions on installing the 7318 software.
3. The 7318 Model P10 must have been added to the host’s configuration. Refer to ”Configuring a 7318 Model P10”, on page 8-20 for steps to configure the host when adding a 7318.

Procedure

1. To add 7318s to your network, use the **smit ts7318_tty_mnu** fast path to access the **ComNetServer TTY Configuration** menu.
2. Select **Add a TTY to a ComNetServer**.
3. Select the 7318 on which to add this tty.
4. Select the port on which to add this tty.
5. Select the type of tty to add to this port.
6. Fill in the dialog fields with the appropriate information. Do not change the Interface Type field unless this is the first and only tty defined on this port. This value (EIA 232 or EIA 422) should be the same for all hosts that configure sessions on this 7318 port.
7. When you have finished, select **Do**.

Enabling ttys

If you generated virtual devices on any host for your Model P10 ports and you plan to configure a single port for multiple hosts, refer to ”Configuring Ports for Dedicated or Multiple Hosts”, on page 8-43 for important configuration information. The following information explains how to enable a tty on the host. You must enable the tty to start a **getty** process, which is a prerequisite to obtaining a login screen.

Prerequisites

You must have root user authority.

Procedure

To enable a tty effective on the next system reboot, use the following steps:

1. Use the **smit chgtty** fast path to access the list of available ttys to enable. Select the desired tty, and press Enter.
2. Select **Change/Show TTY Program**. This option is valid for AIX Version 3.2.5 systems only. If you are on an AIX Version 4 system, go to step 3.
3. Change the Enable LOGIN field to **enable**.
4. Select **Do**.
5. Select **Exit**.

   Repeat the previous steps for each terminal you want to enable. By default, all ports are enabled at 9600 baud, 8–bit characters, 1 stop bit, and no parity. Using these settings, you can hook up most terminals to start using your 7318. To add terminals, printers, or modems that do not use these default settings, refer to ”Installing a Terminal with Defaults”, on page 8-27, ”Installing and Configuring Printers”, on page 8-34, and ”Modems”, on page 8-37.

6. To enable the tty or ttys for the present session (without a system reboot), use the **penable** command. To enable a single tty, type:

   ```
   penable ttyX
   ```

   where the *X* parameter specifies the tty number to enable. Use the **/usr/lpp/cns/samples/build_ttys** (on Version 3.2.5) or the **/usr/samples/7318/build_ttys** (on AIX Version 4) sample shell script to enable a large number of ttys.

   Note: The **penable** command does not change the ODM database. Perform steps 2 through 8 to ensure that your ttys are enabled with each system startup.

Removing a 7318 from a Host

Use the following information if you are removing a 7318 Model P10 from a host.

Prerequisites

1. You must have root user authority.
2. You must have a 7318 installed.

Procedure

Use SMIT to remove a 7318 from your system. Removing the 7318 also removes all serial and parallel port attached devices as well. Repeat the following steps for each 7318 you are removing from the host. (Since each port can have multiple sessions, it is not necessary to remove sessions from one host to configure them on another host.)

1. Use the **smit ts7318_cs_mnu** fast path to access the **ComNetServer Server Configuration** menu.
2. Select **Remove a ComNetServer**.
3. Select the **ComNetServer** to remove.
4. Select **Do**.

Removing 7318 Software from Version 3.2.5

The following steps will help you remove the software for upgrades or remove the package from your system.

Prerequisites

1. The software must not be committed.
2. You must have root user authority.

Procedure

1. Use the **smit install_remove** fast path to access the **Remove Applied Software Products** menu.
2. At the SOFTWARE name field, type the name of the 7318 software package to remove. You can enter **cns*.obj** to select all 7318 software packages, or select the **List** option for a list of possible software packages to remove.
3. Select **Do** to remove the package.
4. Select **Exit** to exit the SMIT interface.

Removing 7318 Software from Version 4.1

The following steps will help you remove the software for upgrades or remove the package from your system.

Prerequisites

You must have root user authority.

Procedure

1. Use the **smit install_remove** fast path to access the **Remove Software Products** menu.
2. At the SOFTWARE name field, type **devices.7318.s20**.
3. Change the PREVIEW only? field to **no**.
4. Change the REMOVE dependent software? field to **yes**.
5. Select **Do** to remove the package.
6. Select **Exit** to exit the SMIT interface.

Removing 7318 Software from AIX Version 4.2 (or later)

The following steps will help you remove the software for upgrades or remove the package from your system.

Prerequisites

You must have root user authority.

Procedure

1. Use the **smit install_remove** fast path to access the **Remove Installed Software** menu.
2. At the SOFTWARE name field, enter **bos.cns**.
3. Change the PREVIEW only? field to **no**.
4. Change the REMOVE dependent software? field to **yes**.
5. Select **Do** to remove the package.
6. Select **Exit** to exit the SMIT interface.

7318 – P10 Installation Worksheet

Use your favorite editor and the System Management Interface Tool (SMIT) to fill in the blanks from the **/etc/netware/NPSConfig** file:

Host Name	Host 1	Host 2

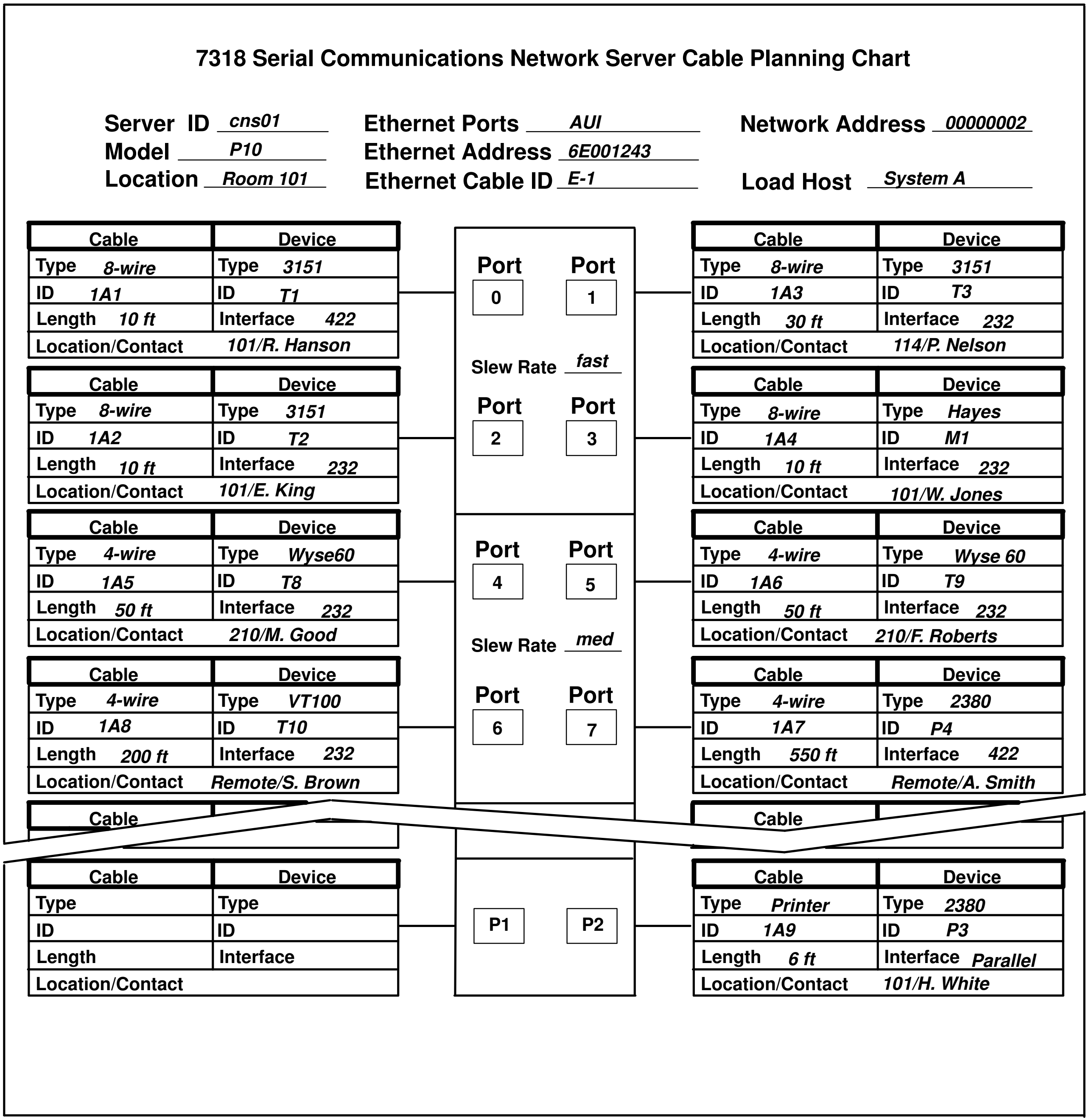
From the Host:	Host 1	Host 2
Check for `spx = active.` If `active` is not specified, change to `active` and consult your network administrator to verify the other SPX/IPX network administrator.		
Find the host's `internal_network=` number. This must be unique for all hosts on the network.		
Find Netware's Ethernet interface:		
`lan_x_if_name=`		
`lan_x_ppa =`		
`lan_x_network =`		
Fill in the Ethernet interface being used (concatenate the _if_name and _ppa fields, for example, `en0`)		
Check to see that the Ethernet interface is running. (Use the **smit chinet** command and select the correct interface.)		
(Check when complete)		
rcnetw uncommented in **/etc/inittab**		

From the 7318 Model P10:	
Serial Number	
Ethernet Address (see back of unit)	
Ethernet Port(s) connected (circle all that apply)	AUI T TX Daisy–Chain

Using SMIT:		
Device Number (01–99)		
Auto–Generated Devices	ttys only, ttys and virtuals, none	ttys only, ttys and virtuals, none

Interface Type	EIA 232 EIA 422	EIA 232 EIA 422
Host Enabled to Boot the 7318 (YES, NO)		

7318 Model P10 Cable Planning Chart: Example



The example of the 7318 Terminal Server Cable Planning Chart shows connections for six terminals, one modem, one serial printer, and one parallel printer. The IDs assigned in the example are assigned by the configuration planner. A blank chart is available on the following page.

7318 Serial Communications Network Server Cable Planning Chart

Server ID ________ Ethernet Ports ____________ Network Address ______

Model __________ Ethernet Address __________

Location ________ Ethernet Cable ID __________ Load Host __________

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 0 — Port 1

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Slew Rate ______

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 2 — Port 3

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 4 — Port 5

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Slew Rate ______

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 6 — Port 7

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 8 — Port 9

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Slew Rate ______

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 10 — Port 11

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 12 — Port 13

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Slew Rate ______

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Port 14 — Port 15

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

P1 — P2

Cable	Device
Type	Type
ID	ID
Length	Interface
Location/Contact	

Serial Communications Network Server (7318) Model P10 Configuration

The following describes how to configure terminals and multiple sessions, printers, and modems. Configuring a Model P10 for a multihost environment as well as a high–availability environment is also explained.

Installing a Terminal with Defaults

Use the following procedure to install a terminal, using the default settings. Later information gives details on modifying the terminal settings and specifics about terminal types.

Prerequisites

1. The 7318 must be installed and operational.
2. The cable from the 7318 serial port to the serial port on your terminal must be connected using an appropriate cable adapter. You can generally use a terminal adapter (Feature Code 7904).

Procedure

1. Turn on the terminal.
2. Verify settings of the terminal. Use the Setup key on the terminal (or whatever method the terminal uses to set its parameters). The settings should be:
   - 9600 baud
   - 8 data bits
   - 1 stop bit
   - no parity
   - full duplex
   - XON/XOFF flow control.
3. Enable the tty device for logins if you did not already do so during software installation.

This completes the installation of terminals that use the default settings. The following information describes other configurations, as well as commands that report the state of the terminal settings. Two additional features of the 7318 Hibaud capability and multiple sessions, are also described.

Changing Terminal Parameters

When you installed the 7318, you specified the initial configuration of the terminals. You can change the configuration for any terminal, using the System Management Interface Tool (SMIT).

Prerequisites

You must have root user authority.

Procedure

The following steps change the configuration of a terminal:

1. To change the configuration of a terminal, use the **smit chgtty** fast path.
2. SMIT prompts for the tty to configure. Scroll through the list and select the desired tty.
3. Select **Change/Show HARDWARE TTY Characteristics**. This option is valid on AIX Version 3.2.5 systems only. If you are on an AIX Version 4 system, proceed to step 4.
4. In the displayed dialog fields, select the BAUD rate entry and choose from the list of acceptable rates.

Note: The tty baud rate must be the same as the device baud rate.

5. Select **Do**.
6. Select **Exit** to exit the SMIT interface.

Enabling Hibaud Capability

The 7318 has the ability to operate at data rates in excess of the 38.4 Kbps supported by AIX. To use a port at rates above 38.4 Kbps, you must set the Hibaud property for that port. Hibaud causes certain data rates set and reported by the **smit** command to be reinterpreted as higher rates. Many modern terminals, modems, and printers support these higher speeds.

Operation of all 16 serial ports at 115.2 Kbps full duplex is not recommended using character sizes of fewer than 8 bits with 2 stop bits. To avoid data loss using character sizes of fewer than 8 bits, the following maximum data rates are recommended; 38.4 Kbps for 5–bit characters, 57.6 Kbps for 6–bit characters, and 76.8 Kbps for 7–bit characters.

The mapping of rates is shown in the following table:

Hibaud Baud Rate Mapping		
stty Reported Data Rate	**Actual Data Rate Hibaud Not Set**	**Actual Data Rate Hibaud Set**
2400	2400	57600
4800	4800	76800
19200	19200	115200

A similar mapping affects other commands that manipulate data rates such as **stty**, **getty**, **cu**, and **uucp**.

Use the **smit** command to configure Hibaud.

Prerequisites

You must have root user authority.

Procedure

The following steps change the data rate of a terminal to 115.2 Kbps:

1. To change the data rate of a terminal to 115.2 Kbps, use the **smit ts7318_port_mnu** fast path to access the **ComNetServer Port Options Configuration** menu.
2. Select **Show/Change Hibaud Configuration**.
3. Select the 7318 that has the terminal on which the tty is located.
4. Select the port or ports on which you want to configure Hibaud.
5. Select **yes** to enable Hibaud. Select **no** to disable Hibaud. Ports that display `not configured` are not yet configured ttys or printers and may not be changed.
6. Select **Do** to set the Hibaud property.

Note: When using data rates of 76.8 Kbps or 115.2 Kbps, ensure that the slew rate is set to fast. Slew rates of slow and medium will not operate properly. Slew rates must be set in groups of four ports ((0–3), (4–7), (8–11), or (12–15)).

Setting Slew Rate

The slew rate selections have been optimized around common baud rates and maximum distance. The following table describes the capabilities of each slew rate setting:

Slew Rate Settings			
Slew Rate Setting	**Max. Bits Per Second (bps) Rate**	**Distance (ft/m) RS–423/EIA 422**	**Distance (ft/m) EIA 232**
slow	9600	< 2300/701	< 200/60
medium	38400	< 625/190	< 200/60
fast	115,200	< 150/45	< 50/15

Slew rates can be set through the SMIT interface in much the same way that Hibaud is set. The default slew rate is fast. Refer to the slew rate information in ”Data Transmission”, on page C-1 for a more complete understanding.

Prerequisites

You must have root user authority.

Procedure

1. To set the slew rate, use the **smit ts7318_port_mnu** fast path to access the **ComNetServer Port Options Configuration** menu.
2. Select **Show/Change Slew Rates**.
3. Select the ComNetServer containing the port group or groups to be changed.
4. Select the **(0–3)**, **(4–7)**, **(8–11)**, or **(12–15)** port group or groups.
5. Using the **List** option, select the slew rate. The default slew rate is **fast**.
6. Select **Do** to set the slew property.

Multiple Sessions

The multiple–sessions feature on the 7318 enables a terminal to support multiple login sessions on one or more hosts. Each session acts like a unique host tty device. Multiple sessions enable you to quickly switch between different applications on the same terminal, using a hot key sequence.

When using multiple sessions, you must enable a tty for the device associated with a session. Each session retains its own logical **stty** settings. All sessions on a single physical port should use the same physical parameters, such as data rate. The **smit** command attends to these values when you configure multiple sessions.

When you switch from one session to another, the first session freezes. Output to the frozen session blocks until you resume that session.

Note: Multiple sessions can only refresh or update the screen when you switch back to a session, if the terminal supports multiple display pages. For terminals that do not support multiple display pages, the default action on a session switch is to clear the screen. This action can be modified in the configuration file. If you enable more sessions than supported display pages, the 7318 cannot perform a screen refresh for those sessions without display pages.

To switch between sessions, you must define a set of hot keys to tell the 7318 to switch sessions. The set of hot keys in use are normally associated with a particular terminal type, but they can be uniquely defined for each port independent from the terminal type. A default set of hot keys is defined for each of the terminal types known to the system.

The default configuration file has all ports configured to a terminal type of `IBM3151`. This terminal type has hot keys defined so you can use multiple sessions at a time. To use multiple sessions, you must change the default configuration file.

Using Hot Keys

The default configuration file establishes four sessions for each port, numbered 0 through 3. When hot keys are defined for a port, you can press a hot key to switch to another session.

The default configuration file also defines hot keys for certain other functions:

WHO Show what session you are currently using.

HELP Show what hot keys are defined.

STOP Hang up all sessions and exit.

QUIT End the current session only.

The hot key definitions for the defined terminal types (except for ANSI and VT100) assign these functions and the session switch functions to Shift–Function key sequence as shown in the following table.

Shift Function Keys		
Key	**Function**	**Description**
Shift–F1	Switch 0	Switch to session 1.
Shift–F2	Switch 1	Switch to session 2.
Shift–F3	Switch 2	Switch to session 3.
Shift–F4	Switch 3	Switch to session 4.
Shift–F5	Switch 4	Switch to session 5.
Shift–F6	Switch 5	Switch to session 6.
Shift–F7	Switch 6	Switch to session 7.
Shift–F8	Switch 7	Reserved for transparent print.
Shift–F9	WHO	Display current session number.
Shift–F10	HELP	Display current hot keys.
Shift–F11	STOP	End all sessions and logout.
Shift–F12	QUIT	End current session only.

To type one of these hot keys, press the Shift key and one of the Function keys on your keyboard.

Many terminals, including the IBM 3151, allow you to redefine the function keys. If the keys fail to work as expected, you should check the actual character strings that are being sent when the function key is pressed. The values in the above table assume the default values for the IBM 3151 function keys.

The hot key definitions for the ANSI and VT100 terminal types assign the session switch functions and the four special functions to a two–character sequence as shown in the following table.

Two–Character Sequence		
Key	**Function**	**Description**
Ctrl–A–1	Switch 0	Switch to session 0.
Ctrl–A–2	Switch 1	Switch to session 1.
Ctrl–A–3	Switch 2	Switch to session 2.
Ctrl–A–4	Switch 3	Switch to session 3.

Ctrl–A–5	Switch 4	Switch to session 4.
Ctrl–A–6	Switch 5	Switch to session 5.
Ctrl–A–7	Switch 6	Switch to session 6.
Ctrl–A–8	WHO	Display current session number.
Ctrl–A–9	HELP	Display current hot keys.
Ctrl–A–0	STOP	End all sessions and logout.
Ctrl–A– –(minus)	QUIT	End current session only.
Ctrl–A–Ctrl–A	FORWARD	Pass a Ctrl–A to the application.

To type one of these hot keys, simultaneously press the Ctrl key and the A key on your keyboard, and then, while still holding the Ctrl–A down, press one of the number keys or the minus key. When using this terminal type, the hot key timeout is set to infinite. This means that the hot key interpreter will wait forever for you to type the next key. To pass a Ctrl–A to your application, you must type two Ctrl–A's in sequence.

Note: The ANSI and VT100 terminal types support hot keys for only seven possible sessions.

Establishing a Terminal Type

The 7318 has the hot key data for selected terminals defined. To use hot keys on one of these terminals, set the terminal type in the configuration file as explained in ”Enabling Multiple Sessions”, on page 8-33.

The following terminal types are predefined for the 7318.

Defined Terminal Types		
Name	**Applicable Terminals**	**Notes**
dumb	Any	No function keys are recognized.
ansi vt100	VT100 VT3XX VT4XX IBM 3153 (VT100 emulation)	Uses a two–character hot key sequence, not function keys. No display pages are supported, so the screen is cleared on session switch.
wy50 wy60	Wyse 50 Wyse 60 Wyse 160	Assumes wy50+ emulation mode with enhanced mode off. Six display pages for sessions 0–5. Session 6 and 7 clear screen on session switch.
ibm3151 ibm3161 ibm3164	IBM 3151 IBM 3161 IBM 3164	No display pages are supported, so the screen is cleared on session switch.

Custom Hot Keys

If the terminal you are using is not one of the defined types, you can still use hot keys and multiple sessions. In this case, you must define the hot keys explicitly in the appropriate port section of your configuration file.

To allow multiple sessions on a particular port, you need to use additional entries in the [Port*NN*] section. Those entries are:

nhotkeys=*NN* Defines the number of hot keys to be used for switching between multiple terminal sessions on a single port.

hotkey*NN***=** *String1***,** *String2***,** *String3***,** *Word***,** *NN*
Defines the hot key sequences for this port where the *String1* parameter is a help string, the *String2* parameter is the input string to switch to this session, the *String3* parameter is the string that outputs to the terminal, and the *Word* parameter is a command function for the key:

NONE No function performed.

WHO Identifies current session.

HELP Outputs help strings.

STOP Stops session.

QUIT Quits session.

FORWARD Passes *String3* to the application.

SWITCH *NN* Key switches to defined session.

If the command is **SWITCH**, the session number must be defined by including the *NN* parameter. It is ignored by the other commands.

In the hot key definition, the first parameter, the help string, displays the key definition when the help key is entered. This string is only the key portion of the message, not the entire help message. For example, if you define a hot key to be associated with the Shift–F1 key sequence, you might enter the string "Shift–F1" as this value. This string can be a maximum of 10 characters long.

The next value, the input string, is the sequence of characters expected from the terminal or user when this hot key is entered. For terminal–derived sequences, this is normally a group of characters beginning with an escape character. For hot keys explicitly entered by a user, the input string can be one or more characters. The maximum length of the input string is six characters. All characters that can form a potential match for the input string are delayed in the 7318. When enough characters have been entered to unambiguously resolve whether the sequence is a hot key, or when the timeout expires, the string is passed to the host. In the configuration file, this string is a *quoted string*.

The next value, the output string, is the sequence of characters that echoes back to the terminal when a switch or forward hot key is recognized and passed on to the host. A maximum of 8 characters can be output. You can use this string for page switching on the terminal, clearing the terminal screen, or other functions. For a forward function, it is normally the escape key but can be any sequence of characters thus creating a simple macro capability. You can define multiple forward hot keys up to the maximum number of 12. In the configuration file, this string is a quoted string.

The following example defines a custom hot key configuration on `port 1`. Only two sessions are recognized, and no special functions are supported. The sessions are activated by typing control characters rather than function keys. The page switch in this example is for a Wyse 60 terminal.

```
[Port01]
...
nhotkeys=2
hotkey00=Control A,^A,\033w0,SWITCH,0
hotkey01=Control B,^B,\033w1,SWITCH,1
```

In addition to configuring hot key definitions, you may need to define the hot key timeout. Most terminal function keys transmit a sequence of characters that begin with the same characters used by normal applications. To distinguish between a function key and normal application keys, the 7318 only recognizes function keys if all the keys in the sequence arrive within the hot key timeout period. If the timeout period expires without the function key sequence having been entirely received, the character sequence is transmitted to the host as normal application keys.

The default hot key timeout is 0.1 seconds. You can change this if necessary by setting the `hotkeyTimeout` entry of a port section. The value should be the number of microseconds for the timeout, for example, 100000 for the default value of 0.1 seconds. If you set this value to 0, the timeout period is assumed to be infinite. In this case, characters that match the beginning of a hot key sequence will not be transmitted to the host until at least one character that differs from any of the possible hot keys is typed.

The hot keys are normally bound to function keys, which assumes that the terminal will pass the entire sequence of characters to the system in one burst. You can also define an alternate style of hot keys that involve an Escape key followed by another key that specifies the function. For this style of hot key, you should set the timeout to 0 (infinite). You should also define the **FORWARD** function so that you can pass the Escape key to the application by typing it twice.

Enabling Multiple Sessions

Prerequisites

1. You must have root user authority.
2. All users should be logged off the 7318.

Procedure

To enable multiple sessions on a tty port that does not have multiple sessions, remove the tty, then add it again with multiple sessions. The following steps remove a tty and then add a tty with multiple sessions.

1. Run the **smit** command from the AIX command line. To go directly to step 5 and skip the intervening SMIT menus, use the **smit ts**7318**_mnu** fast path.
2. Select **Devices**.
3. Select **Communication**.
4. Select **Serial Communications Network Server (ComNetServer)**.
5. Select **ComNetServer TTY Configuration**.
6. Select **Remove a TTY from a ComNetServer**.
7. Select the 7318 on which the tty is already set up from the list.
8. Select the desired tty from the list displayed.
9. Select the **Do** option to remove the tty; press Enter again when you see the prompt asking if you are sure. This completes the removal of the tty.
10. Select **Cancel** twice to get to the ComNetServer TTY Configuration menu.
11. Select **Add a TTY to a ComNetServer**.
12. Select the 7318 on which the tty was set up from the list that displays.
13. Select the available tty port that you just removed.
14. From the Available TTY Types menu, select **TTY**.
15. In the next dialog, select **yes** in the Add Associated Virtual TTYs field.
16. Select the **Do** option to create the tty devices.
17. For each of the session ttys, you can now enable the ttys as described in ”Enabling ttys”, on page 8-21. You will also need to enable the main tty.
18. Select the **Exit** option to exit SMIT.
19. Run the **cnsview –c ”show eaddr” /dev/cns***XX* command to display the Ethernet hardware address of the 7318. Alternatively, you can find the Ethernet address on the back of the 7318.

20. To set the terminal type and enable multiple sessions, copy the default P10 configuration file to the *Ethernet#***.cfg** file by typing:

```
cp /usr/lib/cns/p10.cfg /usr/lib/cns/Ethernet#.cfg
```

21. Using your favorite editor, edit the **/usr/lib/cns/***Ethernet#***.cfg** file.

22. Locate the [Port*NN*] section for the port on which you want to activate multiple sessions.

23. Uncomment the following line:

```
mscreen = 1
```

 to enable multiple sessions on the port.

24. Following the `mscreen = 1` line should be a line similar to the following:

```
terminalType = TermType
```

 where the `TermType` parameter specifies the terminal type for the port on which you are configuring multiple sessions. The function keys for the following terminals are predefined.

 – **ibm3151**
 – **ibm3161**
 – **ibm3164**
 – **wy50**
 – **wy60**
 – **vt100**
 – **ansi**
 – **dumb** (defines no hot keys).

 Refer to ”Multiple Sessions”, on page 8-29 for detailed information on the hot key definitions for the defined terminal types.

25. Issue the **cnsview –c ”reboot” /dev/cns***XX* command to force the 7318 to reload its configuration file.

Installing and Configuring Printers

The following information describes how to:

- Install printers on the 7318
- Configure a printer
- Make the physical connection to a printer
- Use transparent printing on a terminal.

The 7318 supports serial printers, parallel printers, or terminal–transparent printers. Some printers offer a choice of interfaces. When practical, parallel interfaces are the easiest to use and give the highest performance. However, parallel interfaces require close proximity between the printer and the 7318.

Serial printers permit longer distances between the printer and the 7318 but are somewhat more troublesome to configure. A terminal–transparent printer is an ASCII printer attached to a terminal that is used independently from the terminal. Terminal–transparent printers can free up a 7318 port, but offer lower performance and can be troublesome to configure.

Wiring Serial Printers

The physical connection concerns relating to printers are essentially the same as those for terminals. ”Connector Pinouts”, on page B-1 illustrates pinouts for serial printer cables.

Serial Printer Transmission

The transmission concerns for printers mirror those for terminals. Be sure to read ”Installing a Terminal with Defaults”, on page 8-27 and ”Data Transmission”, on page C-1 for more information on these topics.

Data Rate

Laser printers typically contain more powerful processors than terminals and receive data at a much faster rate. Therefore, when using a laser printer, set the data rate as high as possible. Refer to ”Enabling Hibaud Capability”, on page 8-28 and ”Setting Slew Rate”, on page 8-29 for more information on high data rate settings.

Flow Control

When using a serial printer, you must use flow control. For an EIA 232 interface, use hardware flow control when possible. Serial printers normally use Data Terminal Ready (DTR) for hardware flow control. Ensure that you have enabled the DTR line discipline. EIA 422 does not support hardware flow control; therefore use XON/XOFF flow control.

Parallel Printers

Parallel printers provide better performance than serial printers. However, the parallel interface limits the distance between the system and the printer. You can use a standard PC printer cable (Feature Code 3100) to attach a printer to the parallel port. Data rate and flow control are automatically done as a part of the parallel port protocol and cannot be configured. Refer to ”Connector Pinouts”, on page B-1 for more information on parallel port pinouts.

Installing a Printer

The following information details the SMIT steps for installing a printer.

Prerequisites

You must have root user authority.

Procedure

1. Run the **smit** command from the AIX command line. To go directly to step 5 and skip intervening SMIT menus, use the **smit ts**7318**_mnu** fast path.
2. Select **Devices**.
3. Select **Communication**.
4. Select **Serial Communications Network Server (ComNetServer)**.
5. Select **ComNetServer Printer Configuration**.
6. Select **Add a Printer to a ComNetServer**.
7. From the list, select the 7318 where you want to set up the printer.
8. Select the printer type. The list is quite long, and you may have to scroll many screens to find the desired printer type.
9. Select the printer interface (**parallel**, **EIA 232**, or **EIA 422**).
10. Select the port.
11. Select the printer type.
12. From the next dialog, ensure that values such as the configured baud rate, parity, data bits, and stop bits are the same as your printer’s data transmission values.
13. Select the **Do** option to install the printer.

Although the printer is physically attached to the 7318, it will look and act as though it is a local printer attached to the host.

Configuring a Transparent Printer for a Model P10

You can attach printers to the auxiliary EIA 232 ports of some terminals and use them as general purpose system printers for light to medium use. Activity on this printer may affect terminal performance.

The communication between the terminal and the printer must be set up at the terminal. The terminal should have a setup mode to set the communications parameters of the auxiliary port. The terminal parameters must match the printer parameters. The following list shows important parameters and recommended settings.

Parameter	Setting
Baud Rate	9600
Stop Bits	1
Parity	None
Character Size	8
Flow Control	Hardware

Note: No software flow control (XON/XOFF) is used.

To use a transparent printer without affecting terminal operation, you must prevent the printer from issuing a flow control command to the terminal. If it does, the terminal blocks and the screen no longer updates.

The transparent printer device includes a mechanism that helps prevent the printer from flow controlling. The *Rate* parameter automatically throttles the average data rate to a specified number of characters per second. The parameter does this by pausing between bursts of characters to slow down the average character rate.

Note: The parameter does not change the baud rate. Each character is transmitted at the normal baud rate. You should set the rate to the highest number of characters per second that your printer can sustain without overflowing its buffers. Normally this is much lower than the equivalent baud rate used to transmit individual characters.

You can calculate your printer's effective printing rate by measuring how long it takes to print a large file. Divide the number of characters in the file by the number of seconds it takes to print the file.

Alternatively, you can set the transparent printer device rate to an arbitrary value such as 100 characters per second, then tune it to improve performance. If you set the rate too high, your terminal will pause intermittently or your print output will be garbled. In this case, set the rate to a lower value.

Prerequisites

1. You must have root user authority.
2. The terminal must support transparent print.

Procedure

To set up a transparent printer device:

1. Use the **smit ts7318_prt_mnu** fast path to access the **ComNetServer Printer Configuration** menu.
2. Select **Add a Printer to a ComNetServer**.
3. From the list, select the 7318 that has the terminal on which the transparent printer is located.
4. Select the printer type.
5. Select **EIA 232** or **EIA 422** as the transparent printer interface.

6. Select the port on which your tty is defined. This port will drive the transparent printer. The tty must already be configured to configure a transparent printer (all virtual tty sessions must be defined for the port).
7. Select **TPRT Transparent Printer**.
8. From the next dialog screen, type the effective data rate of the printer in characters per second in the Transparent Printer Rate field.
9. In the Transparent Printer 'Enter' Sequence field, type the escape sequence (specific to the terminal to which the printer is attached) used to enter transparent print mode.
10. In the Transparent Printer 'Exit' Sequence field, type the escape sequence (specific to the terminal to which the printer is attached) used to leave transparent print mode.

    The escape sequences are used by your terminal to redirect output from the display screen to the printer port. Not all terminals have this ability. If your terminal does not have a suitable escape sequence, you cannot use transparent printing. The following table shows the escape sequences you should supply for some popular terminals.

Transparent Print Escape Sequences		
Terminal	**Enter**	**Exit**
VT100, VT200 VT220, VT300, VT420	\E[5i	\E[4i
Wyse 30	\030^X	\024^T
Wyse 50, Wyse 60, Wyse 160	\Ed#	^T
Qume/ITT 101	^R	^T\EA
Televideo	\E`	\Ea
IBM 3151	\020\022	\020\024

    The \E (backslash, E) describes the ESC character (033). Control characters are entered by preceding them with the ^ (caret) character. Thus, ^T (caret, T) means Ctrl–T. When you use the configuration file to set a terminal type and one of the built–in terminals is selected, the Enter and Exit escape sequences are automatically defined.

11. Select **Do** to configure the printer.

Modems

Modems are devices that transmit digital data over telephone lines. The following information explains how to connect, configure, and use modems with the 7318. You should install and configure modems just like terminals, using the System Management Interface Tool (SMIT) tty menus.

The following information describes:

- Configuring the 7318 for a Modem , on page 8-38
- Modem Transmission, on page 8-39

Configuring the 7318 for a Modem (AIX Version 3.2.5 Systems Only)

AIX systems use modems in two ways: for dial–in logins to a system and for dialing out to other computers. Dial–in logins allow you to call your AIX system from a terminal at home. You can configure a single modem for either function or both.

To configure a 7318 port for modem transmission, use the SMIT interface and configure the port as you would were you configuring a tty.

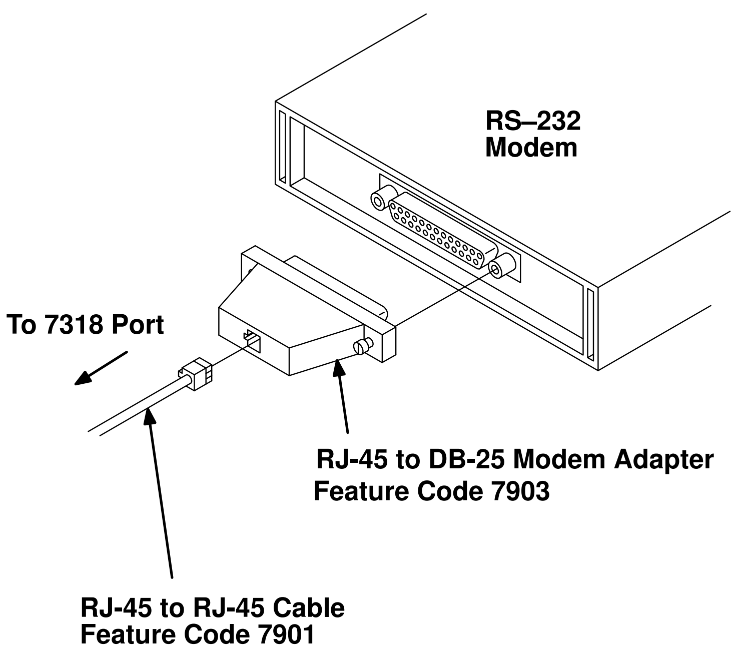
1. Use the **smit tty** fast path to access the **TTY** menu.
2. Select **Change/Show Characteristics of a TTY**.

3. SMIT prompts for the tty to configure. Scroll through the list and select the desired tty.
4. Select **Change/Show HARDWARE TTY Characteristics**.
5. Update the BAUD rate, PARITY, BITS per character, and Number of STOP BITS fields with information specific to the tty to which the modem will be connecting.
6. Select **Do**.
7. Select **Cancel** to return to the **TTY** menu.
8. Select **Change/Show Characteristics of a TTY**.
9. SMIT prompts for the tty to configure. Scroll through the list and select the desired tty.
10. Select **Change/Show TTY Program**. Use the stty attributes for RUN TIME field to specify **–clocal** so that when the 7318 loses Carrier Detect, it closes the shell.
11. Select **Do**.
12. Exit the SMIT interface.

Configuring the 7318 for a Modem (AIX Version 4 Systems)

1. Use the **smit chgtty** fast path to access the **TTY** screen.
2. SMIT prompts for the tty to configure. Scroll through the list and select the desired tty.
3. Update the BAUD rate, PARITY, BITS per character, and Number of STOP BITS fields with information specific to the tty to which the modem will be connecting.
4. Change the stty attributes for RUN time field to **–clocal** so that when the 7318 loses Carrier Detect, it closes the shell.
5. Select **Do**.

Most modems use DB–25 connectors. Use an RJ–45 to DB–25 modem adapter (Feature Code 7903) when attaching a modem to one of the 7318 serial ports. The RJ–45 to DB–25 Modem Adapter figure depicts the proper configuration for adapting from an RJ–45 plug to a DB–25 port.



RJ-45 to DB-25 Modem Adapter

Note: This configuration assumes your modem does not require DSR and RI. If it does, use the buddy–mode adapter and cabling. Refer to ”Configuring Buddy Mode”, on page 8-41 for information on buddy mode.

Modem Transmission

Transmission concerns for modems are similar to those of terminals. Refer to ”Installing a Terminal with Defaults”, on page 8-27 and ”Data Transmission”, on page C-1 for more information on these topics. Data transmission involves:

- Data rate
- Parity
- Stop bits
- Flow control

Data Rate

Data rate is the speed at which data travels between the modem and computer, measured in bits per second. To successfully transmit data, both devices must transmit at the same data rate; otherwise you may get garbled data or no data at all.

When you use modems in conjunction with the 7318, you may deal with as many as three data rates:

- The data rate between the 7318 and the modem
- The rate at which the modem sends data over the phone wire
- The data rate of the corresponding equipment (modems) on the other side of the connection.

With current modems, these rates are not necessarily the same.

7318 to Modem Data Rate

The 7318 supports baud rate cycling; that is, it has the capability to step through different baud rates when negotiating a connection.

You can configure the data rate between the 7318 and the modem in one of two ways:

- Set at a fixed rate
- Set dynamically

You should set the 7318–to–modem data rate to a fixed rate. This eliminates one potential source of errors. When using a fixed rate, flow control must be operating properly in both directions.

There are two instances when you may have to configure the modem to dynamically set the data rate:

- When using an older modem that does not support flow control
- When using Hayes or Hayes–compatible modems that automatically set their data rate based on the Hayes AT commands.

Historically, older modems transmitted data at the same rate at which they sent data over the phone wire. UNIX was unable to determine the rate over a phone wire; therefore UNIX had to be configurable for a different rate.

Some Hayes and Hayes–compatible modems automatically set their data rate based on AT commands. Though AT commands are normally used for dial–out operations, you can explicitly send AT commands to the modem by directly connecting the modem. In this case, the modem sets its rate to the host’s rate.

Data Rate over a Phone Wire

The modem controls the data rate sent over the wire. For example, a 2400–baud modem can send data at a maximum rate of 2400 baud. Current modems, however, negotiate the transmit data speed. This speed can be a function of the receiving modem or the line conditions. Furthermore, modems with V.42 or V.42*bis* capability implement compression on the data they transmit.

This means they can effectively send data at a higher rate than their basic rate when viewed from the 7318's point of view. Because of these variables, you should set the modem to the fastest speed the modem supports. Use flow control to manage the rate compensation.

Data Rate on the Remote Side

The remote side of the connection faces the same data rate concerns as the local side, that is, the data rate between the 7318 and modem, and the data rate over the phone wire. Just because you use 9600 baud on one end does not necessarily mean the other end is also using 9600 baud.

Parity

Parity is a coding scheme for checking the validity of data characters. An extra bit transmitted with each character indicates whether the sum of the other bits in the character is even or odd.

For a modem, parity is not as important as it is for other devices, such as terminals. Modems use an error correction and detection scheme that is more sophisticated than parity on the phone line. Also, data errors are less likely on the short cable typically used between a modem and the 7318.

The parity setting on the AIX host connected to the 7318 must match the parity setting of the device to which you are speaking. The modem transparently replicates the parity through the phone system, so the settings on both sides must agree.

Stop Bits

Stop bits delineate the end of a character in the data stream. Use one stop bit unless the device requires otherwise.

Flow Control

Flow control paces the data transmission so that the receiver has a chance to process all of the incoming characters before the transmitter sends additional data.

For modems, flow control can occur in three places:

- Between the 7318 and a modem
- Between modems
- Between modem and terminal.

Between the 7318 and a Modem.

Although you can use either hardware or software flow control between the 7318 and a modem, you should use hardware flow control. This leaves the Ctrl–S and Ctrl–Q key sequences available to applications, file transfers, and to users for managing screen data.

Between Modems

Modems use a form of flow control between themselves. This is a function of modem standards and does not require you to set any parameters.

Between Modem and Terminal

Although you can use either hardware or software flow control between the modem and a terminal, you should use hardware flow control. This leaves the Ctrl–S and Ctrl–Q key sequences available to applications, and to you for managing screen data. If you communicate with another computer, using an internal modem, this is not an issue.

Using Software Flow Control

You can configure a modem for XON/XOFF flow control. You should use software flow control only when your cabling or modem cannot support the modem control signals for hardware flow control.

Software flow control prohibits the use of the Ctrl–S and Ctrl–Q key sequences from the keyboard, because the system confuses the user's Ctrl–S and Ctrl–Q key sequence with those issued by the modem.

Note: Software flow control should not be used on interfaces performing UUCP transfers. UUCP can send binary data that includes Ctrl–S and Ctrl–Q characters in the data.

Slew Rate

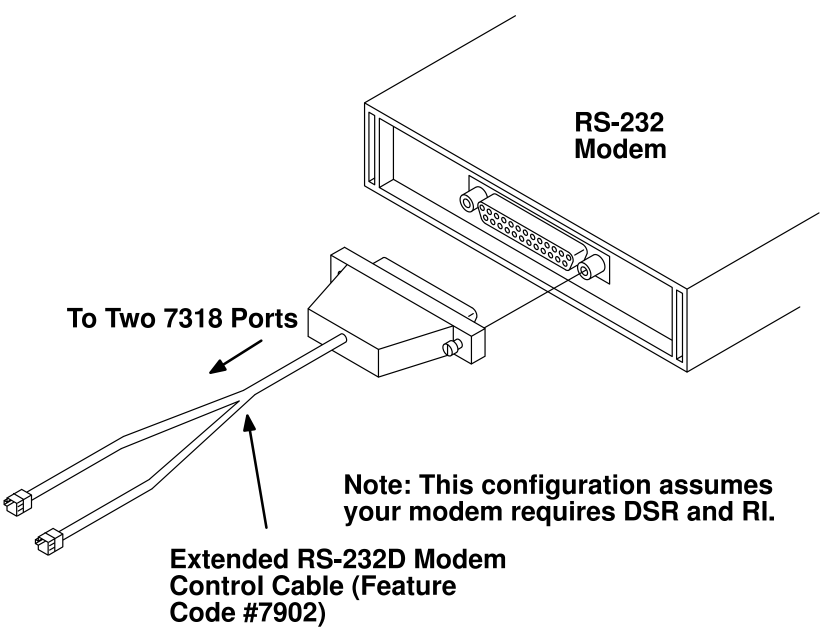
If the modem is closer than 150 ft to the 7318, slew rate is not a consideration and should be set to the default. If the modem is farther than 150 ft from the 7318, refer to "Setting the Slew Rate", on page 8-29 to change the slew rate to the appropriate setting. Refer to "Data Transmission", on page C-1 for more general information.

Configuring Buddy Mode

Two common EIA 232 signals not provided on the normal port pinouts are DSR and RI. For most applications today, these signals are not used. However, for those applications that do require these signals, the 7318 provides *buddy mode*.

In buddy mode, two adjacent ports are paired to supply the additional DSR and RI signals. The ports must be an even–numbered port and the next–higher odd–numbered port (for example, 0 and 1, or 8 and 9). Buddy mode redefines the pins on the the two ports to provide DSR and RI. The two ports become a single, logical port, which is configured on a port–by–port basis.

The Extended EIA 232D Modem Control cable must be used to connect two RJ–45 connectors to a single DB–25 connector (Feature Code 7902). Refer to the Extended EIA 232D Modem Control Cable figure for a graphic illustration.



Extended RS-232D Modem Control Cable

Refer to "Connector Pinouts", on page B-1 for details on RS–423, modem, and extended EIA 232D modem control pinouts.

The following information explains the SMIT steps used to configure buddy mode for the 7318.

Prerequisites

You must have root user authority.

Procedure

1. Use the **smit ts7318_port_mnu** fast path to access the **ComNetServer Port Options Configuration** menu.
2. Select **Show/Change Buddy Cable Configuration**.
3. Select the 7318 on which to configure buddy mode.
4. From the next menu, select the port pair to configure in buddy mode and change the value to **yes**.
5. Select **Do**.
6. Select **Exit** to exit the SMIT interface.
7. Enter the **smit ts7318_tty_mnu** fast path to access the **ComNetServer TTY Configuration** menu.
8. Configure the 7318 TTY. Refer to ”Generating a Single 7318 tty Device”, on page 8-21 for details on configuring the 7318 tty.

   Note: Configure a tty for the even–numbered port only.

Enabling Hardware Flow Control for Version 3.2.5

Hardware flow control requires a cable that uses the modem pinout shown in ”Connector Pinouts”, on page B-1. If you use a cable with only send and receive connected, hardware flow control will not work.

Prerequisites

1. This procedure assumes that you are enabling hardware flow control on a tty that has already been configured. If the tty has not been configured, refer to ”Configuring a 7318 Model P10”, on page 8-20.
2. You must have root user authority.

Procedure

To enable hardware flow control:

1. Enable hardware flow control on the modem.
2. Make sure that the cable includes a circuit from the flow control pin on the modem—usually data transmit ready (DTR) or request to send (RTS)—to the Clear to Send (CTS) circuit on the 7318. The 10 ft RJ–45 cable (Feature Code 7901) and modem adapter (Feature Code 7903) provide this function.
3. From the terminal associated with the port, type:

   ```
   stty add rts
   ```

   and

   ```
   stty add dtr
   ```

Enabling Hardware Flow Control for AIX Version 4

The following procedure describes how to enable hardware flow control for AIX Version 4.

Note: If all flow control on a 7318 P10 port is disabled, unpredictable results can occur. Large amounts of data throughput can overflow the P10 internal buffers. Disabling flow control is not recommended.

Prerequisites

1. This procedure assumes that you are enabling hardware flow control on a tty that has already been configured. If the tty has not been configured, refer to ”Configuring a 7318 Model P10”, on page 8-20.
2. You must have root user authority.

Procedure

1. Use the **smit chgtty** fast path to access the **TTY** screen.
2. Select the tty to be changed.
3. Change the FLOW CONTROL to be used field to **rts**.
4. Select **Do**.
5. Exit the SMIT interface.

Configuring Ports for Dedicated or Multiple Hosts

The 7318 Model P10 has the unique ability to be shared by more than one system. Sharing a Model P10 between hosts can be done in two ways. First, individual ports can be directed to separate hosts. Second, individual port's sessions can be directed to separate hosts. In the first case, each of 16 users could access a different host from the same 7318. In the second case, an individual port or user can access a different host on each of its seven sessions.

Dedicate P10 Ports to Different Hosts

The following information describes how to configure a host for a port–to–host type of connection.

Prerequisites

This procedure assumes that the Model P10 is already configured with all ports on one host, the load host. If not, take the steps necessary to configure all ports on the load host. Make certain that all ports are operational to the load host before configuring additional hosts.

Procedure

1. Remove the port or ports to be used by another host from the first host (assumed to be the load host). On the load host, remove the tty or LP device from the 7318, using the **smit ts7318_tty_mnu** fast path or the **smit ts7318_prt_mnu** fast path.
2. Select **Remove a TTY from a ComNetServer** or **Remove a Printer from a ComNetServer**.
3. From the lists, select the appropriate 7318 and port.
4. Select **Do**.
5. Add the 7318 to the second host. Refer to ”Configuring a 7318 Model P10”, on page 8-20 to add a 7318 to the host and the network and to generate ttys.

   Ensure that the host's internal network number in the **/etc/netware/NPSConfig** file is unique on the network.

   Although it is possible to select to load from the second host, you should not load from two hosts unless it is required for availability reasons. Loading the 7318 from two hosts will complicate network administration.

   Select **None** in the Autogenerated TTY Devices field.
6. Now that the 7318 is known to the second host, the individual tty or LP device can be added to this 7318 from the Add a TTY to a ComNetServer or Add a printer to a ComNetServer SMIT menu. Select the ports desired for connection to this host.

7. Additional hosts can be added to a 7318, using steps 1 through 6. You should verify the operation of the newly configured ports before each new host is added.

Configure a Single P10 Port for Multiple Hosts (Multisession)

The following information describes how to configure a host for a session–to–host type of connection.

A 7318 Model P10 port with six virtual tty devices has a total of seven sessions, numbered 0 through 6. When virtual tty devices are created, they are suffixed with a letter indicating which session they are associated with. The following table shows the association among seven sessions for port 0 for each of two hosts.

7318 Port/Device/Session Association				
Host	**Session 0**	**Session 1**	**Session 2 . . .**	**Session 6**
Host1 Device Names	tty0	tty0a	tty0b	tty0f
Host2 Device Names	tty32	tty32a	tty32b	tty32f

Because session 0 on this port can only be associated with a single host, only one of the tty devices in the first column (Host1:tty0, Host2:tty32) should be enabled. Enabling more than one tty device on a port/session causes unpredictable results. You should enable either Host1:tty0 and Host2:tty32a or Host1:tty0a and Host2:tty32 to prevent a conflict. The unused devices and sessions remain present but nonfunctional.

Prerequisites

This procedure assumes that the Model P10 port is already configured with multiple sessions on one host, the load host. If not, take the steps necessary to configure the port with all seven virtual ttys on the load host. Make certain that all sessions are operational to the load host before proceeding to configure additional hosts. To validate all sessions, enable the tty and virtual ttys and toggle to them using hot keys.

Procedure

1. Disable the specific virtual tty on the load host to be used by another host. On the load host, issue the **pdisable /dev/tty***XXY* command where **/dev/tty***XXY* is the applicable virtual tty. The *Y* parameter specifies an alphabetic character, a through g, appended to the tty digits *XX* indicating which of the seven sessions it corresponds to. For example, the third virtual tty for `tty31` would be `tty31c`.
2. Add the 7318 to the second host. Refer to ”Configuring a 7318 Model P10”, on page 8-20 to add a 7318 to the host and the network and to generate ttys.

   Ensure that the host's internal network number within the **/etc/netware/NPSConfig** file is unique on the network.

   Although it is possible to select to load from this second host, you should not load from two hosts unless it is required for availability reasons. Loading the 7318 from two hosts can complicate network administration.

   Select **None** in the Autogenerated TTY Devices field unless all 16 of the 7318 ports will have sessions shared by these two hosts. If you select to autogenerate ttys, advance to step 4.
3. Now that the 7318 is known to the second host, the individual ttys can be added to this 7318 with the Add a TTY to a ComNetServer SMIT options. Select to generate ttys and virtual ttys for this port.
4. At this point, both hosts have seven sessions configured for the same port. Ownership of sessions is controlled by enabling only those virtual ttys associated with that host. Refer to ”Enabling ttys”, on page 8-21 to enable the virtual ttys. It is important to configure the

ttys the same on both hosts. For example, baud rate for a port should be set to the same value for all sessions on both hosts.

5. Additional hosts can be added to a port by repeating steps 1 through 4, making certain to enable only those virtual ttys on the desired host. You should also verify proper port and session operation after configuring each host. Do not attempt to configure more than one host before verifying proper operation. The system administrator should document the tty device numbers for each port session and the controlling host name. This is recommended because the tty device names for a 7318 port, **/dev/tty***XX*, are not always the same on all hosts.

Enabling Serial Line Internet Protocol (SLIP) on a Model P10 tty

The following information describes the necessary steps for configuring SLIP on a tty.

Prerequisites

1. You must have root user authority.
2. No sessions should be configured on the 7318 port running SLIP.

Prepare the Model P10 for a SLIP Connection

1. Use the **smit tty** fast path to access the **TTY** menu.
2. Select **Change/Show Characteristics of a TTY**.
3. SMIT prompts for the tty to configure. Scroll through the list and select the desired tty.
4. Select **Change/Show TTY Program**. (This field is valid on AIX Version 3.2.5 systems only.) If you are an AIX Version 4 user, proceed to step 5.
5. Delete **icanon** and **opost** from the stty attributes for RUN TIME field.
6. Change the FLOW CONTROL to be used field to **rts** or **none**. (This field is valid on AIX Version 4 systems only.) If you are an AIX Version 4 user, proceed to step 13.

Steps 7–12 apply to AIX Version 3.2.5 systems only:

1. Select **Do**.
2. Select **Cancel** to return to the **TTY** menu.
3. Select **Change/Show Characteristics of a TTY**.
4. SMIT prompts for the tty to configure. Select the same tty as in step 3.
5. Select **Change/Show Hardware Characteristics of a TTY**.
6. Select **XON–XOFF handshaking** and change the value to **no**.
7. Select **Do**.
8. Select **Exit** to exit SMIT.
9. On AIX Version 3.2.5 systems only, issue the following command:

   ```
   stty get < /dev/P10TTY
   ```

   If `xon` is one of the outputs on the stack, issue the following command:

   ```
   stty del xon < /dev/P10TTY
   ```

Create a SLIP Interface and Start SLIP

1. To create a SLIP network interface on the local host, use the **smit mkinet** fast path to access the **Add a Network Interface** menu.
2. Select **Add a Serial Line INTERNET Network Interface**.
3. From the TTY PORT for SLIP Network Interface screen, select the tty changed in the preceding procedure.

4. Specify the IP address of the local host in the INTERNET ADDRESS (dotted decimal) field.
5. Update the DESTINATION Address (dotted decimal) field with the address of the remote host.
6. Change the Network MASK, BAUD RATE, and DIAL STRING fields if desired.
7. Select **Do**.
8. Exit the SMIT interface.
9. To create a SLIP network interface on the remote host, repeat steps 1–7, except exchange the information in the INTERNET ADDRESS and the DESTINATION Address fields.
10. Select **Do**.
11. Exit the SMIT interface.
12. Add the following two entries to the **/etc/hosts** file on both the local and remote hosts:

    `IPAddress` *`LocalHostName`*

    `IPAddress` *`RemoteHostName`*

    The name you assign should be unique. In other words, if the Ethernet interface on the local host is already assigned the name `local`, assign the SLIP interface a name such as `local_slip`.
13. If you are configuring SLIP over a modem or if you are configuring SLIP over a null modem cable, add the following line to the **/usr/lib/uucp/Devices** file on both the local and remote hosts:

    `Direct` *`TTY`* `- ModemBaudRate direct`

    If you are configuring SLIP over a null modem cable. The *TTY* parameter is the tty specified in step 3.
14. If you are configuring SLIP over a modem, start SLIP on the local host by running the command:

    `slattach` *`TTY`* `ModemBaudRate`

    The *TTY* parameter is the tty specified in step 3.
15. If you are configuring SLIP over a modem, start SLIP on the remote host by running the command:

    `slattach` *`TTY`* `ModemBaudRate ' "" AT OK ATDT555-1234 CONNECT "" '`

    The meaning of this command is: Use the tty specified by the *TTY* parameter at `ModemBaudRate` baud. Send `AT` to the modem. The modem should respond with `OK`. Dial the phone number `555-1234`. The modem should respond with `CONNECT`.
16. If you are configuring SLIP over a null modem cable, issue the following command on both the local and remote hosts:

    `slattach` *`TTY`*

    The *TTY* parameter is the tty specified in step 3.
17. To add the necessary routes on the local host, run the **route add** *RemoteIPAddress LocalIPAddress* command.
18. To add the necessary routes on the remote host, run the **route add** *LocalIPAddress RemoteIPAddress* command.
19. To test the SLIP connection on the local host, run the **ping** *RemoteIPAddress* command.
20. To test the SLIP connection on the remote host, run the **ping** *LocalIPAddress* command

If both tests succeed, the SLIP connection is ready for use. If not, return to step 1 and verify that the configuration on both the local and remote hosts is correct.

7318 High–Availability Environment for the Model P10

The following information describes the 7318 HA feature, which can be used with High–Availability Cluster Multi–Processing (HACMP) servers. HACMP enables the creation of a cluster of loosely coupled systems.

Introduction

The Internet Protocol Exchange (IPX) version of the High–Availability (HA) feature of the 7318 can be used with a HACMP/6000 cluster environment. When there is a redundant network, this feature allows the 7318 to recover from a network failure.

The HA feature is designed to allow a 7318 to physically attach to two distinct local area networks (LANs), using both its 10BaseT and AUI port connectors. Once attached, the 7318 will be given the address of a host or hosts that will determine which network to use.

The 7318 will come online and automatically switch to the interface configured as the primary interface. It will then begin to ”ping” the host. As long as those pings are returned, the 7318 will remain on that Ethernet. If, however, the pings stop being returned, the 7318 will switch (after a length of time) to the backup interface. It will then try to ping the host again. If this succeeds, the 7318 will stay on that port. If it fails immediately, or after some amount of time, the 7318 will switch back to primary port and stay there until a ping is returned.

The HA feature is enabled by editing a default configuration file, supplied in the **/usr/lib/cns** directory. For the Model P10, this configuration file enables the HA feature and multiple sessions. ”Configuring the High Availability Environment for the Model P10”, on page 8-50 details the steps to edit the configuration file.

The IPX version of the HA feature uses an IPX echo packet to determine if the host is online. In the IPX case, there is one IPX address for the 7318 and one for the host.

Definitions

The following definitions are important in understanding the IPX version of the HA feature:

Primary host Specifies the host used to determine the operability of the Ethernet. In configuring the HA feature, it will be necessary to know the IPX address of that host.

Backup host Specifies the host used to determine the operability of the Ethernet. In configuring the HA feature, it will be necessary to know the IPX address of that host.

Primary interface Specifies the first interface on the 7318 used by the HA feature.

Backup interface Specifies the second interface on the 7318, which is used by the HA feature if pinging fails on the primary interface.

Ping interval Specifies the time periods between pings.

Switch to backup Specifies the number of seconds that the HA feature keeps the 7318 on the primary interface without pings, before switching to the backup interface.

Switch to primary
Specifies the number of seconds that the HA feature keeps the 7318 on the backup interface without pings, before switching to the primary interface.

IPX Network Address Format
The Model P10 HA configuration uses IPX network addresses (internal address). The IPX address consists of a network and a node number (Ethernet hardware address). These consist of two hexadecimal numbers (acceptable hexadecimal values are 0–9 and a–f). The first number is 8 digits long. The second number is 12 digits long, and normally set to 1 for host systems. The network is separated from the node by a : (colon). The use of hexadecimal in an IPX address is implied. A typical IPX address might look like the following:

```
00101491:00406E000001
```

General Operation of the HA Feature

The 7318 initializes on the primary interface when the HA feature is enabled. Once started, the HA module will "ping" the primary host address using the IPX protocol at a rate determined by the `PingingInterval` entry. If the ping is not returned, the Switch to Backup timer will start. If that timer expires without receiving any ping response, the HA module will switch the interface to the backup interface.

Once again, it will ping the backup host. If the ping is not returned, the Switch to Primary timer will be started. If that timer expires without receiving any ping response, the HA module will switch the interface back to the primary interface. At this point, the HA module will again start pinging the primary host.

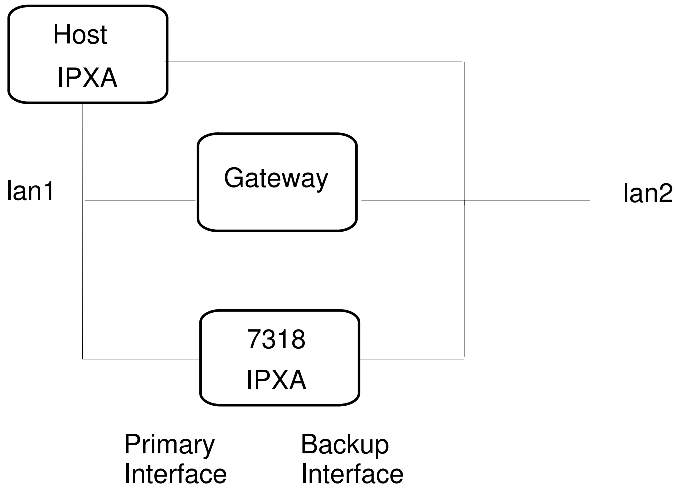
However, in this state, the 7318 will stay on the primary interface until ping responses are received (for example, the interface will not keep switching in the case where there is not response from either port). Once a ping response is received in this state, the HA module resets and returns to its initial state where an absence of ping response will again cause the interface to switch.

Note: In the HA mode, the 7318 will not respond to physical level information from the LAN port. The decision to switch the interface is based only on the ping response from the respective host.

Configuration File Format of the Model P10 HA Feature

There are two sections in the configuration file which relate to the HA feature, the [HighAvail] section and the [IPX] section, and an optional section, [SPX]. The [HighAvail] section enables and configures the HA feature. The [IPX] section specifies an internal network number so that two distinct networks can be configured.

The IPX Version of the HA Feature figure graphically shows the layout of a 7318 attached to two separate IPX interfaces.



IPX Version of the HA Feature

[HighAvail] Section

The [HighAvail] section contains the following entries:

enable=0/1	Disables or enables the HA feature on the 7318. Acceptable values are 0, which disables the HA feature, and 1, which enables it. Setting the `enable` entry to 0 is useful for turning the HA feature off, while saving the rest of the information in the configuration file.
addressType=ipx	Specifies the type of ping protocol used by the 7318 to ping the primary and backup hosts.
primaryhost=*Address*	Specifies the IPX network address of the primary host.
backuphost=*Address*	Specifies the IPX network address of the primary host.
	Note: The addresses of the primary host or the backup host can be the same on both interfaces or you can choose to ping different hosts. Normally you should use the internal address of the host, not its Ethernet address.
SwitchToBackup=*Integer*	Specifies the integer time in seconds before the 7318 switches from the primary to the backup Ethernet port in the case of failure.
SwitchToPrimary=*Integer*	Specifies the integer time in seconds before the 7318 switches from the backup Ethernet port to the Primary port, in the case of failure.
PingingInterval=*Integer*	Specifies the time in seconds between pings.
PrimaryInterface= AUI/10baseT	Specifies either the primary or backup interface.
BackupInterface= AUI/10baseT	Specifies either the primary or backup interface.

Note: Do not configure both primary and backup interfaces to be the same address.

[IPX] Section

If the high–availability option is being used, and the Internet Packet Exchange (IPX) protocol is being used to ping the host, the 7318 IPX addressing must be changed to allow for handling connection to two different LANs. To do this, an internal network address must be specified for the 7318 (this is similar to the host). The [IPX] section contains the following entry:

internalNet=*Address*
Specifies the IPX network address. The address takes the following address format:

`nnnnnnnn:dddddddddddd`

where `nnnnnnnn` is a hexadecimal representation of the internal network number, and `dddddddddddd` is a hexadecimal representation of the Ethernet hardware address.

[SPX] Section

The [SPX] section contains the following entries:

idleTimeout=*N*
Specifies the time in seconds to wait for a keepalive response before the link drops, assuming no data is being exchanged. The default is 120 seconds. The **idleTimeout** value should be at least twice the **idleInterval** value.

idleInterval=*N*
Specifies the time in seconds between keepalive messages, that are used to keep the link alive when no data is being exchanged. The default is 30 seconds.

transmitRetries=*N*
Specifies the number of data retries on the transmit side. The default is 20.

receiveRetries=*N*
Specifies the number of data retries on the receive side. The default is 20.

Configuring the High–Availability Environment for the Model P10

Configuring the HA feature involves editing the default configuration file and uncommenting appropriate lines.

Prerequisites

You must have root user authority.

Procedure

1. Copy the default configuration file to a 7318–specific file by typing:

   ```
   cp /usr/lib/cns/p10.cfg /usr/lib/cns/7318EtherAddress.cfg
   ```

   where the 7318`EtherAddress.cfg` parameter specifies the Ethernet hardware address of the 7318 to be configured for HA.

2. Uncomment the following entries in the 7318 default configuration file and change the values as required for your configuration:

```
[HighAvail]
enable=0

;addressType=ipx

;primaryhost=00101492.000000000001
;backuphost=00101492.000000000001

;Timeouts in seconds
;switchToBackup=90
;switchToPrimary=90
;pingingInterval=20

;The following interfaces should be set to 10BaseT or AUI.
;The BackupInterface should be the opposite of the primary.
;primaryInterface=10baseT
;backupInterface=AUI

[IPX]
;The internal network is only used when HighAvail is enabled for
ipx.
;internalNet=01020304:00406E000001

[SPX]
;If no data and no keepalives are received in idleTimeout
seconds,
;the link will drop.
;idleTimeout=15

;The following interval specifies the time between keepalive
messages.
;These are used to keep the link alive when no data is being
exchanged.
;The idleTimeout should be at least twice the idleInterval value.
;idleInterval=7
```

Serial Communications Network Server (7318) Administration

This section describes the BIOS console and its configuration commands, its diagnostic and troubleshooting procedures, and the procedures for logging load events during loader execution. For information on changes to the parameters of a 7318 and its associated devices and drivers, see the **cnsview** command in the *AIX Commands Reference, Volume 1*.

7318 BIOS

The 7318 includes a simple command interpreter, the basic input/output system (BIOS), in nonvolatile random access memory (NVRAM), which permits configuration even when no other software is loaded.

The BIOS has four main functions. The most basic function is to obtain and execute an operational kernel from a load host. This load sequence is done automatically. Once you plug in the 7318, the BIOS runs until an operating image downloads into the 7318 RAM and takes control from the BIOS.

The second function is to verify the basic functionality of the 7318 hardware *before* the load image is downloaded and executed. The BIOS does this using power–up diagnostics, which disallow a download if the platform is deemed unstable. An unstable platform might be able to run the operational kernel; however, the operation might be erratic.

The third function is to run post power–up diagnostics (PPDs), which are used for field troubleshooting of a 7318 that might pass power–up diagnostics but operate improperly. Normally, with the BIOS, you can take a 7318 out of the shipping carton, attach it to the Ethernet, plug in the line cord, install the software, and download.

The fourth function provides a console to set various nonvolatile fields stored in the NVRAM of the 7318. The following partial list specifies key fields you can set using the BIOS console:

- Download image
- Host from which the 7318 loads
- Protocol used to download
- Software configuration file to load
- High–availability environment parameters.

Notes:

a. The default settings of the NVRAM fields do not require the use of the BIOS console. Interaction with the BIOS is usually reserved for exceptional circumstances when problems are encountered or for modification of the load sequence.

b. The BIOS console is designed for system administrators who understand both the 7318 and networking. Most end users do not interact with this portion of the 7318.

BIOS Download Overview

The BIOS download has a general sequence of operation as follows:

1. At power–up, the power–up diagnostics are run.
2. If successful, the BIOS checks for a console request.
3. If the console is requested, the BIOS prompts for input from the user.
4. If the console is not requested, the BIOS seeks an operational kernel, downloads, and runs it.
5. The kernel seeks a configuration file and downloads it.

Once the BIOS has successfully downloaded, it puts the hardware in a known state and runs the kernel. The BIOS does not regain control until one of the following:

- The 7318 is powered down, then up.
- A reboot command is issued from the operational kernel.
- The kernel watchdog timer expires, and the 7318 resets.

In the case of a reboot, the BIOS can either reload the normal operational image or reload a directed image (using the **set loadi** command).

BIOS Upload Sequence

In the case of certain reboot commands, and in the case when the watchdog timer expires and the NVRAM upload field is set (using the **set upload** command), the BIOS attempts to upload a dump image, which then can be used as a diagnostic tool.

When uploading, the BIOS operates differently than previously described. The sequence is:

1. The watchdog timer expires, and the 7318 resets itself. The watchdog will not expire unless there is a software or hardware failure. The BIOS examines memory contents to distinguish between a cold start and a watchdog timer restart. If the memory appears to have an image loaded, the BIOS assumes that the watchdog timer has expired.

   In the event of a brief power outage, memory may not erase sufficiently for the BIOS to recognize that a power failure has occurred instead of the watchdog timer timing out. The BIOS may decide to upload even though the system is operating normally. Log records from this type of upload can be ignored.
2. The BIOS does a very low–level diagnostic and loads fields from the NVRAM.
3. The BIOS checks for a console request.
4. If the BIOS determines that an upload is necessary, it immediately uploads the image.
5. Whether or not the upload succeeds, the BIOS resets the 7318.
6. The BIOS proceeds as if the 7318 had been powered up.

The BIOS sends the upload image to the /**usr**/**lib**/**cns** directory, where it creates the *XXXXXXXX***.uld** file. The *XXXXXXXX* parameter is the lower eight digits of the 7318 Ethernet address.

7318 BIOS Load Sequence

The `interface` and `protocol` fields control the process by which the 7318 locates a load image. If the `protocol` field is set to 0, the Internet Packet Exchange (IPX) and User Datagram Protocol/Internet Protocol (UDP/IP) sequences alternate, beginning with the IPX sequence, until one of them succeeds. The `inet`, `loadi`, `host1`, and `host2` fields are also used in the configuration process as the following information details.

7318 Load Sequence Using IPX

The load sequence proceeds as follows:

1. The first step has one of two possibilities:
   a. An undirected load issues a Service Advertising Protocol (SAP) message and places the responses in a host list.
   b. A directed load initializes the host list from the `host1` and `host2` fields in NVRAM.
2. The 7318 issues an IPX Routing Information Protocol (RIP) request to get network information for each host in the host list.
3. The 7318 issues an IPX/BOOTP request, one at a time, and listens for replies for each host on the host list.
4. When a valid Boot Request Protocol (BOOTP) response is received, a Trivial File Transfer Protocol (TFTP) request is sent for the download image to the responding host. In the case where the `loadi` field in the NVRAM is `default`, the name of the file contained in the BOOTP response is used for the TFTP request. Otherwise, the `loadi` field specifies the load image for the TFTP request.

7318 Load Sequence Using UPD/IP

If the criteria for the *directed* load are satisfied, it runs. Otherwise, the *undirected* load runs.

With the directed UDP/IP load sequence, the following fields must be set:

- `host1` and/or `host2`
- `inet`
- `loadi`

The undirected UDP/IP load has two steps:

1. The 7318 issues a BOOTP broadcast request to locate a valid load host.
2. When a load host responds, fetch the load image file using the TFTP protocol.

The directed UDP/IP load has two steps:

1. The 7318 issues an Address Resolution Protocol (ARP) message to the directed IP address.
2. The 7318 issues a TFTP request for the load image specified by the `loadi` field (from the host or hosts specified by the `host1` or `host2` fields).

7318 Configuration File Load Sequence

After the operating kernel loads, the BIOS tries to load a configuration file in the following order:

1. If the `config` field is set, the BIOS tries to load that configuration file. If this fails, no configuration file loads and an error message displays.
2. If the `config` field is set to `default`, the BIOS uses the following sequence to locate a default configuration file:
   a. The BIOS uses the download path and the 12–digit form of the Ethernet address as the file name (hexadecimal digits a–f in lowercase, for example, `00406e0002e3.cfg`).
   b. If step 2a fails, the BIOS uses the download path and the 8–digit form of the Ethernet address as the file name (hexadecimal digits a–f in lowercase, for example, `6e0002e3.cfg`).
   c. If step 2b fails, the BIOS uses the download path specified in the downloaded image.

   Failure to find any of the three files results in an error message. For the Model S20, this failure is fatal, and the 7318 will immediately attempt to reload. For the Model P10, this failure is nonfatal; however, all functionality defined by the configuration file remains inactive.

BIOS Console Operation

The BIOS console allows a terminal to communicate with the 7318 BIOS in an interactive way. Using the BIOS console, the system administrator can change certain basic 7318 fields as well as customize the load sequence.

If you need to use the BIOS console on the 7318, do the following:

1. Unplug the 7318.
2. Connect an ASCII terminal to any one of the 7318 serial ports.
3. Set the terminal to 9600 baud, 8 data bits, one stop bit, no parity.
4. Plug in the 7318.
5. Type `#` (pound sign, ASCII 0x23) continually until the console echoes `####` (four pound signs).
6. The BIOS should respond with a console header and the `>` (redirect symbol) prompt.

The BIOS console interaction follows the same form for all commands.

The **set** command, which changes fields, interacts as follows:

```
#>set Field Parameter
```

giving the response:

```
Error: Message
```

OR

```
OK
#>
```

An indication of success or failure always follows a **set** command, that is, either the text string `Error:` and a message, or `OK` displays. The BIOS always supplies a command prompt for input.

For fields that are informational, such as the **show** command, the interaction is:

```
#>show Field
```

Response ...

```
#>
```

The response to the **show** command is specific. Responses to the other informational commands have a variety of output formats.

The BIOS console has two different privilege levels. These privilege levels allow the user different access to the NVRAM in the 7318. The modes are:

- Normal mode
- Password privilege mode

In password privilege mode, the BIOS console operator can change certain NVRAM fields necessary for reconfiguring the 7318.

The BIOS requests input by issuing a prompt. The prompt has the following forms:

```
>
#>
```

The prompts correspond to normal mode and password privilege mode, respectively. To enter password privilege mode, use the **admin** command. For more information on the **admin** command, refer to "admin Command", on page 8-61.

Note: All **set** subcommands run only in password privilege mode.

BIOS Console Configuration Commands

The BIOS console is a simple interactive command–response language. Typed characters echo to the console and can be deleted by typing Backspace (Ctrl–H). A carriage return ends a line.

The following commands are useful for configuring the 7318:

- **set** *Field Value*
- **show** [*Field*]
- **save**
- **admin** *Password*
- **default**

Refer to "BIOS Command Summary", on page D-1 for a complete listing of BIOS commands.

set Command

The **set** command configures the fields in the NVRAM. The BIOS keeps a local copy of the NVRAM, from which it runs. The **set** command changes this local copy. The user must run the **save** command to modify the actual NVRAM and to continue with the load.

Some of the **set** command fields define standard user functions, while others toggle debugging functions. For information on the troubleshooting fields, refer to "7318 Troubleshooting", on page 8-64.

The format for inputting parameters for the **set** command varies, depending on the command. The following information defines the allowed parameter formats. The **set** command makes use of the following special strings:

default	Specifies the default parameter.
auto	Specifies automatic selection (may also have a numeric equivalent).
disable	Turns off (may also have a numeric equivalent).
enable	Turns on (may also have a numeric equivalent).
null	Specifies the default parameter (same as default).
none	Specifies the parameter is not used.

If a parameter is a Boolean value, then the numeric values of 0 (False) or 1 (True) toggle the function. The `enable` or `disable` strings may also be used for Boolean parameters.

If a parameter is a decimal value, a positive integer from 0–32768 (depending on the bounds of that value) defines that function. Integers in the BIOS are not signed.

A string is a sequence of characters entered for a parameter: 0–9, a–z, A–Z, # (pound sign), _ (underscore), and / (slash). Strings start after the first space after the command field (for example, **set name** *StringStartsHere*) and may contain blanks. A carriage return ends the string. If a parameter is a string, the string `default` leaves the field undefined. Strings should not contain quotes.

The BIOS allows entry of either IPX or IP addresses. In either case, the string `null` leaves the address undefined. An IPX address is of the form:

```
nnnnnnnn:dddddddddddd
```

where *nnnnnnnn* is a hexadecimal representation of the internal network number, and *dddddddddddd* is a hexadecimal representation of the node number. An example of an IPX address might be:

```
0000b325:000000000001
```

An IP address is of the form:

```
xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* is a dotted decimal representation of the network number.

The following configuration subcommands require password privilege mode.

Note: In the following command syntax information, italics identify parameters whose actual names or values are to be supplied by the user.

Set Name (P10)

The **set name** command enables a system administrator to modify the NVRAM name field, a requirement for use with the SAP protocol. The default name is **CNS***EthernetAddress*, where the *EthernetAddress* parameter specifies the 7318's 12–digit hexadecimal Ethernet number (alpha digits A–F are in uppercase). The system administrator can specify an alphanumeric name for a 7318, which then is assigned automatically using the **save** command. To specify a unique name for a 7318, type:

```
set name Name
```

where the *Name* parameter is an alphanumeric string.

Set Load Image (P10 and S20)

The download image field for the 7318 is normally set to `default`. In this case, the host determines the download image. However in some circumstances, it is useful to preconfigure the load image. The **set loadi** command specifies a complete path name on the desired host. If the path name isn't found, the 7318 does not download. Set the download image field by typing:

```
set loadi Path
```

As mentioned, setting the *Path* parameter to null sets the download image field to null.

Note: AIX paths should be used in this command.

Set Config (P10 and S20)

The **set config** command specifies a configuration file for the 7318. The values for the **name** parameter are as follows:

default	The *XXXXXXXXXXXX*.cfg file is the default download configuration image for the 7318, where *XXXXXXXXXXXX* is the 7318's 12–digit hexadecimal Ethernet address (alpha digits a–f in lowercase). The host directory path for this file is the same as the download image path. If the 7318 does not find the *XXXXXXXXXXXX*.cfg file, it looks for the *XXXXXXXX*.cfg file, where *XXXXXXXX* is the lower 8 digits of the 7318's 12–digit hexadecimal Ethernet address. Failure to find either of these files results in a request for the **p10.cfg** or **s20.cfg** file. Failure to find all three files results in no error message.
none	Indicates that no configuration file is downloaded. The 7318 enables a single session on each port. This is only valid for the model P10.
<string>	Indicates the full pathname of the configuration file to load. BIOS tries to load only this file.

Set the config parameter by entering:

```
set config Name
```

Note: AIX file names should be used in this command.

Set Password (P10 and S20)

Initially, no password is defined. As long as the password is `null`, access to password privilege mode is unrestricted; that is, the **admin** command requires no password. Set the password to restrict access to password privilege mode. Because the **set password** command requires password privilege mode, you must run the **admin** command prior to setting the password. The maximum length of this password is 10 characters. Set the password by issuing the command:

```
set password String
```

Once you have set the password, you must supply it with the **admin** command to enter password privilege mode.

Note: If you forget or misplace this password, call your next level of support.

Set Frametype (P10 and S20)

Run the **set frametype** command by typing:

```
set frametype Type
```

Acceptable values for the *Type* parameter are the following:

auto	Specifies that the 7318 will automatically select the frametype.
ethernet_ii	Specifies a standard Ethernet version II frametype.

ethernet_802.3	Specifies IEEE 802.3 Ethernet frametype for Model P10.
ethernet_SNAP	Specifies IEEE 802.3 Ethernet frametype for Model S20.

Set Interface (P10 and S20)

The interface field, a decimal value, determines from which network interface (if any) to load. Valid values for the *Integer* parameter are:

0 or **auto**	Determines interface automatically.
1	Specifies a 10BaseT interface.
2	Specifies an AUI interface.
3	Reserved.
4	Disable loading.

This field defaults to 0 or auto. To set the network interface from which to load, type:

```
set interface Integer
```

Set Protocol (P10 and S20)

The protocol field, a decimal value, determines the network load protocol type. Valid values for the *Integer* parameter are:

0 or **auto**	Use SPX/IPX or UDP/IP for load.
1	Use SPX/IPX for load.
2	Use UDP/IP for load.

The default for this field is 0 or auto. To set the network load protocol, type:

```
set protocol Integer
```

Set Host1/Host2 (P10 and S20)

When downloading in an SPX/IPX or UDP/IP environment, you can force the BIOS to direct its network load to a particular host. This is done by issuing one of the following commands:

```
set host1 nnnnnnnn:dddddddddddd
set host2 nnnnnnnn:dddddddddddd
```

OR

```
set host1 xxx.xxx.xxx.xxx
set host2 xxx.xxx.xxx.xxx
```

Use the SPX/IPX format when downloading in a NetWare environment. Use the UDP/IP address format when downloading in a BOOTP environment. The default value for the `host1` and `host2` fields is `auto`. Setting this field to `auto` disables the feature. Refer to ”7318 BIOS Load Sequence”, on page 8-54 for more information on loading. Refer to ”set Command”, on page 8-57 for information on acceptable address formats.

Set Inet (S20)

The **set inet** command assigns an IP address to the 7318. The IP address needs to be assigned when the 7318 uses the UDP, BOOTP, and TFTP to download the boot image and configuration file. If you use an SPX/IPX environment, an Internet address is unnecessary. If the **bootptab** file specifies the IP address, it is unnecessary to set the address in NVRAM (using the **set inet** command). Assign an IP address by typing:

```
set inet xxx.xxx.xxx.xxx
```

Set Netmask (S20)

The **set netmask** command assigns a gateway netmask. Specify the netmask if the operating kernel is using TCP/IP and the `netmask` field is not set in the configuration or **bootptab** file. To set the netmask, type:

set netmask *xxx.xxx.xxx.xxx*

Set Gateway (S20)

The **set gateway** command assigns a default gateway address. Specify the gateway if the operating kernel is using TCP/IP and the `gateway` field is not set in the configuration or **bootptab** file. To set the gateway, type:

set gateway *xxx.xxx.xxx.xxx*

Set Nameserver (S20)

The **set nameserver** command assigns a name server address. Specify the name server if the operating kernel is using TCP/IP and the `nameserver` field is not set in the configuration or **bootptab** file. To set the name server, type:

set nameserver *xxx.xxx.xxx.xxx*

Set Domain (S20)

The **set domain** command specifies the Internet domain name of the 7318. The default value for the `domain` field is undefined. To set the domain, type:

set domain *String*

show Command

The **show** command displays the internal settings of the 7318 as they pertain to the BIOS. The **show** command invoked without arguments displays many of the field values, translating some of the field settings into more meaningful output.

In cases where the translated output differs significantly from the command syntax value, the syntax value (as well as its mnemonic) are shown in parentheses. To run the **show** command, type:

```
show
```

The output from the **show** command is as follows:

```
#> show
Internet Address (inet=0.0.0.0):            0.0.0.0
Netmask (netmask=0.0.0.0):                  0.0.0.0
Gateway (gateway=0.0.0.0):                  0.0.0.0
Nameserver (nameserver=0.0.0.0):            0.0.0.0
Domain (domain=null):                       null

Primary Download Host (host1=auto):         auto
Download Framing (frametype=auto):          ETHERNET_II
Download Image Name (loadimage=default):    /usr/lib/cns/cns-s20
Configuration File (config=default):        default
Name (name=CNS<ethernet>):                  CNS00406E0002DB

Ethernet Address:                           00406E0002DB
Debug Console (console=disable):            disable
Ram Available:                              2M
Self Test Result (self):                    passed
BIOS version:                               5.20
BIOS date:                                  12/01/94

#>
```

Useful items from this list are:

`Name`	Specifies a name assigned to the 7318.
`Ethernet Address`	Shown on a sticker on the back of 7318.
`Ram Available`	Specifies the 7318's configured memory size (M=megabyte).

Any field that can be set, using the **set** command, can be shown by typing the command:

```
show Field
```

For more information about the **show** command's troubleshooting capabilities, refer to "BIOS Console Troubleshooting Commands", on page 8-80. The following **show** subcommand, as well as the **save** and **default** commands, are useful as installation commands.

Show Version

The **show version** command displays the BIOS version in the form:

```
N.M
```

where the *N.M* parameter specifies the version.

admin Command

The **admin** command places the BIOS console in password privilege mode. The format for the command is:

```
admin Password
```

where the *Password* parameter is the administrative password. The system returns to normal mode when you type:

```
admin
```

without a password. If no password is defined, typing `admin` without a password places the BIOS console in password privilege mode.

default Command

The **default** command resets the contents of the NVRAM to initial settings. Depending on the mode of operation (normal or password privilege), some or all of the BIOS NVRAM can be reset.

Note: Remember to run the **save** command to write the reset values to the NVRAM.

reboot Command

The **reboot** command resets the 7318 hardware and restarts the BIOS. The **reboot** command is useful when users want to restart the 7318 from the BIOS console instead of power–cycling the unit.

Note: The **reboot** command acts like a warm start for the BIOS.

save Command

The **save** command rewrites the NVRAM and stores local copy values, changed with the **set** command, in a permanent manner. Run the **save** command before running the **load** command to save changes. Refer to "set Command", on page 8-57 for a description of configuration fields.

Note: Issuing the **load** command after having changed parameters without saving results in a warning.

Logging During Load

The log flag enables the basic input/output system (BIOS) to log load events during loader execution. This logging is useful when the 7318 is not downloading. In addition, system administrators can use the logging feature to diagnose LAN problems that may not be directly caused by the 7318. A request for the BIOS console automatically enables logging.

The BIOS has three major logging phases: discover, BOOTP, and TFTP.

Discover Logging – IPX Loads Only

The discover phase of the network load determines basic network information about available hosts and routers on the network and potentially determines the load host. The discover phase follows and logs a three–step process:

1. If you choose the Service Advertising Protocol (SAP) load method, the discover phase sends a find–any–server request to the network. The BIOS, upon receiving a response, has a potential load host.

```
#> load
Probing for an IPX load host
IPX load host named levesconte responded (00000011)
IPX load host named lester responded (00000001)
```

2. Once the BIOS receives a SAP response, the discover phase sends a series of one or more Routing Information Protocol (RIP) packets to the network. These packets request low–level information about how to reach the load host (received from the find–any–server request or from a directed load host). In this phase, the BIOS may delay for a while, after which it displays all hosts that responded to the RIP queries.

```
Attempting to resolve IPX addresses
Resolved IPX address: 00000011 at ethernet address 02608C2F15D9
Resolved IPX address: 00000001 at ethernet address 02608C2EB396
```

3. Finally, the discover phase advertises itself, using a SAP broadcast. The discover phase sends the following packet both when the BIOS comes up as well as every 60 seconds after that:

```
Netloader - DISCOVER 5.0
Using the AUI interface
Probing for an IPX load host
IPX external network is 00000002
IPX load host named muc2CNSLOADHOST responded (00000001)
Resolved IPX address: 00000001 at ethernet address 02608C2EB87D
Sending SAP advertisement (we are: CNS00406E0002DB_FFFFFFFF_BT)
```

Note: In the case of a User Datagram Protocol (UDP) load, the discover phase has no function. In a mixed load, the discover phase runs; however, UDP interaction only takes place in the BOOTP phase.

If there are problems in loading a 7318, the discover phase can be useful in problem determination. The determination of the Internet Packet Exchange (IPX) network number must happen; if it doesn't, the 7318 is not properly cabled to the LAN, or there are no IPX hosts sending packets on the network.

If the network is communicating and there are IPX hosts, there should be at least one response to the probe for a load host. While this response does not guarantee a download, it is necessary for the download process to progress. Also, after the 7318 has sent its SAP advertisement, you should be able to run the **cnsview explore** command from the host and see that particular 7318 in the list.

BOOTP Logging

The discover phase handles the preliminary steps of finding an IPX load host, unless one is specified by the **set host1** or **set host2** command. In the latter case, the discover phase resolves the preconfigured addresses to communicate with the host.

The BOOTP phase is used to determine if the host can download that 7318. For a UDP load, this is the first active phase, since the discover phase is not used for a UDP load. For the BOOTP phase to operate correctly, the following must be true:

- For IPX, a load host must have been found.
- For IPX, the **cnsd.conf** file must contain an entry for this 7318.
- For UDP, the **bootp** daemon configuration file must contain an entry for this 7318.
- For both cases, the entry in the **cnsd.conf** or **bootp** daemon configuration file must specify a valid file, or the 7318 NVRAM must contain a valid host as a result of the **set host1** or **set host2** command.

If the above criteria are met, the load host responds to the BOOTP request. If the BOOTP request file is null, the BOOTP response specifies the correct file to download. In the UDP download, the BOOTP packet may also have other IP information.

If the BOOTP phase does not get a response, but the discover phase has completed successfully, the host may not have been configured properly, or an incorrect file may have been specified. The BOOTP phase generates a trace similar to the following:

```
NetLoader - BOOTP 5.0 using IPX
Request file from: levesconte
Received BOOTP reply
Download file is /usr/lib/cns/cns-s20
```

TFTP Logging

If the BOOTP phase completes successfully and a load host replies, it is likely that the TFTP phase will complete successfully. If the TFTP request is not honored, there may be a problem with the host. If the file transfer does not complete, there may be a problem with transmission on the Ethernet. The TFTP phase sends a packet similar to the following:

```
NetLoader - TFTP 5.0 using IPX
Request download of /usr/lib/cns/cns-s20 from:
00000011:000000000001:83A2
Loading VERSION_3.44.1 in format: 5.12 uncompressed

Last block: 2418
Attempt to load config file: /usr/lib/cns/00406e0002db.cfg
Request download of /usr/lib/cns/00406e0002db.cfg from:
00000011:000000000001:83A2

Last block: 65
```

7318 Troubleshooting

The following troubleshooting information is divided into the following areas:

- System Checkout Using the Standalone BIOS Console, on page 8-64
- System Checkout Using the Service Aid, on page 8-66
- Problems with BIOS Console Communication, on page 8-67
- Front Panel Light Status (Startup and Loading), on page 8-68
- Problems with Terminals, on page 8-73
- Problems with Printers, on page 8-74
- Problems with Modems, on page 8-76.

The troubleshooting information is intended as a guide to help in problem isolation. Some of this information assumes more technical knowledge of the devices or applications involved.

One approach to isolation is substituting known good components or devices for suspect ones, including another 7318 if it is available. This approach is not included in these isolation procedures because the availability of hardware at the installation site is unknown.

In all cases that ask you to check the cabling or devices, you may replace cables, adapters, devices, null modems, and so forth to help isolate the problem.

To use this section most effectively:

1. Select the area that covers the problem.
2. Follow the recommended actions and procedures.

System Checkout Using the Standalone BIOS Console

The following procedure guides you through a full checkout of the 7318, providing a quick test to tell if the 7318 hardware needs to be replaced. The procedure uses the 7318 BIOS console commands to run diagnostic wrap tests on all of the ports on the 7318. If a failure is detected, no Service Request number is generated and you must replace the unit.

Note: Customer Engineers should run this procedure for 7318 FRU isolation. If the 7318 hardware passes this checkout, use the other troubleshooting information and procedures in this section to isolate the problem. The problem may originate from a wide spectrum of hardware and software variables including devices, cables, networks, and configuration parameters.

Prerequisites

1. An ASCII terminal must be available for the BIOS console.
2. All ports on the 7318 must be inactive (all users logged off and all applications stopped).
3. Establish a BIOS console. Refer to ”BIOS Console Operation”, on page 8-55 and ”7318 Diagnostics”, on page 8-83 for a description of the BIOS console and the **set**, **show**, and **diag** commands. To request a BIOS console, perform the following steps:
   a. Unplug the 7318 power cord.
   b. Connect an ASCII terminal to any one of the 7318 serial ports.
   c. Set the terminal to 9600 baud, 8 data bits, one stop bit, no parity.
   d. Plug in the 7318 power cord.
   e. Type # (pound sign, ASCII 0x23) continually until the console echoes #### (four pound signs).
   f. The BIOS should respond with a console header and the > (redirect symbol) prompt.

4. Enter password privilege mode, using the **admin** command. Refer to "admin Command", on page 8-61 for information on how to use the **admin** command.

Run the Ethernet Port Wrap

1. Remove all the Ethernet cable(s) from the 7318, noting the ports (A,T, T–X, I, O) they are connected to for later reattachment.

   Note: If the daisy–chain cable is not removed when doing LAN diagnostics, the diagnostic will fail along with any further LAN diagnostics. A reboot of the 7318 is needed to clear this failure.

2. Attach the AUI Ethernet wrap plug (Part Number 71F1167) supplied with your system to the AUI Ethernet port (labeled "A" on the back panel).

   a. Run the Ethernet port wrap test by entering the following commands, noting any error indications that appear:

   ```
   set device lan
   set interface 2
   diag –w
   ```

   b. Remove the AUI Ethernet wrap plug from the AUI Ethernet port.

3. Attach the RJ–45 Ethernet wrap plug (Part Number 00G2380) supplied with the 7318 to the T Ethernet port.

   a. Run the Ethernet port wrap test by entering:

   ```
   set device lan
   set interface 1
   diag –w
   ```

   b. Remove the RJ–45 Ethernet wrap plug from the T Ethernet port.

4. Attach the RJ–45 Ethernet wrap plug supplied with the 7318 to the T–X Ethernet port.

   a. Run the Ethernet port wrap test by entering:

   ```
   set device lan
   set interface 1
   diag –w
   ```

   b. Remove the RJ–45 Ethernet wrap plug from the T–X Ethernet port.

5. Attach the DB–9 daisy–chain wrap plug (Part Number 65G2382 male) to the O port. This wrap plug is only available with Feature Code 7909. If not used, skip to step 7.

   a. Run the Ethernet port wrap test by entering:

   ```
   set device lan
   set interface 0
   diag –u
   ```

   b. Remove the wrap plug from the O port.

6. Attach the DB–9 daisy–chain wrap plug (Part Number 65G2407 female) to the I port. This wrap plug is only available with Feature Code 7909. If not used, skip to step 7.

   a. Run the Ethernet port wrap test by entering:

   ```
   set device lan
   set interface 0
   diag –u
   ```

   b. Remove the wrap plug from the I port.

7. Reattach all Ethernet cables to the 7318.

8. If there were any error indications, then this procedure has failed. Stop problem isolation and replace the 7318.

Run the Parallel Port Wrap

Remove the parallel port cable or cables from the 7318 connectors (labeled P1 and P2) noting the ports they are connected to for later reattachment.

1. Attach the parallel port wrap cable (Part Number 65G2381) supplied with your 7318 to the parallel ports.
2. Run the parallel port wrap test by entering the following commands, noting any error indications that appear:

```
set device lpt1
diag –m

set device lpt2
diag –m
```

3. Remove the parallel port wrap cable and reattach the parallel port device cables.
4. If there were any error indications, then this procedure has failed. Stop problem isolation and replace the 7318.

Run the Serial Port Wrap

Run the serial port wrap for each serial port, COM00 through COM15 (other than the BIOS console port), on the 7318. Perform the following steps:

1. Remove the cable from the RJ–45 connector for the serial port to be tested.
   a. Install the serial port wrap plug (Part Number 65G2350) supplied with the 7318.
   b. Run the serial port wrap test by entering the following commands, noting any error indications that appear. Change the device identifier, `com00`, to match the port being tested.

```
set device com00
diag –w
```

   c. Remove the serial port wrap plug and reattach the serial cable that was connected to the port.
   d. If there were any error indications, then this procedure has failed. Stop problem isolation and replace the 7318.
2. Remove the BIOS console terminal and reattach the serial cable for the port as it was prior to running the procedure in ”System Checkout Using the Standalone BIOS Console”, on page 8-64.

If any of the diagnostics indicate a failure, replace the 7318.

If there were no error indications, then this procedure has passed. Exit this procedure. If you were referred to this procedure from another procedure, return to the original procedure and continue problem isolation. If you have performed no other diagnostic procedures, load the 7318 from a host on which the 7318 software is installed. Refer to ”Front Panel Status Lights (Startup and Loading)”, on page 8-68 if the load is unsuccessful.

System Checkout Using the Service Aid

The following procedure guides you through the operation of the 7318 Service Aid. It performs wrap tests on the 7318 parallel and serial ports, providing a reliable test of whether the hardware is operational. If the Service Aid detects a failure, it may be caused by many different parts of the system. In this case, perform the ”System Checkout Using the Standalone BIOS Console”, on page 8-64 for further problem isolation.

Note: The Service Aid can only be run from a system on which the software is installed and the devices and ports to be tested have been configured. For Model S20 units, the serial ports may not have P10–style ports configured on the system where the 7318 software support is installed. If there are no P10–style ports configured on the Model S20, the Service Aid will not be operational.

Prerequisites

You must have root user authority to run the Service Aid.

Note: Running some of the diagnostics automatically logs off any users on the 7318; thus, the diagnostics should be used carefully.

Procedure

1. To run the Service Aid, issue the **diag** command from the AIX command line.
2. From the first panel, select **Service Aids**.
3. From the next panel, select **IBM 7318 ComNet Server Service Aid**.
4. Select whether you want standard tests, which do not require special plugs, or advanced tests, requiring a serial port wrap plug (Part Number 65G2350) or a parallel port wrap cable (Part Number 65G2381). Press Enter.

The diagnostics run unattended and display their results in the **diag** command window. At the end of the diagnostics, either `RESULT: SUCCESS` or `RESULT: FAILURE` displays. If there is a failure, at least one additional message describing the problem displays. For example, you may also see a message such as `Device power-on self-test failed`.

Problems with BIOS Console Communication

If you have problems requesting a BIOS console, consider the following symptoms and their possible causes:

BIOS Console Problems	
Symptom	**Action**
Console doesn't echo pound sign.	Verify terminal is cabled correctly. Verify terminal settings (9600/8/1/N). Power–cycle the 7318 and type the pound sign a little later. Try a different serial port.
Pound sign garbled.	Verify terminal settings (9600/8/1/N).
No console header message.	If pound signs echo, try repowering the 7318 and typing the first pound sign faster, (four or more). Try a different serial port.
No Ready light within 20 seconds.	Power–on self–test failed. Replace the unit.

Terminal is Cabled Correctly

If your terminal has a Data Terminal Equipment (DTE) interface, the cable will require a null modem adapter. The cabling must support pins for Transmit Data (TD), Receive Data (RD), and signal ground (TD REF). Refer to ”Connector Pinouts”, on page B-1 for more information.

Terminal Settings

Verify that the terminal transmits and receives at 9600 baud, with 8 data bits, 1 stop bit, and no parity. Check the emulation mode of the terminal (if applicable) to be sure that the # (pound sign) key you are pressing corresponds to an ASCII pound sign symbol (0x23).

Power Cycle the 7318

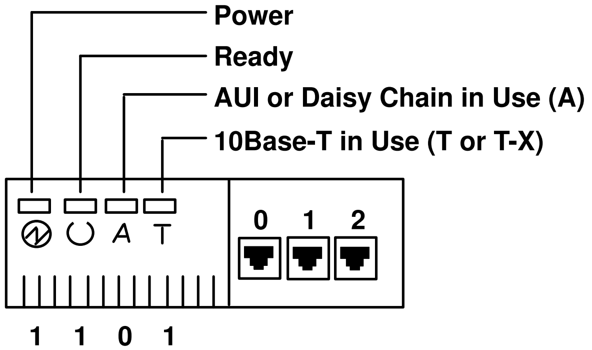
Unplug the 7318 and wait 30 seconds, then plug in the power cord again.

Front Panel Light Status (Startup and Loading)

Familiarize yourself with the following symbols, which are used to reference the light patterns:

- **0** indicates the light is off.
- **1** indicates the light is on.
- **B** indicates the light is blinking.

The Valid 10BaseT Light Pattern figure shows how the preceding symbols correspond to the lights on the front panel. In this situation, the **1 1 0 1** light pattern would signify that the power and the ready light are on, the AUI light is off, and the 10BaseT light is on.



Valid 10Base-T Light Pattern

Examine the front panel lights, and match them with the patterns in the following list:

0 0 0 0 No power to the 7318.

Action: Verify power to the 7318, cycle the power, wait 10 seconds.

1. If the light pattern changes, continue to match the new pattern with those shown in the table below. (This should have no effect on other devices attached to the Ethernet.)

   Attention: Daisy–chained 7318s will lose their connection if a higher 7318 in the daisy chain is power cycled (for example, power–cycling the second 7318 will end network connection for the third and fourth 7318 in the daisy chain).

2. If the light pattern does not change, replace the unit.

1 0 0 0 The power–on self–test appears to detect a fatal error.

Cycle the power, wait 15 seconds. If this pattern persists, replace the 7318.

1 B 0 0 Power–on self–test is complete, 7318 search for load host is in progress. (This light pattern is also seen when the BIOS console is active.)

Wait 30 seconds. If lights remain in this state, the 7318 has not found an operational Ethernet interface, or the transceiver attached to the Ethernet port does not support SQE heartbeat. Continue with the steps described below:

1. If the transceiver attached to port A does not support SQE heartbeat or if SQE heartbeat is turned off, move to light pattern **1 B 0 1** or **1 B 1 0**. Enable SQE heartbeat if the transceiver supports it.
2. If the 7318 is part of a 7318 daisy chain, then:
   a. Check that you cabled the 7318s correctly and power off all 7318s in the daisy chain. Power on the daisy–chained 7318s starting with the first. Refer to ”Attaching Multiple 7318s”, on page 8-15 for information on daisy–chain configuration.
   b. Remove the 7318 from the daisy–chain, and ensure it downloads correctly by itself on the Ethernet.

      If the independent download fails, then continue problem isolation with step 3.

      If the independent download succeeds, run the procedure described in ”System Checkout Using the Standalone BIOS Console” , on page 8-64 for each 7318 in the daisy–chain, looking for an in/out port (I and O port) failure.
   c. Call your next level of support.
3. Check for an Ethernet media problem. Refer to your Ethernet setup/installation guide to check for incorrect cabling, bad termination of the Ethernet, loose connectors, bad cable, down routers, and so forth.
4. Check for an Ethernet interface problem on the host. Refer to your Ethernet setup/installation documentation and look for incorrect definition of the Ethernet interface, host addresses and subnets, **BOOTP** configuration (Model S20), Model P10 SMIT panels, and so forth.
5. Check for a host or 7318 configuration problem. Refer to your 7318 installation and configuration documentation, and look for correct setting of network addresses, Model P10 SMIT panels, configuration file setup, and so forth.
6. Other things to try:
   a. On any host where the 7318 software is installed: the **cnsview –c ”explore”** command should display all 7318s that are powered up and reachable from the host.
   b. On any host where the 7318 software is installed: the **cnsview –c ”ipxping** *EthernetAddress*” command should display the following response:

```
EthernetAddress is alive
```

where the *EthernetAddress* parameter specifies the Ethernet hardware address on the back of the 7318.

c. Attach a BIOS console and issue the **load** command. The console will display log messages as the load sequence progresses, allowing you to check for the following:

   – Network addresses
   – Name resolution
   – Service advertisement and response
   – **Bootp** reply
   – Download image path and file name

d. Attach a BIOS console and issue the **load** command. The console will display log messages as the load sequence progresses, allowing you to check for the following:

   – Network addresses
   – Name resolution
   – Service advertisement and response
   – **Bootp** reply
   – Download image path and file name
   – Ethernet frametype
   – Configuration file path and file name.

   The degree of progress made in these steps may also indicate where the problem is. Refer to ”BIOS Console Operation”, on page 8-55 and ”Logging During Load”, on page 8-62 for more details.

e. Stop and restart the **cnsview** daemon and IPX protocol stack by issuing:

   To STOP:

```
cnsview –c ”daemon stop”
/usr/lpp/netware/bin/stopnps
```

   To START:

```
/usr/lpp/netware/bin/startnps
cnsview –c ”daemon start”
```

f. Run the procedure described in ”System Checkout Using Standalone BIOS Diagnostics Procedure”, on page 8-64.

7. Call your next level of support.

1 B 0 1
1 B 1 0

Load image and configuration file load in progress.

1. Check for an Ethernet interface problem on the host. Refer to your Ethernet setup/installation documentation and look for incorrect definition of the Ethernet interface, host addresses and subnets, BOOTP configuration (Model S20), Model P10 SMIT panels, and so forth.
2. Check for a host or 7318 configuration problem. Refer to your 7318 installation and configuration documentation, and look for correct setting of network addresses, Model P10 SMIT panels, configuration file setup, and so forth.
3. Other things to try:
   a. On any host where the 7318 software licensed program product is installed: the **cnsview –c "explore"** command should display all 7318s that are powered up and reachable from the host.
   b. On any host where the 7318 software licensed program product is installed: the **cnsview –c "ipxping** *EthernetAddress*" command should display the following response:

   ```
   EthernetAddress is alive
   ```

   where the `EthernetAddress` parameter specifies the Ethernet hardware address on the back of the 7318.

   c. Attach a BIOS console and issue the **load** command. The console will display log messages as the load sequence progresses, allowing you to check for the following:
      - Network addresses
      - Name resolution
      - Service advertisement and response
      - BOOTP reply
      - Download image path and file name
      - Ethernet frametype
      - Configuration file path and file name.

      The degree of progress made in these steps may also indicate where the problem is. Refer to "BIOS Console Operation", on page 8-55 and "Logging During Load", on page 8-62 for more details.
   d. Run the procedure described in "System Checkout Using the Standalone BIOS Console", on page 8-64.
4. Call your next level of support.

1 1 0 0

The 7318 has successfully downloaded. The **1 1 0 0** light pattern is valid for:

- Configuration file processing (five minutes maximum).
- Ethernet transceivers that do not support the SQE heartbeat.

If after five minutes the 7318 is still not operational:

1. Link integrity has been lost. Check the Ethernet connections.
2. Run the procedure described in "System Checkout Using the Standalone BIOS Console", on page 8-64.
3. Call the next level of support.

1 0 1 0 **1 0 0 1**	1. Upload in progress. Refer to ”BIOS Upload Sequence”, on page 8-54 for details on the BIOS upload. 2. Call the next level of support.
1 1 1 0 **1 1 0 1**	The 7318 is loaded and operational. Problems that persist with this light pattern are likely to be host configuration problems. 1. Check configuration file and Model P10 SMIT panels for incorrect configuration information. 2. Call next level of support.
1 1 1 1	Light test. If temporary, none. If solid, replace unit.
0 X X 1	Invalid Pattern
0 X 1 X	Invalid Pattern
0 1 X X	Invalid Pattern
Other	Invalid Pattern 1. If the 7318 fails the power–on light test but otherwise operates normally, replace the 7318 at your discretion. If the 7318 fails the power–on light test and operates abnormally, run the procedure described in ”System Checkout using the Standalone BIOS Console”, on page 8-64. 2. Call the next level of support.
Cycling	If the 7318 is cycling through patterns: 1. Load using the default configuration file. 2. Attach a BIOS console and issue the **load** command. The console will display log messages as the load sequence progresses, allowing you to check for the following: – Network addresses – Name resolution – Service advertisement and response – BOOTP reply – Download image path and file name – Ethernet frametype – Configuration file path and file name. The degree of progress made in these steps may also indicate where the problem is. Refer to ”BIOS Console Operation”, on page 8-55 and ”Logging During Load”, on page 8-62 for more details. 3. Call the next level of support.

Replacing a 7318

The following procedure preserves the TTY*x* and LP*x* devices configured for this 7318. If this is a Model S20 with P10–style ports, perform the sequence below on each host on which this 7318 is configured.

Prerequisites

You must have root user authority.

Procedure

1. Use the **smit ts7318_cs_mnu** fast path to access the **ComNetServer Server Configuration** menu.
2. Select **Show/Change Configured ComNetServers**.
3. Select the ComNetServer that was replaced.
4. Modify the ComNetServer Ethernet Address field to reflect the Ethernet address of the new ComNetServer.
5. Select **Do**.
6. Exit the SMIT interface.
7. Modify the bootptab file (**etc/bootptab**), if necessary, to reflect the new Ethernet address if the 7318 was replaced with a Model S20. This is done from the host where the 7318 boots.
8. Rename the customized configuration to reflect the new Ethernet address if there is a customized configuration file for this 7318. This is done from the host where the 7318 boots.
9. Update the NVRAM on the new 7318 to match the values on the old unit if the 7318 has customized NVRAM fields. Set these fields using the BIOS console.

Problems with Terminals

Terminal Problems	
Symptom	**Action**
The terminal does not respond.	Verify the cabling.Verify that the terminal is enabled.Verify the transmission parameters.Run the procedure in ”System Checkout Using the Service Aid” (Model P10 ports only).
You see garbled characters.	Verify the transmission parameters. Verify the slew rate. Verify flow control.
You see output to the terminal but the keyboard does not function.	Verify the transmission parameters. Verify the cabling. Run the procedure in ”System Checkout Using the Service Aid” (operational for Model P10–style ports only).
You see missing or bad cursor addressing.	Verify the terminal type. Verify flow control.
Multiple sessions does not refresh the screen properly on switch.	Verify the terminal type.

Cabling

1. If you are using a modem control device, type the following from any host on which the 7318 software is installed to check the status modem signals:

   ```
   cnsview –c ”show modem” DeviceName
   ```

   where the *Devicename* parameter specifies the name of the device, such as /**dev/tty1**. Data Carrier Detect (DCD) and Clear to Send (CTS) should both be 1.
2. If your cabling implements all of the EIA 232 signals supported by the 7318 and has EIA 232 standard DB–25 pinout, use the procedure for ”System Checkout Using the Standalone BIOS Console, Run the Serial Port Wrap”, on page 8-66 and the serial port

wrap plug supplied with the DPX/20 (part number 6298964) to verify the complete cable path to the serial device.

For Model P10 ports, the 7318 Service Aid Advanced Test may be used rather than the BIOS Console procedure.

Problems with Printers

Printer Problems	
Symptom	**Action**
Serial printer not working.	Verify printer status. Verify printer enabled and accepting requests. Verify serial cabling. Verify printer model. Verify slew rate.
Parallel printer not working.	Verify printer status. Verify printer enabled and accepting requests. Verify parallel cabling. Verify printer model.
Terminal transparent printer not working.	Verify printer status. Verify printer enabled and accepting requests. Verify serial connection. Verify printer model. Verify that you are only sending ASCII data to the printer.
Garbage characters on serial printer.	Verify serial connection. Verify printer model. Verify slew rate.
Garbage characters on parallel printer.	Verify printer model.
Garbage characters on terminal transparent printer.	Verify terminal to printer connection. Verify printer model.
Characters dropped on serial printer.	Verify flow control.
Characters dropped on terminal transparent printer.	Verify transparent printer rate.
Attached terminal of terminal transparent printer stops responding.	Verify transparent printer rate.

Note: The procedure described in ”System Checkout Using the Service Aid” applies to all of the symptoms in the Printer Problems table.

Printer Status

Check your printer to make sure it is online, does not show an error condition, or is not out of paper.

Printer Enabled and Accepting Requests

1. Use the **smit qchk** fast path to access the **Show the Status of Print Jobs** screen.
2. In AIX Version 3.2.5, select **yes** in the Show the status of ALL print jobs? field.
3. In AIX Version 4, enter an * (asterisk) in the PRINT QUEUE name field to display jobs in all queues.
4. Select **Do**.
5. Verify that the Status field for each printer indicates `READY` or `RUNNING`. Other values may indicate the printer is not working for various reasons (for example, out of paper, cable not attached, out of toner).
6. Exit the SMIT interface.

Serial Cabling

1. Use the **cnsview** command to check the status modem signals by entering the following:

```
cnsview -c ”show modem” DeviceName
```

where the *DeviceName* parameter is the name of the device, such as **/dev/tty01**. CTS and DCD should both be 1.

2. If your cabling implements all of the EIA 232 signals supported by the 7318 and has EIA 232 standard DB–25 pinout, use the procedure for "Run the Serial Port Wrap", on page 8-66 and the serial port wrap plug supplied with the DPX/20 (part number 6298964) to verify the complete cable path to the serial device.

   For Model P10 ports, the 7318 Service Aid Advanced Test may be used rather than the BIOS Console procedure.

3. Use an EIA 232 tester plugged into the back of the modem. The modem control line should be asserted, the Data Terminal Ready (DTR) line may flicker, and the TD and RD lines should flicker as data crosses the tester.

Printer Model

To verify the printer/plotter model:

1. Use the **smit chgprt** fast path.
2. From the displayed list, select the desired device.
3. From the **Change/Show Characteristics of a Printer/Plotter** menu, verify that the Printer/Plotter type matches the type of printer/plotter you have.

Parallel Cabling

Use the **cnsview** command to check the printer control signals by entering the following:

```
cnsview –c "show lptctl" DeviceName
```

where the *DeviceName* parameter specifies the name of the device, such as **/dev/lp2**.

When the printer is online but not printing, you should see:

```
selected:   1
peb:        0
error:      0
selection:  1
init:       0
feed:       0
```

If you do not see these signals, try replacing the printer cable.

Flow Control on a Serial Port

Use the **smit chgprt** fast path to take you to the **Change/Show Characteristics of a Printer/Plotter** menu. If your terminal (on AIX Version 3.2.5 systems) is set for XON/XOFF flow control, the XON–XOFF handshaking field should be set to **yes**. On AIX Version 4 systems, set the FLOW CONTROL to be used field to **xon**.

If your terminal (on AIX Version 3.2.5 systems) is set for DTR flow control, ensure that the Use DTR communication protocol field is set to **yes**. On AIX Version 4 systems, set the FLOW CONTROL to be used field to **dtr**.

Slew Rate

- To verify the slew rate on a Model P10 or on a P10–style port on a Model S20, perform the following:
  a. Use the smit **ts7318_port_mnu** fast path to access the **ComNetServer Port Options Configuration** menu.
  b. Select **Show/Change Slew Rates**.
  c. Select the ComNetServer containing the port group or groups to be displayed or changed.
- To verify the slew rate on a Model P10 using the **cnsview** command, type:

```
cnsview -c "show slew" /dev/ttyN
```

- To change the slew rate on a Model P10 using the **cnsview** command, type:

```
cnsview -c "set slew" Rate
```

- To verify the slew rate on a Model S20, view the **/usr/lib/cns/s20.cfg**. Find the [**Port***NN*] subsection that corresponds to the appropriate port. Find the `stty` entry and scan the line to find the slew setting.

Transparent Printer Rate

Use the **smit ts 7318_prt_mnu** fast path to verify the transparent printer parameters in the **Show/Change Configured ComNetServer Printers** dialog.

Problems with Modems

Modem Problems	
Symptom	**Action**
Modem does not answer.	Verify the cabling. Verify that modem autoanswer is enabled.
Modem will not dial out.	Verify the data rate. Verify the cabling. Verify the flow control. Verify the slew rate.
Modem answers with garbled data.	Verify the data rate. Verify the flow control. Verify the slew rate.

Note: The procedure described in "System Checkout Using the Service Aid" applies to all of the symptoms in the Modems table.

Cabling

1. Use the **cnsview** command to check the status modem signals by entering the following:

   `cnsview -c "show modem"` *`DeviceName`*

   where the *`DeviceName`* parameter specifies the name of the device, such as **/dev/tty01**. CTS should be 1. DCD should be 0 until the modem answers and should be 1 after it answers.

2. If your cabling implements all of the EIA 232 signals supported by the 7318 and has EIA 232 standard DB–25 pinout, use the procedure for "System Checkout Using the Standalone BIOS Console, Run the Serial Port Wrap", on page 8-66 and the serial port wrap plug supplied with the DPX/20 (part number 6298964) to verify the complete cable path to the serial device.

   For Model P10 ports, the 7318 Service Aid Advanced Test may be used rather than the BIOS Console procedure.

3. Use an EIA 232 tester plugged into the back of the modem. The modem control lines should be asserted, the DCD line should not be asserted when there is no carrier, and the TD and RD lights should flicker as data crosses the tester.

Modem Autoanswer is Enabled

Verify that the S0 register is not set to zero (disable autoanswer).

Data Rate

If you have configured your modem to use a fixed data rate (recommended), verify that the SMIT setting matches the modem data rate (usually 9600).

Flow Control

To verify flow control settings, perform the following steps:

1. Run the **smit chgtty** fast path.
2. Select a TTY from the list that displays.
3. On AIX Version 3.2.5 systems only, select **Change/Show TTY Program**. If you are an AIX Version 4 user, proceed to step 4.
4. Scan to the end of the field attributes for the STTY attributes for LOGIN field. You should see **ixon** and **ixoff** indicating that the TTY is configured for XON–XOFF handshaking.
5. On AIX Version 4 systems only, set the FLOW CONTROL to be used field to **xon** for XON–XOFF flow control or to **rts** for RTS/CTS flow control
6. On AIX Version 3.2.5 systems only, select **Change/Show Hardware Characteristics of a TTY.**
   a. If you set your modem for XON/XOFF flow control, the SMIT dialog should display `yes` for XON–XOFF handshaking.
   b. If you set your modem for RTS/CTS flow control, you should confirm that the port is configured for RTS/CTS flow control.
7. Select **Do**.
8. Exit the SMIT interface.

To verify the flow control on a Model S20 port that is not configured as a P10–style port, view the default or customized configuration file for the 7318. Find the [**Port**NN] subsection that corresponds to the appropriate port. Find the `stty` entry and scan the line to find the flow control settings `(ixon`, `ixoff`).

Slew Rate

To verify the slew rate on a Model P10, see "Slew Rate", on page 8-75.

Problems with P10–Mode Remote Ethernet

This section discusses how to troubleshoot typical problems you may have over WAN connections or other routed networks. This section assumes that you have a basic understanding of the BIOS console and other administrative functions in **cnsview**.

Booting Across a Router Using Undirected IPX Loading

If you have problems getting the 7318 Model P10 to boot over a router, it is likely this is a configuration problem. The best approach for this is to make certain your configuration works locally first. This way you can be sure that your hardware and cabling work. If you can configure the 7318 to boot locally, then just a few parameters must be changed to have the 7318 boot remotely.

1. Use the **smit ts7318_cs_mnu** fast path to access the **ComNetServer Server Configuration** menu.
2. Select **Show/Change Configured ComNetServers**.
3. Select the desired ComNetServer from the displayed list.
4. Change the ComNetServer Network Address field. This address must match the IPX address of the remote Ethernet segment the 7318 is directly connected to. This number must match the address the router has configured for this Ethernet segment.
5. Select **Do**.
6. Select **Exit** to exit the SMIT interface.
7. Attach the 7318 to the remote segment and attach the BIOS console to a 7318 serial port.

8. Issue the **load** command on the BIOS console and watch the load process. The 7318 will select the Ethernet interface AUI or 10baseT and then probe for a load host. If no load hosts respond, the 7318 will try different frametypes and probe for a load host. If no load hosts respond, the 7318 will try to load using UDP/BOOTP. Since the host is not configured to load using UDP, the 7318 will not succeed and try again with IPX. The sequence is repeated until a load host responds.

   If no load host is found, the problem might be that the router is not recognizing the SAP broadcast packet that the 7318 sends to probe for a load host, or the router did not pick up the SAP broadcast the DPX/20 is sending out claiming to be a load host for the 7318. The router should have a table of SAP entries. The system should be listed with the 043d'h service. This service, 043d'h, is registered with Novell for exclusive use with the 7318. If the SAP entries are not found in the router's tables, verify that the router is properly configured with SAP and RIP enabled for IPX.

9. If the problem persists after SAP and RIP are enabled, verify that the router and system are configured for the same IPX frametype. The frametype is set on the system in the **/etc/netware/NPSConfig** file. At the end of this file is a list of adapter interfaces. Select the proper frametype for your network and ensure the router is configured the same. A useful tool for determining if the host and the router are communicating is the **ipxping** command. This command is similar to the IP **ping** command; however, not all routers support it. If your router supports **ipx echo**, you can use it to verify communications at different nodes along the network path. If your router does not support **ipx echo**, you can trace an IPX packet across the network using a sniffer or other packet–tracing tool.

10. If the frametypes are properly configured and SAP and RIP are enabled for IPX, the router should have the DPX/20 listed with 043D'x service in its SAP tables. If this problem still persists or if **ipxping** fails, get a sniffer trace of the network to determine the cause of the problem.

11. If the host is found in the router tables as a 7318 load host (043D'x), and the 7318 still cannot find a load host, there is something preventing the 7318 and the router from communicating. From the BIOS console, enter:

    ```
    admin
    ```

    Next, enter:

    ```
    ipxping xxxxxxxx:xxxxxxxxxxxx
    ```

    where *xxxxxxxx*:*xxxxxxxxxxxx* is the IPX address of the router.

12. If the **ping** fails, try setting the frametype to another value using the **set** command from the BIOS console.

    ```
    set frametype ethernet_802.3
    ```

13. If **ipxping** fails on all three supported frametypes, and the router configuration has been verified, get a sniffer trace of the network traffic to determine the problem.

    If the 7318 can **ipxping** successfully and the host shows up in the router tables properly as described above, the 7318 will show that an IPX load host responded on the BIOS console during a load sequence. Now that a load host responds, the 7318 will determine its network address. The 7318 determines its network address using the following algorithm:

    xxxxxxxx:*yyyyyyyyyyyy*

    Where *xxxxxxxx* is the network address of the Ethernet segment that the 7318 is directly connected to. The 7318 determines this from SAP broadcasts that other IPX nodes send out on the Ethernet. The variable *yyyyyyyyyyyy* is the Ethernet medium access control (MAC) address of this 7318. The MAC address is listed on the back of the 7318 above the AUI connector.

    The combination of network address and MAC address gives the 7318 a unique network address and eliminated the need for a user to preconfigure the 7318 ROM with an IPX address. If the 7318 cannot resolve IPX addresses, it will not be able to function. However, the fact that the 7318 found a load host indicates it is communicating properly with the router and should not have a problem resolving the IPX address.

The next step the 7318 takes is to send an RIP request to the router to determine the address of a load host. With 5.21 BIOS level, the 7318 responds to both directed and broadcast RIP responses from the router.

Now that the 7318 has the load host address and its own address, it broadcasts a SAP advertisement packet. The host receives this packet and stores the information its SAP table. Once it is received, the **cnsview explore** command shows the 7318 in a status of booting. From this point on, the 7318 sends SAP packets once per minute with its status. A few minutes after booting, the **explore** command indicates the 7318 is online.

The 7318 cannot respond to **ipxping** packets sent from the host until it has determined its IPX address. You should attempt to **ipxping** from the 7318 to the host. The 7318 can **ipxping** without knowing its complete address.

14. If the problems persist and you cannot get a sniffer trace, attempt to boot using a directed UDP load. This is done from the BIOS console on the 7318 using the following commands:

```
set loadi /usr/lib/cns/cns-p10
set protocol 2
set inet xx.xx.xx.xx  (where xx.xx.xx.xx is the 7318 IP address)
set gateway xx.xx.xx.xx
set netmask xx.xx.xx.xx
set frametype 802.3
set host1 xx.xx.xx.xx  (where xx.xx.xx.xx is the host IP address)
```

15. Ensure TFTP is enabled on the host system.

16. The frametype selected by the 7318 at boot time can be overridden for IP or IPX in the configuration file. Search the configuration file for ”frametype.” There is a separate setting in the TCP and the IPX configuration file sections. There are helpful hints in the configuration file comments of each section.

Performance Considerations for P10 functions Over a WAN Link

1. The WAN link should be at least 32 Kbps for one 7318.
2. The line discipline processing should be set to host.
3. Force the line discipline to the host on EACH tty or LP device using the **cnsview** command:

   ```
   cnsview –c ”set ldisc hostld 1” /dev/ttyxx
   ```

   This command must be reissued after a host shutdown. It does not need to be reissued on a 7318 reboot. To accomplish this on the host, a script needs to be added to the **inittab** file.

   Turn the vital product data (VPD) and status functions off in the **/usr/lib/cns/cnsd.conf** file. To turn these functions off, edit the **cnsd.conf** file as follows:

   ```
   #log   config  svclts  2sess  periodic method vpd  boot  stats
    log   config method novpd nostats
   ```

   After editing the file, issue the following command:

   ```
   cnsview –c ”daemon init”
   ```

BIOS Console Troubleshooting Commands

The following commands may be useful for diagnosing problems with downloads and uploads:

- **set** *Item Value*
- **show** *Item*
- **biosload**
- **load**
- **stats**
- **ipxping** *Parameter*

Note: In the following command syntax information, italics identify parameters whose actual names or values are to be supplied by the user.

set Command

The following troubleshooting subcommands require password privilege mode. Use the **save** command to update NVRAM. To return to normal operation, reset the parameters.

The **set** command has several troubleshooting subcommands that toggle debugging, display diagnostic information to the terminal, rerun the load procedure, and loop diagnostic tests. Refer to "BIOS Console Configuration Commands", on page 8-56 for more information on the **set** command.

set Log

The **set log** command determines whether the BIOS uses the BIOS console for communication. Accessing the BIOS console automatically enables the **set log** command.

Note: The **set log** command is a special support function to be used with technical support only.

set Console

The **set console** command sets the 7318 debug console port. When a terminal is in use as a console, the host cannot access it.

Note: The **set console** command is a special support function to be used with technical support only.

set Upload

The **set upload** command uploads a memory image dump of the 7318 to the load host. This memory image can be analyzed to find programming problems.

Note: The **set upload** command is a special support function to be used with technical support only.

show Command

The following **show** subcommands as well as the **load** and **stats** commands are useful for troubleshooting problems with the 7318.

show self

The **show self** command displays on demand the result of the BIOS diagnostic self–test, a 32–bit hexadecimal number. The value of the self–test mask for a 7318 that has successfully passed all of the power up diagnostics is 0xFFFFFFFF. To see the result of the BIOS diagnostic self–test, type:

```
show self
FFFFFFFF
```

The meaning of each bit is as follows:

```
/* selftest results 1=passed, 0=failed */

port 0                           0x00000001
port 1                           0x00000002
port 2                           0x00000004
port 3                           0x00000008
port 4                           0x00000010
port 5                           0x00000020
port 6                           0x00000040
port 7                           0x00000080
port 8                           0x00000100
port 9                           0x00000200
port 10                          0x00000400
port 11                          0x00000800
port 12                          0x00001000
port 13                          0x00002000
port 14                          0x00004000
port 15                          0x00008000

CPU test                         0x00010000
DRAM test                        0x00020000
PROM test                        0x00040000
LAN loopback                     0x00080000
DMA ASIC test                    0x00100000
Static RAM test                  0x00200000
Reserved                         0x00400000
First NVRAM                      0x00800000
Second NVRAM                     0x01000000
CIO #1 Test                      0x02000000
CIO #2 Test                      0x04000000
VPD area                         0x08000000
```

show pmem

The BIOS automatically detects the amount of physical memory in the 7318. It does not, however, change that setting in the NVRAM, unless a conflict exists that makes the platform nonfunctional.

The **show pmem** command displays the results of the BIOS autodetect. It shows the number (1–8) that corresponds to the proper dynamic random access memory (DRAM) setting for that amount of memory. If the **show ram** command displays a different DRAM value than the **show pmem** command, then the 7318 contains a different amount of memory than configured in the NVRAM. To see a 7318's physical memory, type:

```
show pmem
4
```

biosload Command

The **biosload** command reprograms the BIOS NVRAM. Use this program with care, since improper programming of the NVRAM may result in the 7318 needing to be reloaded using the low–level loader.

Note: The **biosload** command is a special support function to be used with technical support only.

load Command

The **load** command loads the 7318 with its download image. After issuing the **load** command, the BIOS downloads an image and transfers control to the executable entry point. To run the **load** command, type:

load *PathName*

where the *PathName* parameter is an optional download path name. If any NVRAM parameters have been changed prior to the load (from the BIOS console), you must use the **save** command before issuing the **load** command. If you do not, the BIOS issues a warning and returns you to the BIOS console prompt.

ipxping Command

The **ipxping** command tests the 7318's hardware and network attachment. To send repeated echo requests to another machine, type:

```
ipxping IPXAddress
```

where the *IPXAddress* parameter specifies the IPX address of the machine to which to issue the **ipxping** echo request.

stats Command

The **stats** command displays information about the state of the LAN interface. The command is a diagnostic tool for debugging Ethernet problems.

7318 Diagnostics

There are three levels of 7318 diagnostics:

- Power–Up Diagnostics
- Basic Input Output System (BIOS) Based Diagnostics
- Service Aids

The BIOS–based diagnostics execute from the BIOS console. They allow field testing of the 7318 hardware.

Power–Up Diagnostics

The BIOS performs a series of diagnostic tests before the 7318 starts its image load. The diagnostic tests are as follows:

1. Lamp test, basic operation, nonvolatile random access memory (NVRAM) checksum
2. NVRAM test
3. Dynamic RAM (DRAM) test
4. Direct memory access (DMA) application–specific integrated circuit (ASIC) register read/write test
5. DMA ASIC static RAM test
6. Counter timer/interrupt test
7. 8530 chip test
8. 82593 local area network (LAN) chip tests

The BIOS stores the last test result in the self–test field. This can be read either with the BIOS console or the **cnsview** command. If the self–test passes, the hexadecimal value 0xFFFFFFFF is stored in the self–test field. If any of these tests fail, the Ready light will not come on within 20 seconds. Replace the unit.

BIOS–Based Diagnostics

You can initiate diagnostics from the 7318's BIOS. See "BIOS Console Operation", on page 8-55 for a description of how to operate the BIOS console. This section describes the BIOS commands you can use for diagnostics. The basic commands for diagnostics are:

- **set** *Item Value*
- **diag** *Options*
- **show** *Item*

Note: In the following command syntax information, italics identify parameters whose actual names or values are to be supplied by the user.

set Command

The **set** command diagnostic subcommands act similar to NVRAM subcommands; however, they are *not* kept in the NVRAM and are not saved after power–off. Refer to "BIOS Console Configuration Commands", on page 8-56 for information on the **set** and **show** diagnostic subcommands.

The **set** command sets the parameters for the **diag** command. This command takes two different kinds of parameter values. If a parameter is a decimal value, then the valid numeric values are 0–*N*, depending on the bounds of that value. If a parameter is a string or a network address, a null string value leaves the item undefined.

Strings should not contain quotes or whitespace. All text after the *Item* parameter is interpreted as a string.

The following sections describe sets that globally modify execution of the diagnostics. These sets apply to any diagnostic.

set device/pair

Using **set device** or **set pair** commands, you can specify which device or devices to test when you run the **diag** command. The commands are entered as follows:

```
set device String1
```

OR

```
set pair String2
```

where the *String1* parameter may have the following values:

cns	Specifies a 7318 (power up diagnostics).
lan	Specifies the LAN controller.
lpt1/lpt2	Specifies line printer port 1 or 2.
com00–com15	Specifies serial ports.

where the *String2* parameter may have the following values:

com00–com15	Specifies serial ports different from *String1*.

The **set device** command attempts to choose a likely ”pair” for any test. The following pairs are selected automatically:

Device	Pair
lpt1	lpt2
lpt2	lpt1

set iterations

The **set iterations** command specifies the number of times a diagnostic executes before control returns to the monitor. The allowed range of iterations is 0 to 10000. A value of 0 means that a diagnostic runs until interrupted by pressing Enter. To set the number of iterations, type:

```
set iterations Value
```

set size

The **set size** command determines the size of the loop–pattern blocks that are written to a device (either LAN, COM, or LPT). The allowed range of the *PSize* parameter is 2–990. The high range of the pattern size is limited by how the BIOS handles Ethernet receive buffers. Set the pattern size by typing:

```
set size PSize
```

set pattern

The **set pattern** command determines the kind of data in the pattern blocks that are written to a device (either LAN, COM, or LPT). Set the pattern parameter by typing:

```
set pattern String
```

where the *String* parameter may have the following values:

00	Specifies all zeros.
11	Specifies all ones.
a5	Specifies a 0xa55a pattern.
0f	Specifies a 0x0ff0 pattern.

w1	Specifies a walking ones pattern.
rd	Specifies a random pattern.
ln	Specifies a printable line (ending with carriage return/line feed).
tc	Specifies the fixed `abcd-nop` pattern.

set rate

The **set rate** command determines the output or input rate of the COM test. The **set rate** command expects a string that corresponds to the different band rates. To set the COM test I/O rate, type:

```
set rate String
```

where the *String* parameter may have the following values: 50, 75, 110, 300, 600, 1200, 1800, 2400, 4800, 7200, 9600, 19200, 38400, 57600, 76800, and 115200.

diag Command

The **diag** command runs a diagnostic. The device and possibly other parameters must be configured before running the diagnostic. The **diag** command runs with flags that change the level of the test.

There are several global variables that control the operation of the **diag** command. These variables should be set before running a test, using the **set** command.

The **diag** command diagnostics run on one of the following devices:

- CNS – 7318
- COM – serial ports
- LPT – line printer ports
- LAN – LAN interface

The **diag** command does several levels of test, depending on flags. For some tests, you need to use a loopback device for the tests to run. The types of loopback devices are:

- Single–channel loopback plug for testing a serial port
- Pair loopback cable for testing two serial ports
- Output drain plug for testing the LPT port
- Printer pair loopback cable for testing the two LPT ports
- 10BaseT wrap plug for testing a 10BaseT port
- AUI wrap plug for testing the 10BaseT port
- Daisy–chain wrap plug for testing a daisy–chain port

The following table describes the diagnostic tests:

diag Command Tests

Set Device	**diag Command Flags**	**Test Device**	**Comments**
cns	2	N/A	Runs DRAM test.
	4		Runs ASIC test.
	5		Runs Static RAM test.
	6		Runs CLO test.
	8		Runs LAN test.
	9		Runs SLO test.
com*XX*	None	Single	Internal wrap (8530).
	–w	Single	External data wrap plug.
	–m	Pair	External data wrap cable.
lpt*X*	None	Output drain	Must be connected to a printer in ready mode.
	–m	Pair	External data wrap cable.
lan	None	N/A	Internal wrap (82593).
	–w	10BaseT	External wrap on T port carrier detect and collision test.
	–w	AUI	External wrap on AUI.
	–u	Uplink	Daisy–chain wrap.
	–u	Downlink	Daisy–chain wrap.

The **diag** command runs various diagnostics. It uses a variety of wrap tests to test larger and larger data loops within the 7318. The **diag** command is directed at a test device, which is determined by using the **set device** command. Depending on the specified mode, you may have to set another device as the pair device. If you select the **–m** flag without setting the pair device, the results are unpredictable. If a pair device is not used for a test, the **diag** command ignores the **set pair** value.

A wrap test sends data out one device and reads it back in, either through the same device (for example, in the case of the LAN) or the pair device. If the device is physically separate, a loop cable must be connected to the 7318 port before running the test.

Shorter loops can determine the base functionality of a 7318. A basic wrap to a COM port checks much of the circuitry out to the 8530 chip. However, since there are components that link the 8530 chip with the RJ–45 port, a loop using either a wrap cable or a wrap plug has to be used to test the whole data path.

The **diag** command does a simple set of steps:

1. Initializes the device or devices
2. Fills an output buffer with the requested pattern and size
3. Sets up and starts the input device
4. Writes the output buffer
5. Reads the input buffer
6. Compares the input buffer to the output buffer
7. Tests control signals in some cases.

The tests run in a sequential manner. While this does not allow simultaneous testing of multiple devices, it does allow the BIOS to know exactly what the sequence of events should be and test for problems along the way.

The **diag** command and the BIOS detect and log problems. The **diag** command checks for the following:

1. Successful initialization of the device
2. Completion of the write operation
3. Completion of the read operation
4. Correct pattern comparison
5. Verification of control sequences.

If any of the five steps fail, the **diag** command prints a message, logs an error, aborts, and then restarts the test if the user requested more than one iteration.

Along with the preceding five steps, the BIOS has error–logging logic, which detects low–level problems. This logic resides in subroutines such as the interrupt handlers. If the BIOS detects an error, it logs the problem (as with the power–up diagnostics) in addition to the **diag** command logging.

Improper setup of the **diag** command can cause steps 2–5 to fail. However error logging is unlikely, since hardware problems usually cause this type of low–level problem.

The user can interrupt the **diag** command by pressing Enter.

Using the diag Command

Before going into detail on the operation of the **diag** command, it is useful to plan how you will diagnose the 7318.

In most cases, the **diag** command is used when there is some kind of hardware problem suspected in a 7318. If this is the case, then testing is usually confined to that area. The following sections discuss each major area.

Diagnosing the 7318

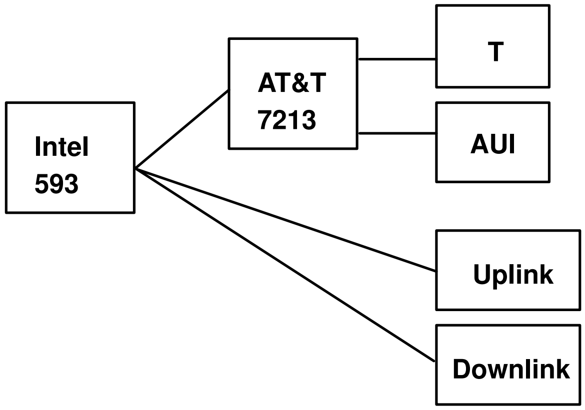
Running the **diag** command on the CNS device runs the power–up self–tests. These tests do a cursory test of all of the basic hardware systems and verify whether the 7318 can be downloaded.

None of these tests require setup. If there is an indication that a 7318 is failing one of the power up tests, the CNS device diagnostic can be used to verify this fact.

The CNS device is also useful in running the memory test. It may be useful to run this repeatedly if there is a suspected failure in the memory system. The post power–up diagnostic (PPD) memory test does a RAM refresh test that the power–up test does not perform.

Diagnosing the LAN Device

The LAN is diagnosed using a variety of loopbacks. The Loopback Diagnostics figure displays the chips that the **diag** command tests:



Loopback Diagnostics

The **diag** command verifies the Intel 593 chip, which controls the LAN. If this command runs correctly, the loopbacks proceed through the Intel 593 out to the other ports of the 7318. You should run the appropriate loop test for the kind of failure you are having. Thus, if the 10BaseT port doesn't work, run the **diag –w** command, using the 10BaseT wrap. If the AUI doesn't work, run the **diag –w** command, using the AUI wrap. If the 7318 is attached to the LAN, using the uplink/downlink cables, run the **diag –u** command.

Notes:

1. The 10BaseT loop test is not very strong. If the diagnostic passes this test and there are still problems, you may want to try the **ipxping** command.
2. Both the 10BaseT and AUI ports go through common logic (the AT&T 7213). If this part is defective, neither port may work.
3. The interface must be set before using the **diag –w** command.

Diagnosing the LPT Device

If a problem is suspected with the printer ports, there are two quick methods of resolution. If a printer is already attached to one of the ports, run the **diag** command on that port, using the default sliding pattern. In this case, set the size parameter to 80 so that the pattern fits on the page. This indicates whether the port is working.

For a more thorough test, attach a printer loopback cable to printer ports 1 and 2 and run the **diag –m** test. This test outputs data to the device side and receives it on the pair side, verifying both the data and the toggling of the control signals on the printer interface.

If the test fails, note the kind of error you are getting. If there is a write timeout, the device side is probably at fault. If there is a read timeout, the pair side is at fault. Switch the sense of the test to the other port, and retry the test.

Diagnosing the COM Device

COM tests should be performed when a terminal does not work. There is a variety of tests that can be run, each more extensive than the previous test.

To verify the Intel 8530 serial controller, run the **diag** command without external loop plugs. Experiment with several speeds. If the command runs and a terminal is attached to the port, rerun the command at the terminal speed with the line pattern to check whether data displays on the terminal. If no data displays, the drivers may be defective or the terminal may be incorrectly cabled.

Use a single port wrap plug to check the drivers and receivers. Insert this plug in the port to be tested, and run the **diag –w** command.

Use loop cables, if you have problems with modem signals. Connect a loop cable from the suspected port to another port, and run the **diag –m** command. Pairs of ports (such as 0 and 1, or 2 and 3) use the same controller. This test is most effective if ports on different controllers are chosen.

show Command

The **show** command can display diagnostic values configurable by the **set** command. To display a diagnostic value, type:

```
show Item
```

where the *Item* parameter can be:

- pattern
- device
- pair
- iterations
- rate

The **show** command displays values based on the last respective **set** command.

Chapter 9. Network Terminal Accelerator Overview

The Network Terminal Accelerator is a unique Ethernet interface adapter that provides host connectivity, increased terminal throughput, and faster response time. There are two models of the Network Terminal Accelerator. The Network Terminal Accelerator–256 supports up to 256 concurrent sessions. The Network Terminal Accelerator–2048 supports up to 2048 concurrent sessions. Both models provide the same functions, with the exception of the number of sessions supported.

Notes:

1. AIX Versions 4.3 and later do not support the Network Terminal Accelerator.
2. This section applies to systems running AIX Versions 4.1.5 to 4.2.1 only.
3. For information on using the Network Terminal Accelerator on AIX Version 3.2.5, see *7318 Network Terminal Accelerator Guide and Reference*.

The Network Terminal Accelerator is a high–performance, single–port Ethernet adapter with associated software and firmware. The adapter reduces the demand on a host computer's processing resources for TCP/IP, Telnet, and rlogin communication protocols. With this adapter, the host CPU is freed to do more processing on user applications. The adapter accomplishes this by performing most of the communication protocols on the adapter using a powerful processor. TCP/IP, Telnet, rlogin, and Ethernet MAC–layer protocols have all been offloaded to the adapter from the host CPU, allowing the host CPU to use the adapter as a simple, efficient serial multiplexer.

The adapter also provides the host workstation with a standard Ethernet interface for applications that do not use rlogin and Telnet protocols. With this capability, the adapter provides an interface for all Ethernet communications, while increasing the number of Telnet and rlogin protocol users the host can support.

There are two adapter feature codes available. Feature code 2402 supports up to 256 concurrent Telnet and rlogin user sessions. This option has 2MB of local memory on the adapter. Feature code 2403 supports up to 2048 concurrent Telnet and rlogin user sessions and has 8MB of local memory on the adapter.

This adapter does not take the place of a network's existing TCP/IP or Ethernet adapter. Instead, the adapter coexists with the existing network space.

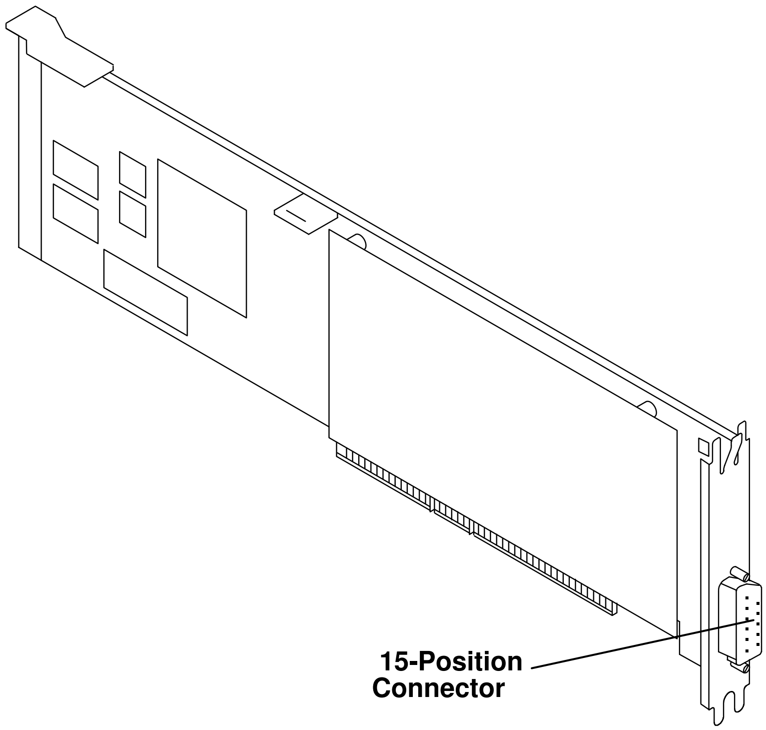
This section discusses the following:

- Hardware Features, on page 9-2
- Software Features, on page 9-4

Hardware Features

The Network Terminal Accelerator is a high–performance, Type 3 Micro Channel adapter. It handles all asynchronous terminal multiplexing for Ethernet transport media and TCP/IP emulation protocols.

The following figure represents the adapter card.



CPU

The adapter uses an Intel SuperScalar i960CA RISC–embedded processor that decodes and executes multiple instructions in parallel from a single instruction stream. At 25 MHz, using a conventional memory system and a burst bus only 32 bits wide, the adapter sustains an execution rate approaching 50 million instructions per second.

Coprocessor Support

The adapter includes a powerful combination of coprocessors:

- A 32–bit Intel 82596 LAN coprocessor to automatically handle Ethernet protocols
- An integrated interrupt controller
- A 1KB, two–way, set–associative instruction cache, eliminating bottlenecks by decoupling the instruction stream from the external bus
- A local register cache, dramatically reducing the number of memory accesses.

Local Memory

The memory architecture on the adapter features two–way bank interleaving, allowing optimized memory access. Memory is accessed in long–word (32–bit) increments. A two–bank interleaf scheme allows the access time of one bank to overlap the data time of the other, permitting the CPU to perform burst read cycles with one initial wait state, followed by zero wait states (1–0–0–0).

Two memory configurations are available on the adapter: the Network Terminal Accelerator–256 comes with 2MB, and the Network Terminal Accelerator–2048 with 8MB.

Micro Channel Interface

The adapter uses the Intel 82325 chip set designed by Bull. The 82325 supports:

- Address and data parity checking
- Burst mode data transfers
- Subsystem control block (SCB) registers
- Programmable DMA channel
- Internal error checking.

Software Features

The Network Terminal Accelerator adapter provides unique integration of standard networking protocols with a host interface similar to a multiplexer. While networked communications servers such as the 7318 Model P10 and S20 view the adapter as a host that offers standard Telnet or rlogin terminal services, the host computer views the adapter as a simple, efficient, serial multiplexer and a standard Ethernet interface. The adapter supports both 802.3 Ethernet thin (BNC) and thick (DIX) wiring.

The adapter's serial multiplexer capability makes the intervening network and its protocols invisible to the host. The host appears to communicate directly with the serial ports of the networked communication server devices. The multiplexer capability completely removes time–consuming network interface tasks from the host, including all network packet interrupt and TCP/IP protocol processing and, more importantly, the **telnetd** daemon, which runs as a user–level process. In addition, this capability eliminates the high overhead of processing context switches normally associated with network–connected terminal I/O.

While the adapter insulates the host from networked terminal I/O processing, it provides most of the communication capability of a standard Ethernet interface. This capability is provided by two unique IP addresses for the adapter's single Ethernet interface. The serial multiplexing software is addressed with its own IP address. The standard Ethernet software is addressed with a different IP address. Users must use the IP address appropriate for their application.

POSIX Functions

The serial multiplexing software provides additional tty processing beyond the protocol processing already mentioned. It handles special character recognition, line buffering and editing, echoing, character translation, and tab expansion. These functions are compliant with the POSIX 1003.1 standard, as applicable to network–attached terminal devices.

Network Terminal Accelerator Devices

The Network Terminal Accelerator subsystem maps to several AIX logical devices. Each adapter in a system must have a corresponding "raw" device named in the /**dev** directory. Any of the commands that configure or report status on an adapter use this raw device interface. For example, the **ntx_load** command opens a raw device to load the microcode onto each adapter. All Network Terminal Accelerator commands that open a raw interface to an adapter default to /**dev**/**rhp0**. If you have more than one Network Terminal Accelerator adapter installed in your system, you can use the **ntx_load** *–d* command to override the default and select another raw device name.

For each adapter installed in a system, there can be up to 256 child devices (2048 on some systems). Each of these corresponds to one channel on the associated adapter. Once the adapter is downloaded and configured, the child devices implement standard interfaces to I/O ports on terminal servers on the network.

Network Terminal Accelerator supports three child device types: hty, tty, and lp. Support for the "generic" tty and lp devices allows for the configuration of channels with these familiar interfaces.

Both hty and tty devices provide the same termio interface. The only difference between them is the name used for the device node in the /**dev** directory and the SMIT panels used to add, delete, and configure them. The tty child devices can be configured through SMIT using the **smit tty** fast path. The hty child devices can be configured through SMIT using the **smit htyDevice_menu** fast path.

Printer devices (/**dev**/**lp***) provide a standard AIX line–printer interface and are configured using the **smit pdp** fast path.

Simple Network Management Protocol (SNMP)

The adapter provides an SNMP agent on the adapter for managing the adapter and its associated software. This agent does not manage the host or host software. Once configured, this agent responds to protocol data units (PDUs) received from the network. The adapter's SNMP agent complies with Management Information Base (MIB) II (request for comment 1213).

Network Terminal Accelerator Installation and Configuration

The following information helps system administrators and Network Terminal Accelerator software application writers understand the installation and configuration support requirements for the adapter program. If you want to install and configure your adapter, refer to the following procedures.

- Installing the Network Terminal Accelerator Adapter Software, on page 9-6.
- Configuring the Network Terminal Accelerator, on page 9-7.
- Configuring hty Devices, on page 9-8.
- Enabling New hty Devices, on page 9-8.
- Configuring the MAC Layer Interface (MLI), on page 9-9.
- Configuring Network Terminal Accelerator SNMP, on page 9-10.
- Removing the Network Terminal Accelerator Software, on page 9-11.

The Network Terminal Accelerator adapter software for AIX Version 4.1.5 and later is included in the Base Operating System (BOS) installation media. If you install AIX Version 4.1.5 or later on a system that has the adapter hardware already installed, the adapter software will be automatically installed during the AIX installation. In this case, you can proceed to the section on ”Configuring the Network Terminal Accelerator”.

Note: If the adapter software is installed during an AIX migration install, the host system must be restarted after completing migration to reset the adapters with the saved configuration data.

If you want to install the Network Terminal Accelerator hardware in a system that is already running AIX Version 4.1.5 or later, use the procedure that follows. Before you install the adapter hardware, you should first install the adapter software.

Installing the Network Terminal Accelerator Adapter Software

The following information explains how to install the adapter software. After installing the adapter software, you will need to restart your system.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. In AIX Versions 3.2 through 4.2, man pages for the adapter commands are included in the InfoExplorer package, **bos.info**.
4. Message catalogs for the Network Terminal Accelerator utility programs and diagnostics are included in the Device Messages package, **devices.msg**, available on AIX Version 4.2 and later.

Installation Using SMIT

1. Use the **smit devinst** fast path to access the **Install Additional Device Software** menu.
2. At the **INPUT device** / **directory for software** field, select the device that has the software.
3. Select the software to install:
   - **devices.common.IBM.ntx** and **devices.mca.0072** for the NTX–256
   - **devices.common.IBM.ntx** and **devices.mca.0095** for the NTX–2048.
4. Select **Do** to start loading the software**.**
5. When the software is finished loading, exit the SMIT interface.

6. Shut down the host machine from the command line using **shutdown –F**.
7. Install the adapter. Refer to *Adapters Information for Micro Channel Architecture Systems* for more information.
8. Restart the host machine.

Installation from the Command Line

1. Make a new directory or change to an appropriate existing directory where you can copy the **installp** images of the Network Terminal Accelerator software.
2. Copy the **installp** images to the directory.
3. Run the **installp –acgqXd** *Dir File* **2>&1** command. *Dir* specifies the directory containing the image. *File* specifies one of the following files:
   - **devices.common.IBM.ntx**
   - **devices.mca.0072**
   - **devices.mca.0095**

   The **installp** command proceeds to install the software.
4. Shut down the host machine with the following command:

   ```
   shutdown –F
   ```
5. Install the adapter. Refer to *Adapters Information for Micro Channel Architecture Systems*, for more information.
6. Restart the host machine.

Configuring the Network Terminal Accelerator

The following information explains the steps, using the System Management Interface Tool (SMIT), to configure the adapter.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. The Network Terminal Accelerator adapter card must be in the host machine.
4. The Network Terminal Accelerator adapter software must be installed.

Procedure

1. Use the **smit ntxConfig_menu** fast path to access the **Network Terminal Accelerator Configuration** menu.
2. Select **Global Modes**.
3. Select a device. The default is the **rhp0** device.
4. Enter **Yes** for the **Commit the Changes** field if any of the default values are overridden.
5. Enter a value in the **Number of Channels to Configure** field for this device. The default value is **64**.
6. Select **Do**.
7. Return to the **Network Terminal Accelerator Configuration** menu using the **Cancel** option.
8. Select **Network Configuration**.
9. Select a device. The default is the **rhp0** device.
10. Enter the **Internet Address** for the adapter hty interface.

11. Select **Do**.
12. Exit SMIT.

If a host name is required for the hty interface, edit the /**etc**/**hosts** file to add a host name for the specified IP address. If you have additional adapters, configure them now. Use the **ntx_ifcfg** command to verify that the network configuration completed.

Configuring hty Devices

The following information explains the steps, using SMIT, to add hty devices to the adapter.

Notes:

1. The installation default sizes of the / (root) and the /**var** file systems are too small for large increases in the number of /**dev**/**hty***XX* entries.

   If the / (root) and the /**var** file systems are too large, the machine crashes after reboot (usually with an 0c9 code).
2. Increasing the number of htys substantially increases the paging space requirements. Paging space problems may cause your system to crash during reboot just after the system console is available.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. The Network Terminal Accelerator adapter must be in the host machine.
4. The Network Terminal Accelerator adapter software must be installed.

Procedure

1. Use the **smit htyDevice_menu** fast path to access the **Hty Devices** menu.
2. Select **Add HTY Devices**.
3. At the Network Terminal Accelerator Adapter field, use the **List** option to display a list of adapters. Select the adapter to which the hty device will be added.
4. Specify a single channel number (for example, **5**), a range of channel numbers (for example, **0–15**), or the word **all** in the Channels to Configure field. An hty device will be created and made available on each specified channel.
5. Select **Do**.
6. Exit SMIT.

Enabling New hty Devices

After you have configured the Network Terminal Accelerator adapter, you must modify the /**etc**/**inittab** file to enable newly created hty devices. You can use your favorite editor to edit the /**etc**/**inittab** file, or you can use the **penable** command. The following steps will help you to enable newly created devices.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. The Network Terminal Accelerator adapter must be in the host machine.
4. The Network Terminal Accelerator adapter software must be installed and configured.

Using the penable Command

1. Issue the **penable** command for each hty in the **/etc/inittab** file as follows:

```
penable hty0
```

Alternatively, you can enable all the htys in the file using the following script:

```
for i in /dev/hty*
do
penable `basename $i`
echo "penable $i"
done
```

2. Ensure that you have successfully enabled the htys by issuing the following command:

```
ps -ef | grep hty
```

The system should return something similar to the following:

```
root 1666     1  0  17:08:01  ?    0:00  /usr/sbin/getty /dev/hty1
root 9999  5852 2  17:12:22 hft/0 0:00  grep hty
root 10372   1  0 17:08:04   ?    0:00  /usr/sbin/getty /dev/hty2
root 10624   1  0 17:07:56   ?    0:00  /usr/sbin/getty /dev/hty0
```

3. Check the configuration by using the **telnet** or **rlogin** command to contact the host machine from a remote host. You should specify the Internet address for the adapter card, not the host machine. If you are successful, you should see a login window and, after supplying a login ID and password, you should see a command prompt.
4. Run the **tty** command to confirm that you are running on an hty.

   The normal getty process does not check the **$HOME/.rhosts** or **/etc/hosts.equiv** file for the rlogin's automatic login feature. Run the **/usr/sbin/getty_hty** command to enable getty processes that check the **$HOME/.rhosts** file after receiving an **rlogin** request.

Using an Editor

1. Open an editing session on the **/etc/inittab** with your favorite editor.
2. Find the lines that define the htys.
3. Change the word `off` to `respawn` for each line.
4. Save and close the **/etc/inittab** file.
5. Issue the following command to reinitialize the processes in the **inittab** file:

```
telinit q
```

Configuring the MAC Layer Interface (MLI)

The following explains how to use the System Management Interface Tool (SMIT) to configure the MLI side of the Network Terminal Accelerator adapter.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. The Network Terminal Accelerator adapter card must be in the host machine.
4. The Network Terminal Accelerator adapter software must be installed.
5. The Network Terminal Accelerator adapter software must be configured for the hty interface.

Procedure

1. Use the **smit ntx_menu** fast path.
2. Select **MAC Layer Interface (MLI) Configuration**.
3. Select the interface for the MAC layer.
4. Enter the INTERNET ADDRESS for the adapter MAC Layer Interface. .

   Note: This IP address must be different from the IP address of the adapter hty interface.
5. Specify **up** for the Current STATE field.
6. Select **Do**.
7. Exit SMIT.

If a host name is required for the MLI, enter **smit tcpip** to get to the standard TCP/IP SMIT menu. Select **Minimum Configuration** to specify the host name.

Note: If you use a name server, use different host names for hty and the MLIs.

Configuring Network Terminal Accelerator SNMP

The following explains how to use the System Management Interface Tool (SMIT) to configure SNMP on the adapter.

Prerequisites

1. You must have root user authority.
2. AIX must be installed and running.
3. The Network Terminal Accelerator adapter card must be in the host machine.
4. The Network Terminal Accelerator adapter software must be installed.

Procedure

1. Use the **smit ntxSnmp_menu** fast path to access the **SNMP Configuration** menu.
2. Select **SNMP Status**.
3. Select the device. The default device is **rhp0**.
4. Specify the desired SNMP Agent State.
5. Select **Do**.
6. Return to the **SNMP Configuration** menu using the **Cancel** option.
7. If the SNMP Agent State field was set to **on**, use the remaining menus to complete the configuration of SNMP Communities, Network Management Station authentication, Traps, and Site Specific Variables.
8. Exit SMIT.

Files

/usr/include/st.snmp.h	Contains SNMP implementation information for users who need to know MIB information and data structures.

Removing the Network Terminal Accelerator Software

The following steps will help you remove the software for upgrades or remove the package from your system.

Prerequisites

1. You must have root user authority.
2. Close all adapter child device channels and wait until there is no outstanding I/O.
3. Detach the network interface driver using the **chdev –l en***X* **–a state=detach** command.
4. Remove the adapter device using the **rmdev –l rhp***X* **–R –d** command.
5. Remove the network interface driver using the **rmdev –l en***X* **–d** command.

Procedure

1. On AIX Version 4.1.5 systems, use the **smit install_remove** fast path to access the **Remove Software Products** menu. On AIX Version 4.2 and later systems, use the **smit remove** fast path to access the **Remove Installed Software** menu.
2. Enter the SOFTWARE name of the fileset to remove:
   - **devices.common.IBM.ntx.asw**
   - **devices.common.IBM.ntx.rte**
   - **devices.mca.0072.diag**
   - **devices.mca.0072.rte**
   - **devices.mca.0095.diag**
   - **devices.mca.0095.rte**

   Or select the **List** option for a list of possible software filesets to remove.
3. Set PREVIEW only? to **no**.
4. Select **Do**.
5. Select **Exit** to exit the SMIT interface.

Network Terminal Accelerator Installation and Configuration Support

The following information helps system administrators and Network Terminal Accelerator software application writers understand the installation and configuration support requirements for the adapter program.

This section discusses:

Software Distribution

The adapter directories and files contain the sample programs and executable files to support the adapter. The software distribution contains the following components:

/usr/lib/drivers/htydd Specifies the adapter's tty device driver.

/usr/lib/drivers/rhpdd Specifies the adapter's raw device driver.

/usr/lib/drivers/mlidd Specifies the MAC Layer Interface (MLI) Ethernet device driver.

/usr/lib/methods/* Specifies the AIX device driver configuration methods.

support Specifies the AIX error logging and trace templates.

commands Specifies the programs to set and get the adapter's configuration.

man pages Specifies the man pages that describe the adapter commands features and usage.

download image Specifies the software image that is downloaded to the adapter.

Target Directories

The executable images from the software distribution are installed in the following directories:

/usr/lib/drivers Specifies the directory where the **htydd**, **mlidd**, and **rhpdd** device drivers reside.

/usr/lib/methods Specifies the directory where the device driver configuration methods reside.

/usr/bin Specifies the directory where the adapter's commands reside.

/usr/lpp/info Specifies the directory where the adapter's reference pages reside on an AIX Version 3.2, 4.1, or 4.2 system.

Note: In AIX Versions 3.2 through 4.2, the InfoExplorer package must be installed to access the reference pages.

/usr/lib/microcode Specifies the directory where the download image resides.

/etc Specifies the directory where the **hty_config** and **rc.ntx** files reside.

System Configuration

The ODM database maintains the system configuration and driver support. The system configuration includes ODM device object classes and configuration methods that manage the database for the adapter. Additionally, the **/etc/inittab** file is modified during the adapter's installation to support the product's initialization.

Device Driver Configuration Methods

The device driver configuration methods manage the ODM database for the adapter. Each device driver has a set of configuration methods. They define the run–time attributes and channel configuration to the AIX system. The following table contains the device drivers and methods.

Device Driver Configuration Methods			
Driver	**Define**	**Configure**	**Change**
htydd	**defhty/udefhty**	**cfgtty/ucfghty**	**chghty**
mlidd	**defmli/udefmli**	**cfgmli/ucfgmli**	*none*
rhpdd	**defrhp/udefrhp**	**cfgrhp/ucfgrhp**	**chgrhp**

The configuration methods are run automatically during a system restart. To run them from the command line, use the **cfgmgr** command.

Inittab File

The **/etc/inittab** file contains entries to execute adapter initialization commands and to enable logins. The **/etc/rc.ntx** file is called by the **inittab** file and contains the adapter's initialization commands.

The **/etc/rc.ntx** file contains shell script command lines. The commands run at the conclusion of the system boot procedure. You can use the **/etc/rc.ntx** file to perform site–specific configuration commands such as setting the adapter Internet addresses, default route, and hty device call–type and protocol–port parameters.

The **/etc/inittab** file contains hty device entries. These entries identify the hty device, run level, action, and getty login process. The default run level is 2, and the default action is off. The **penable** and **pdisable** commands enable and disable logins for an hty device. The following hty device entry defines hty device 7:

```
hty7:2:off:/usr/sbin/getty /dev/hty7
```

Refer to the **rlogin** command information in "Host Commands", on page 9-25 for information on how to enable automatic logins.

Man Pages

The Network Terminal Accelerator software is distributed with AIX–style man pages. The man pages describe the adapter commands, library routines, and special files. Refer to *AIX Commands Reference* for more information on the Network Terminal Accelerator commands.

Note: In AIX Versions 3.2 through 4.2, the InfoExplorer package must be installed to access the man pages.

Download

Two versions of the download image file are installed in the **/usr/lib/microcode** directory. The **ntx8023.04F.00** file is downloaded when the adapter is configured with the TCP Protocol Timeout value set to **long** (default value). Otherwise, the **ntx023.04F.01** file is downloaded. The **/usr/bin/ntx_load** command transfers the download image to the adapter. The **cfgrhp** configure method runs the **ntx_load** command to download the adapter at system startup.

You must restart the system after the software installation. The reboot procedure loads the new drivers and downloads the adapter's program. To verify the driver software installation and download image, use the **ntx_ver** command after the system boots. The following command retrieves and displays the software version information from the adapter drivers and download image for the **/dev/rhp1** adapter:

```
ntx_ver -d/dev/rhp1

Title: NTX-8023 Download Software
Version: 04F
Date: Dec 28 09:49:43 1995
Description: TCS/Telnet with TCP/IP

Title: rhp
Version: 01A
Creation Date: Feb 6 18:54:36 1996
Description: Network Terminal Accelerator Raw Driver

Title: mli
Version: 01A
Creation Date: Feb 6 18:55:59 1996
Description: Network Terminal Accelerator MLI Driver
```

The adapter program does not accept download requests until the self–test sequence completes.

Terminal Connectivity Services

The following terms are essential to understanding how the Network Terminal Accelerator adapter's terminal services configuration works:

Channel One of the 256 or 2048 logical channels supported by each adapter (depending on the adapter's memory size). Channels are always numbered consecutively from 0 to 255 (or 2047) on each adapter. Each channel may be *open* or *closed*. Open channels may be waiting for connection requests, attempting to establish outgoing connections, or connected to a remote terminal. Before any data can pass through a channel to or from a user terminal (or printer), it must be *open* and *connected*.

Ports Identifies network connection endpoints. Ports are numbered from 0 to 65535. Some special port numbers have been established. For example, requests for connections to the Telnet service are normally directed to port number 23, while those for rlogin service are directed to port number 513. By default, the Network Terminal Accelerator adapter follows these well–known port numbering conventions.

Addresses Refers to the IP address of a node on the network. Normally, each adapter is assigned a single unique address. When a channel is connected, the connection is uniquely identified by the address/port pair at the remote end and the address/port pair on the Network Terminal Accelerator adapter. Multiple connected channels on the adapter can share the same local port number. They are still different connections, however, since each is connected to a different remote address/port.

Services Refers to the collection of higher level protocols to which users can request connections. Currently, the Network Terminal Accelerator adapter offers Telnet, rlogin, and transparent (no operation, or nop) services.

Devices Logical I/O entities on the host system, as named in the system's /**dev** directory. The AIX logical I/O device is mapped to an adapter channel when the device is created. **Devices** can also be used as a generic reference to various pieces of equipment involved with the adapter's operation (for example, network devices).

The Configuring Channels and Services figure illustrates the logical relationship among devices, channels, ports, and services for an adapter running in a multi–user AIX host. Several channels are open and connected to remote user terminals. One channel is connected to a remote printer. Several other channels are open, but not connected.

Channel Definition

The adapter's software provides standard Telnet and rlogin protocol services to support network virtual terminal sessions for hty devices. The AIX **htydd** device driver in conjunction with the adapter's terminal control software (TCS) manage the session logins.

The **htydd** device driver interfaces the host applications to the TCS and manages the POSIX–line discipline emulation. The TCS performs the line–discipline character processing on the input and output data streams. The tty configuration defines the hty device characteristics.

The hty channels are allocated to the adapters and defined to the **htydd** device driver and TCS with the /**usr**/**bin**/**hty_load** command. The **hty_load** command reads the /**etc**/**hty_config** configuration file, which contains the hty channel assignment for the adapters in the system.

The Network Terminal Accelerator configuration methods generate the **hty_config** file based on each adapter's **no_of_htys** attribute. You can set this attribute in the Number of Channels to Configure field in the **Set GLOBAL Modes** SMIT menu.

The **hty_config** file has the following format:

- Comment lines start with the # (pound sign) in the first column.
- Blank lines are ignored.
- Fields for each entry are delimited by tabs or spaces.
- One entry for each adapter.
- The first field of an entry is the adapter minor number `X` from /**dev**/**rhp***X*.
- The second field is reserved and must always equal 1.
- The third field is the number of hty devices. For the Network Terminal Accelerator–256, this may be any number from 1 to 256. For Network Terminal Accelerator–2048, this may be any number from 1 to 2048.

This is a sample **hty_config** file. It defines 256 hty channels for the /**dev**/**rhp0** adapter, 2048 hty channels for the /**dev**/**rhp1**, and 700 hty channels for /**dev**/**rhp2**. Both /**dev**/**rhp1** and /**dev**/**rhp2** must be Network Terminal Accelerator–2048 models for the configuration to be valid.

```
# Adapter      Cluster           Number
# minor #      address           of channels
    0             1                256
    1             1                2048
    2             1                700
```

Default Protocol–Port Services

The adapter's software binds services to numbered protocol–ports. The **ntx_pserv** command modifies and displays the protocol–port assignments. The following list contains the default protocol–port and service relationships:

Service	Port
Telnet	TCP port 23
rlogin	TCP port 513
SNMP	UDP port 161
SNMP–trap	UDP port 162

Configuring Channels and Services

The default channel configuration for the adapter TCS is call–in mode with dynamic channel assignment. In call–in mode, the TCS accepts connection requests from user terminals and establishes a connection to any defined service. The default channel protocol–port number is 0, which specifies dynamic assignment of sessions to channels based on the port to which the original incoming connection request was directed. For example, when an incoming connection request arrives on port 23, the TCS will automatically assign its Telnet service to handle data that passes through that connection. The first available channel supporting the Telnet service is selected for the connection.

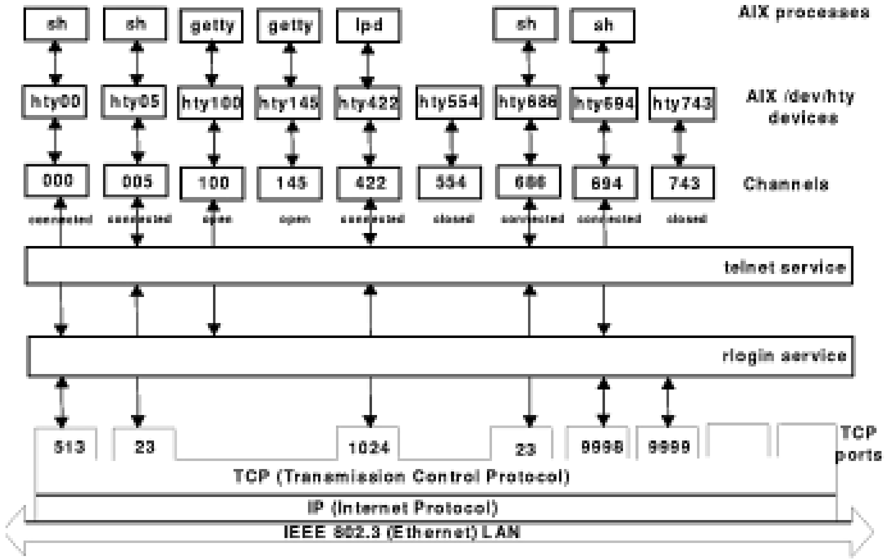
If the remote device is passive (for example, a printer or dial–out modem) and cannot initiate the connection request, the hty channel may be configured in call–out mode. In this mode, the TCS initiates the connection request to a specific remote IP address and protocol port number whenever an application opens the hty device mapped to that channel.

Call–in channels may be configured for dynamic or fixed channel assignment. Call–out channels always use fixed channel assignment.

Dynamic Assignment for Call–In Channels

The Network Terminal Accelerator adapter can accept connection requests from terminal servers or remote clients and dynamically assign each session to an available channel. In this mode, the adapter waits for incoming connection requests. The first available open channel is assigned as requests arrive at the adapter. Since the correspondence between the channels and the AIX devices is fixed, a terminal may become connected to any **/dev/hty** device, depending on when you logged on.

In the Configuring Channels and Services figure, on page 0, `channels 000`, `005`, and `686` were connected to terminals in this fashion. At the time the session on `channel 000` was established (when an rlogin session on the host was requested), `channel 000` was the lowest–numbered open, unconnected channel available for an rlogin connection. The initial request arrived on `TCP port 513` (the "well–known" rlogin port) and the adapter's rlogin service honored the request by establishing a new rlogin session on `channel 000`. The session on `channel 005` was set up in exactly the same way. In this case, the initial request was for a Telnet session and arrived on `TCP port 23`.



Configuring Channels and Services

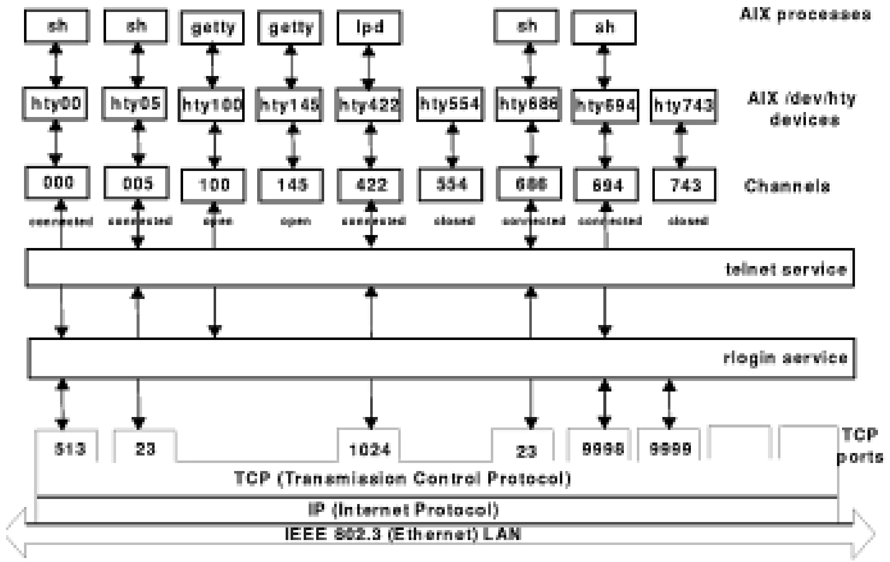
Also, as illustrated in the ”Configuring Channels and Services” figure, the adapter will honor the next request that arrives on `port 23` for a Telnet session by connecting `channel 145` to the Telnet service. Note that `channel 100` is unavailable to the Telnet service since it is reserved for exclusive use of the rlogin service.

Fixed Assignment for Call–In Channels

It may be necessary to establish a fixed relationship between a particular terminal (or printer or modem) on the network and a particular AIX logical device. For example, a print spooler must know which device to open to send output to its printer. In some applications, it may be desirable to establish such a fixed relationship for user terminals as well.

Remote devices normally direct their connection requests to ”well known” port numbers on the Network Terminal Accelerator adapter (for example, `port 23` for Telnet and `port 513` for rlogin). The adapter assigns these connections to the first available open channel. If desired, channels can be assigned specific address/port pairs. Remote users can then direct their connection requests to a specific address or port and know the channel (/**dev** device) to which they will be connected.

In the ”Configuring Channels and Services” figure, on page 0, `channel 694` is configured to only accept rlogin connection requests directed specifically to `TCP port 9998`. To make proper use of this facility, the system administrator can take steps to ensure that only one user requests connections to this particular port by, for example, appropriately configuring the user’s terminal server.



Configuring Channels and Services

To establish the configuration for `channel 694` as shown in the "Configuring Channels and Services" figure, on page 0, both the **ntx_pserv** and **ntx_chaddr** commands must be used. The **ntx_pserv** command can be run any time after the board is downloaded (even before the **ntx_ifcfg** command is run). It associates Network Terminal Accelerator services with ports or ranges of ports. For example, the following would assign the rlogin service to ports `9998` and `9999`:

```
%ntx_pserv rlogin 9998-9999 tcp
```

After the **hty_load** command establishes the channel configuration, use the following command to associate channel 694 with port 9998:

```
%ntx_chaddr –p 9998 694
```

In both examples, assume that the adapter being configured is the default `/dev/rhp0`.

Use the following procedures to configure call–in channels for fixed assignment to addresses and/or ports. When the desired configuration has been established and tested, you must add the **ntx_pserv** and/or **ntx_chaddr** commands to the end of the **/etc/rc.ntx** file to restore the configuration after every system boot.

Fixed Assignment Using Address/Port Number Pairs from the Command Line

1. Use the **ntx_pserv** command to assign the desired port numbers to the Telnet or rlogin service. Use port numbers > 2048, since port numbers 0–2048 are reserved by convention and may be defined for other services.
2. Use the **ntx_chaddr** command to assign one of the defined port numbers to a channel. The assignment of an IP address is optional with **ntx_chaddr**. If no IP address is specified, the channel remains assigned to the default IP address for the adapter hty interface (as shown by the **ntx_ifcfg** command).
3. On the remote terminal or client, enter **telnet** *IP_Address Port_Number* to establish a connection with the hty channel specified in step 2.
4. After testing the new configuration, add the **ntx_pserv** and **ntx_chaddr** commands to the **/etc/rc.ntx** file.

Fixed Assignment Using Address/Port Number Pairs from SMIT

1. Use the **ntx_pserv** command to assign the desired port numbers to the Telnet or rlogin service. Use port numbers > 2048, since port numbers 0–2048 are reserved by convention and may be defined for other services.
2. Use the **smit htyDevice_menu** fast path to access the **HTY Devices** menu.
3. Select **Configure HTY Channels**.
4. Enter the adapter name in the Network Terminal Accelerator Adapter field.
5. Accept **Call–in** as the Call Type field value.
6. Leave the Service field blank.
7. Enter the desired Protocol Port Number and Channel Number.
8. Enter the desired Internet Address. This field is optional.
9. On the remote terminal or client, enter **telnet** *IP_Address Port_Number* to establish a connection with the hty channel specified in step 7.
10. After testing the new configuration, add the **ntx_pserv** and **ntx_chaddr** commands to the /**etc**/**rc.ntx** file.

Fixed Assignment Using Addresses from the Command Line

1. Use the **ntx_chaddr** command to assign a unique IP address to each channel requiring fixed assignment. The default IP address for the adapter hty interface (as shown by the **ntx_ifcfg** command) remains assigned to all other channels.
2. On the remote terminal or client, enter **telnet** *IP_Address* to establish a connection with the hty channel specified in step 1.
3. After testing the new configuration, add the **ntx_chaddr** command to the /**etc**/**rc.ntx** file.

Fixed Assignment Using Addresses from SMIT

1. Use the **smit htyDevice_menu** fast path to access the **HTY Devices** menu.
2. Select **Configure HTY Channels**.
3. Enter the adapter name in the Network Terminal Accelerator Adapter field.
4. Accept **Call–in** as the Call Type field value.
5. Leave the Service field blank.
6. Enter the desired Internet Address and Channel Number.
7. On the remote terminal or client, enter **telnet** *IP_Address* to establish a connection with the hty channel specified in step 6.
8. After testing the new configuration, add the **ntx_chaddr** command to the /**etc**/**rc.ntx** file.

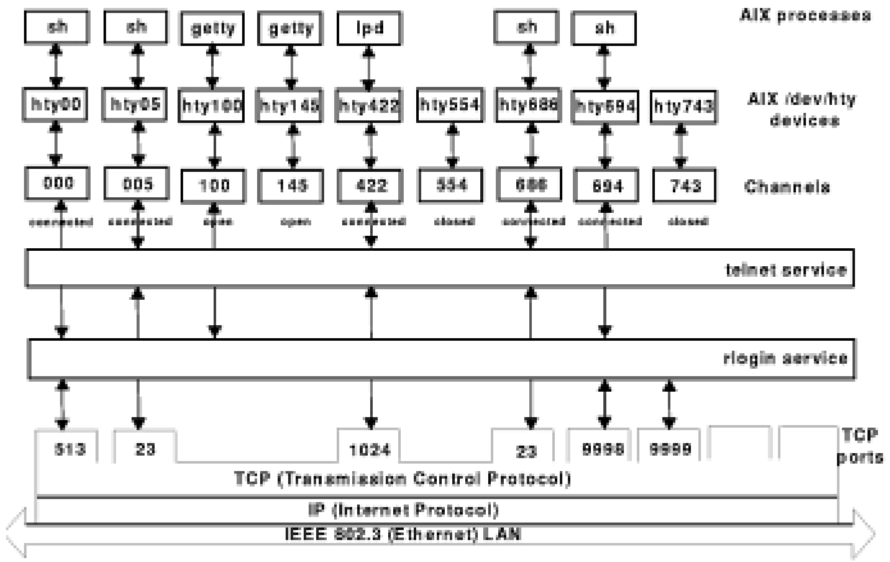
Fixed Assignment for Call–Out Channels

The call–out mode can be used to establish a fixed relationship between remote devices and Network Terminal Accelerator adapter channels. This method is best suited for passive devices, such as printers and dial–out modems, which cannot take the initiative to request a connection. This feature can also be used to establish connections to user terminals.

When a host process (for example, getty or a print spooler) opens a device mapped to a call–out channel, the adapter attempts to connect to a predefined remote address and port. If the connection is established, the host process is able to transfer data to and from the remote port.

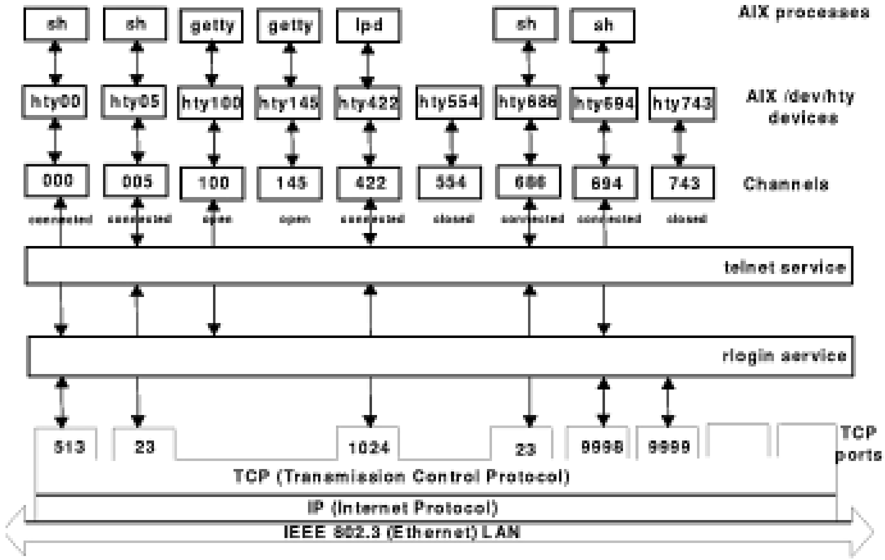
To operate channels in call–out mode, the channel must be configured to make an outgoing connection request to a specific address and port, and the remote device must be configured to accept an incoming connection request on the same address and port.

In the ”Configuring Channels and Services” figure, on page 0, `channel 422` has been configured for call–out using the Telnet service. This channel has been opened by the system line–printer daemon. At the remote end of this connection, there is a terminal server with an attached printer. The adapter automatically assigns port numbers to its end of the call–out connections. Ports are assigned starting at number 1024 and working consecutively upward.



Configuring Channels and Services

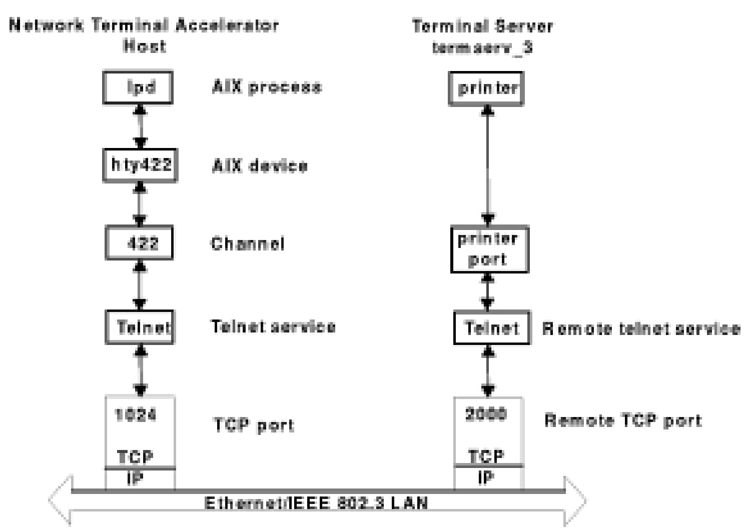
To establish the configuration for `channel 422` as shown in the ”Configuring Channel and Services” figure, on page 0, enter:



Configuring Channels and Services

```
%ntx_chaddr -c telnet -p 2000 -i termserv_3 422
```

The host must be able to resolve the name `termserv_3` to a reachable IP address (the address of the terminal server to which the line printer is connected). Also, the terminal server must be configured to accept a Telnet connection request on its `port 2000`, and honor this request by establishing a Telnet session with the physical I/O port to which the printer is attached. The following figure shows both ends of such a printer connection after it has been established.



Channel 422 Connected to a Remote Printer

Use the following procedure to configure call–out channels for fixed assignment to addresses and ports. When the desired configuration has been established and tested, you must add the **ntx_chaddr** command to the end of the **/etc/rc.ntx** file to restore the configuration after every system boot.

Fixed Assignment Using Addresses and Ports from the Command Line

1. Use the **ntx_chaddr** command to specify the remote service and IP address for each channel requiring fixed assignment. Specifying a service enables the channel as call–out. The assignment of a port number is optional. If no port number is specified, the ”well–known” port number for the service is used.
2. On the remote terminal server or client system, configure the same service, port, and IP address for the remote device to be connected with the hty channel specified in step 1.
3. After testing the new configuration, add the **ntx_chaddr** command to the /**etc**/**rc.ntx** file.

Fixed Assignment Using Addresses and Ports from SMIT

1. Use the **smit htyDevice_menu** fast path to access the **HTY Devices** menu.
2. Select **Configure HTY Channels**.
3. Enter the adapter name in the Network Terminal Accelerator Adapter field.
4. Enter **Call–out** in the Call Type field.
5. Enter the desired Service, Internet Address, and Channel Number.
6. Enter the desired Protocol Port Number. This field is optional.
7. After testing the new configuration, add the **ntx_chaddr** command to the /**etc**/**rc.ntx** file.

TCP/IP Configuration

The adapter supports network virtual terminal sessions on hty devices in a TCP/IP environment. You must set the adapter's Internet address for this hty interface in the ODM database to enable the Ethernet interface controller. You set the Internet address during installation using the System Management Interface Tool (SMIT). After installation, the configuration methods run the **ntx_ifcfg** command at system startup. The **ntx_ifcfg** command modifies and displays the adapter's network interface and broadcast addresses for the hty interface.

You can set the network address manually using the **ntx_ifcfg** command. The following command configures the /**dev**/**rhp1** adapter with Internet address 192.6.153.90:

```
ntx_ifcfg –d/dev/rhp1 –i 192.6.153.90
```

You can also use the **ntx_ifcfg** command to display the adapter's Internet address and state of the adapter's network interface. The following commands retrieve and display the Internet addresses for the /**dev**/**rhp1** and /**dev**/**rhp2** adapters:

```
ntx_ifcfg –d/dev/rhp1
ie0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 192.6.153.90 netmask ffffff00 broadcast 192.6.153.255

ntx_ifcfg –d/dev/rhp2
ie0: flags=02<BROADCAST>
inet 0.0.0.0 netmask 00000000 broadcast 0.0.0.0
```

The **ntx_ifcfg** command reports that no Internet address has been specified for the /**dev**/**rhp2** adapter.

The **ntx_ping** command verifies the operational state of the adapter's Ethernet controller and network connection. It asks the adapter to send an Internet Control Message Protocol (ICMP) echo request datagram (ping) to an active host on the network.

Note: The **ntx_ping** command accepts the dotted decimal Internet address notation or host name from the /**etc**/**hosts** file.

The following example displays both the command that pings a host machine named `bach` and the system response to the command:

```
ntx_ping bach
bach is alive
```

In addition to the **ntx_ifcfg** and **ntx_ping** commands, the adapter software provides the following commands to manage adapters in a TCP/IP environment:

ntx_route Sets and flushes the accelerator's network routing tables. You can use this command to update the accelerator's network routing tables from the host.

ntx_arp Modifies and displays the accelerator's Internet–to–Ethernet address translation table. This command provides adapter support for the Address Resolution Protocol (ARP). ARP translates Internet addresses to Ethernet addresses.

ntx_ttype Modifies the accelerator's terminal–type database. The adapter maintains a terminal type database. The Telnet service uses this database to retrieve terminal types that are known to the host.

ntx_stat Retrieves and resets the network statistics. The adapter maintains a set of statistics that report various network activity parameters. The **ntx_stat** command reports statistics on the following:

Ethernet controller

TCP, IP, UDP, IFNET, and ICMP

Route

MLI Ethernet Interface

The adapter's MAC Layer Interface (MLI) is an Ethernet device driver for the host TCP/IP network interface driver. The network interface driver is identified as **en***X*, where *X* equals 0 for the primary network interface, 1 for the secondary network interface, and so forth. The MLI devices are identified as /**dev**/**ent***X*, where *X* is the MLI device minor number.

The **ifconfig** command binds the /**dev**/**en***X* network interface driver to the /**dev**/**ent***X* Ethernet device driver. The following command enables the primary network interface driver on MLI device /**dev**/**ent0**:

```
ifconfig en0 HostName
```

where the /**etc**/**hosts** file defines the host specified by the *HostName* parameter.

The following command disables the primary network interface driver on MLI device /**dev**/**ent0**:

```
ifconfig en0 detach
```

SNMP Management

The adapter program supports the Simple Network Management Protocol (SNMP). The adapter's SNMP agent complies with the Management Information Base (MIB) II (RFC 1213) and SNMP protocol (RFC 1157). The following commands allow you to manage and configure the adapter's SNMP:

ntx_snmp Starts and stops the SNMP agent. The SNMP agent must be started from the host. You can also use this command to display the state of the SNMP agent.

ntx_comun Configures the SNMP community profile. The default community profile has read–only access to the entire MIB for the community public.

ntx_nms Configures Network Management Station (NMS) authentication for the SNMP agent. You can configure the SNMP agent to allow specific NMSs to access the SNMP agent.

ntx_traps Configures SNMP traps. The host may enable the SNMP agent to generate traps.

ntx_descr Assigns a site–specific value to an SNMP MIB variable in the SNMP agent. The host may configure site–specific information for the SNMP MIB.

Host Commands

The **telnet** command, the **rlogin** command, and the nop service provide support for the adapter host. This section discusses these utilities.

Note: The following information repeatedly refers to an environment buffer. The adapter stores information in this environment buffer.

The telnet Command

The **telnet** command negotiates the window size (NAWS option, RFC–1073) of the remote user. If the remote accepts the NAWS option, it sends NAWS suboptions when the user's window size changes. When the adapter receives NAWS suboptions, it stores the last received value in the environment buffer.

The **telnet** command also negotiates the terminal type (RFC–1091, TTYPE option) of the remote user. If the remote agrees to send terminal types, the adapter sends TTYPE SEND suboptions until it finds a case–insensitive match in the terminal database or until the remote has sent all of its types. When the adapter finds a match, it saves that terminal type.

If the remote host sends `ANSI`, and `ansi` is in the database, the adapter forms a match and saves `ansi` (not `ANSI`) as the terminal type. If the remote sends all of its types (last type sent twice) and the adapter finds no match, the adapter asks for the type again. The remote starts over, sending its first type (its *preferred* type), which the adapter saves.

Additionally, the **telnet** command negotiates the X display location (RFC–1096, XDISPLOC option) of the remote user. If the remote agrees to send the X display location, the adapter sends the XDISPLOC SEND suboption and stores the response in the channel environment buffer.

Finally, the **telnet** command negotiates the terminal speed (RFC–1079, TSPEED option) of the remote user. If the remote agrees to send the terminal speed, the adapter sends the TSPEED SEND suboption and stores the response in the channel's environment buffer. The response contains the comma–separated input and output speeds.

The first speed is stored as the output speed and the second as the input speed. The **telnet** command sends the XON and XANY suboptions if the remote accepts the toggle remote flow control option (RFC–1372, TOGGLEFC option). Previously, the **telnet** command would only send the ON and OFF suboptions (RFC–1080).

The **/usr/sbin/getty_hty** command passes the preceding user and host information to the **/bin/login** command when using the **telnet** command. Sample source code for the **getty_hty** command resides in the **/usr/samples/ntx** directory. Do not issue the **getty_hty** command from the command line. Refer to the ”Enabling the getty_hty Command”, on page 9-26 for procedural details on how to use this command.

The rlogin Command

When the adapter receives window size commands, it stores the last received value in the environment buffer.

When the adapter receives the last environment string (the terminal information), it sends an AE_ENVIRO event to the device driver if the host set the **ENV_EVENT** flag.

The **/usr/sbin/getty_hty** command allows automatic login when using the **rlogin** command. Sample source code for the **getty_hty** command resides in the **/usr/samples/ntx** directory. Do not issue the **getty_hty** command from the command line. Refer to ”Enabling the getty_hty Command”, on page 9-26 for procedural details on how to use this command.

The nop Service

The nop service is transparent and has no extra protocol associated with it. Therefore, the adapter cannot obtain any remote user environment information.

Enabling the getty_hty Command

The **getty_hty** command is a sample program that passes user and host information for the **telnet** command and facilitates automatic login for the **rlogin** command.

Note: The **/usr/sbin/getty_hty** command and **/usr/samples/ntx/getty_hty.c** sample source code do not enable the Ctrl–D key sequence for ending a session. Execute the **exit** command to end a session. Alternatively, you can add the following line to the **$HOME/.profile** or **$HOME/.login** file to enable the Ctrl–D key sequence upon login:

```
stty eof ^D
```

Prerequisites

1. You must have root user authority.
2. You must set the **User Environment Events** field to **enabled** and the **Rlogin Environment Information** field to **strip** in the **Global Modes** SMIT configuration menu.

Procedure

To enable the **getty_hty** command, perform the following steps:

1. Edit the **/etc/inittab** file using your favorite editor.
2. Find the hty that you want to enable and change the command field from `getty` to `getty_hty`.
3. In the same line, change the action field from `off` to `respawn`, if necessary.
4. Save and close the file.
5. Run the **telinit q** command to request the **init** command to reread the **/etc/inittab** file.

AIX Programming Interface to the Network Terminal Accelerator

The API allows a host application program access to the user's environment information. The **st.ioctl.h** file includes the **TCGETCINFO** and **TCGETUENV** ioctl operations for the tty driver:

```
#define BIOC        ('B'<<8)
#define TCGETCINFO  (BIOC|20) /* get network connection info    */
#define TCGETUENV   (BIOC|21) /* get network user environment info
*/
```

The information is available from the adapter at any time, so there is no wait to obtain it.

You can use the `ci_family` field to verify the underlying protocol family. Currently, it is always **CIF_INET**. If the `ci_family` field is **CIF_INET**, the `ci_locaddr` and `ci_remaddr` fields are **cinaddr** structures. Other families would use different address structures.

The `ci_service` field identifies the service assigned to the connection. A host application can use this field to determine how to handle the session.

The `ci_flags` field provides some details of the status of the session. The only bit with a defined meaning (although others may be set) is `CIF_RECONNECT`. This identifies a session that held its network connection across a close–and–open sequence. This allows a host application to handle a reconnect differently than a first–time connect. For example, the host might not allow auto–login for an rlogin protocol reconnect.

The `ci_locaddr` and `ci_remaddr` fields provide the local and remote Internet address information for the session. This includes the IP address in the `in_addr` field and the TCP protocol port in the `in_port` field.

The following structures show these fields in detail:

```
/*
*  Structure for TCGETCINFO.
*/

struct cinaddr

  {

  unsigned    long     in_addr;    /*  Internet address         */
  unsigned    short    in_port;    /*  Internet port            */
  unsigned    short    in_res[5];
  };

struct cinfo
        {
      unsigned        short   ci_family; /* protocol family */
      unsigned        short   ci_service; /* type of service */
      unsigned        short   ci_flags; /* various flags */
      unsigned        short   ci_res0[5];
      struct          cinaddr ci_locaddr; /*local address info */
      struct  cinaddr ci_remaddr; /* remote address info */
      unsigned long   ci_res1[4];
      };

/*
 *  Values for ci_family.
 */
#define CIF_INET         1        /* Internet family */
```

```
/*
 * Values for ci_service.
 */

 #define CIS_TELNET      1       /* telnet service ID */
 #define CIS_RLOGIN      2       /* rlogin service ID */
 #define CIS_NOP         3       /* no-op service ID */

/*
*  Fields for ci_flags.
*/

#define CIF_RECONNECT 0x0080 /* Connection held from previous
open  */
```

The **TCGETUENV** ioctl information is available from the adapter a short time after the connection is established. The driver delays completion of the operation until that time. The information returned by this operation is used for administration, authentication, and setting the user's initial environment on the local host.

The definition of the operation is as follows:

```
/*
* Structure for TCGETUENV operation.
* This operation uses a 512-byte buffer that is structured
* as follows:
*
+------+--------+------+-----+------+--------+------+----+-----+
* | type | length | data | ... | type | length | data | ... | ...
|
*
+------+--------+------+-----+------+--------+------+----+-----+
* The length field is the number of bytes in the data field.
* The actual size of the entry might be less than this length.
* For example, the UET_TTYPE entry might have a data length of
* 40 bytes with the data set to ”ansi” and the remaining 36 bytes
* set to zero. A type of value zero is used to terminate the
list.
*/

struct uenv
{
         unsigned        char    ue_type;
         unsigned        char    ue_length;
          char           ue_data[512-2*sizeof(unsigned char)];
};

/*
* Values for ue_type.
*/

 define       UET_WINSZ         1 /* winsize structure          */
 #define      UET_TTYPE         2 /* Terminal type              */
 #define      UET_STDERR        3 /* Standard error port ID     */
 #define      UET_REMUSR        4 /* Remote user name           */
 #define      UET_LOCUSR        5 /* Local user name            */
 #define      UET_ISPEED        6 /* Input terminal baud        */
 #define      UET_OSPEED        7 /* Output terminal baud       */
 #define      UET_XDISPLOC      8 /* X display location         */
```

The host uses the **TCGETUENV** operation to retrieve the environment information for a channel. This operation returns all of the environment information. Each entry has a two–byte header that specifies the environment type in the first byte and the length of the entry in the second byte. This operation allows a host application access to the remote user's environment. The **nop** service never has any environment information. The following are the defined environment types:

UET_WINSIZE Contains the binary **A_WINSIZE** structure in network endian order. This entry is somewhat redundant, since the application gets the window information with the host's **TIOCGWINSZ** ioctl operation to the tty driver. But this is where the adapter holds the information, so it shows up here.

UET_TTYPE Specifies the ASCII terminal type. For the **telnet** command, it is the type negotiated using the terminal–type database. For the **rlogin** command, it is the type received in the initial environment information.

UET_STDERR Specifies the ASCII port number to use for standard error. This is part of the rlogin protocol and is received in the initial environment information, but the **rlogin** command does not use a separate connection for standard error. If this entry exists, an application would typically reject the rlogin protocol connection.

UET_REMUSR Specifies the ASCII name of the remote user. This entry only exists for rlogin protocol and is the name of the remote user making the request.

UET_LOCUSR Specifies the ASCII name of the local user. This entry only exists for the **rlogin** command and is the local user name by which the remote user wants to be known.

UET_ISPEED Specifies the ASCII value of the input baud rate.

UET_OSPEED Specifies the ASCII value of the output baud rate.

UET_XDISPLOC Specifies the ASCII name of the X display location for the remote user's X window system. This entry only exists for the **telnet** command.

Problem Determination

This section provides information for users who encounter problems while operating the Network Terminal Accelerator adapter. Use this section to determine what to do if you run into problems using the adapter.

Firmware Self–Test

Whenever power is first applied to the Network Terminal Accelerator adapter, it initializes and tests its circuitry. The i80960CA processor executes an on–chip self–test, and then loads its initial memory image (IMI) from ROM. After successfully loading the IMI, the i80960CA begins executing a set of self–test routines from ROM–resident firmware. These tests verify that the adapter hardware components listed below are operating properly:

- Internal processor data RAM (0004H through 03FFH)
- Processor register cache
- RAM parity bit
- DRAM
- UART (polled and interrupt modes)
- EPROM
- Real–time clock
- i82596 (Ethernet controller chip) self–test
- Network Terminal Software (NTS) download area dynamic RAM test
- Host handshake detection/execution
- Adapter memory–to–memory direct memory access (DMA) test
- Host handshake test

The Network Terminal Accelerator adapter diagnostics can be run from the System Management Interface Tool (SMIT) and from standalone initial program load (IPL) of the host. The diagnostics verify correct hardware operation of the adapter components and the Ethernet interface.

The following information describes the test units (TUs) used during the different diagnostic operations.

TU 10 (Internal POST)	Invokes the adapter power on/reset self–test by issuing a reset command to the board. The self–test program tests most of the functional blocks on the adapter. The Ethernet transceiver and connector are not checked.
TU 20 (POS/VPD)	Includes two parts: POS register test and VPD test. The POS register test reads the adapter POS registers and validates the contents. The test tests portions of the Micro Channel interface and the host's ability to access the board. The VPD test checks portions of the Micro Channel interface, microprocessor, and main memory.
TU 30 (DMA Loopback)	Checks the data path between the Micro Channel and the RAM on the adapter. This test does not send data out the Ethernet port.
TU 50 (Adapter/Controller Wrap Test)	Checks the entire data path of the adapter, including the Ethernet controller, transceiver, and connector. This test requires the use of Ethernet wrap plug, PN: 70F9625.

SMIT Options

The following list indicates which test units are used for specific problems:

- Concurrent advanced problem determination: TU 20 and TU 30
- Concurrent advanced system verification: TU 20 and TU 30
- Concurrent diagnostic problem determination: TU 20 and TU 30
- Concurrent diagnostic system verification: TU 20 and TU 30

Standalone Options

The following list indicates which test units should be used for specific problems on standalone machines:

- Standalone advanced problem determination: TU 10, TU 20, TU 30, and TU 50
- Standalone advanced system verification: TU 10, TU 20, TU 30, and TU 50
- Standalone diagnostic problem determination: TU 20 and TU 30
- Standalone diagnostic system verification: TU 20 and TU 30

Error Logging and Analysis

The Network Terminal Accelerator adapter presents error information to the host. During system startup, or under control of the adapter diagnostics, the host can monitor the progress and results of the adapter on–board self–test. The driver should log any self–test failures as fatal. When it logs a self–test failure, the driver should log the self–test checkpoint or error code reported by the adapter. The self–test error codes and checkpoints are listed, on page 9-32 and, on page 9-35, respectively.

During the host handshake portion of the self–test, the driver is responsible for detection of some errors, as well as for logging them. In these cases, the driver should log additional data related to the error (for example, expected versus read data values).

During normal operation, the adapter reports error conditions through the DPRAM interface along with normal control and status information. These error codes are detailed, on page 9-32. The driver should log any such errors as fatal.

The following hardware failure error messages are logged by the adapter driver modules upon detection of any of the errors described above. ”Error Codes”, on page 9-32 describes in detail the various status and error codes logged with each error. In all cases, replacement of the failing adapter is the recommended corrective action.

rhpdd Error Messages

The following list details error messages from the **rhpdd** device driver indicating power–on self–test failures:

Error Message	Comment
`Self-test error code 0x`*HH*	*HH* is a self–test error code.
`Self-test timed out, code 0x`*HH*	*HH* is a self–test checkpoint.
`Self-test: handshake timed out, got 0x`*HH*	*HH* is the last data read during the handshake test.
`Self-test: unexpected interrupt`	Driver received a spurious interrupt from the board during the handshake test.
`Self-test: ISR ready error, flag 0x`*HH*	Adapter did not indicate `ready` during the interrupt portion of the host handshake test.
`Self-test: ISR test error. exp 0x`*HH*`, got 0x`*HH*	Data compare error during interrupt–driven handshake test.

Error Message	Comment
`Self-test: interrupt not received`	Driver received no interrupt during handshake test.
`Self-test: completion timed out, code` *HH*	*HH* is a self–test checkpoint.

The following list details error messages from the **rhpdd** device driver indicating hardware failures detected during operation:

Error Messages	Comments
`invalid dp_obuf count=`*N*	Byte count presented in the adapter's `dp_ocnt` field is out of range.
`code/message= 0x`*HH*`/0x`*HHHHHHHH*	Indicates that the adapter has crashed, and gives the error code and message present at the time of the crash.

FRU List

There are two models of the adapter. They differ only in the amount of main memory installed on the board. The following list gives the field–replaceable unit (FRU) part number corresponding to each model.

Model	FRU Part Number
2MB (256 users)	51G8538
8MB (2048 users)	51G8539

Error Codes

The following list details the error codes that the adapter board might log if the POST fails (at power–up or during the execution of TU 20). The codes are grouped according to type.

These error codes indicate a hardware malfunction of some kind, generally requiring adapter repair or replacement.

Internal Data RAM Test

80	Internal data RAM locations 0004H to 03FFH were written with each location's 32–bit address; when read back, the written addresses did not match the true addresses.
81	Internal data RAM locations 0004H to 03FFH were written with either the data pattern 55555555H or AAAAAAAAH; when checked, the pattern was not correct.
82	Internal data RAM locations 0004H to 03FFH were written with 00; when read back, the written data did not equal 00.

Register Cache Test

84	During the testing of the internal register cache RAM, one of the values read back from the cache RAM did not match the value written.

UART Test

89	The universal asynchronous receiver/transmitter (UART) status register contains an incorrect value.
8A	Data is looped through the UART in polled mode; the byte read does not match the byte written.

8B	A parity, overrun, framing, or break error occurred during the polled loop.
8C	One character was sent to the transmitter, but more than one was received in the polled loop.

Interrupt UART Test

8D	After waiting 500 ms, all the characters have not looped through.
8E	The received data does not match the transmitted data.
8F	An unexpected UART interrupt occurred.

Clear and Check DRAM Parity

90	A DRAM parity error was detected.

PROM Checksum Test

94	The calculated programmable read–only memory (PROM) checksum did not match the precalculated checksum.

RAM Size Test

98	The calculated DRAM size does not match the valid sizes allowed for this product.

Real–Time Clock Period Test

9C	A real–time clock tick did not occur within 500 ms.
9D	The 52 ms real–time clock period is not within a 5% tolerance (49.4 to 54.6 ms).

Memory Region Byte Order Test

A1	The value read back from memory region 1 did not match the value written.
A2	The value read back from memory region 2 did not match the value written.
A3	The value read back from memory region 3 did not match the value written.

NINDY RAM Test

A4	One of the following patterns read back from the upper 32KB of DRAM did not match the value written: 55555555H, AAAAAAAAH, 5A5A5A5AH, or A5A5A5A5H.
A5	The 32–bit value read from one of the upper 32KB of DRAM did not match the value written into it. The value written was the address of the 32–bit location.
A6	The 32–bit value read from one of the upper 32KB of DRAM did not match the value written into it. The value should have been 0.

Destructive DRAM Test

B0 After writing one of the following patterns, the read value did not match the written value:

- 8–bit values, 55H, AAH, A5H, 5AH
- 16–bit values 5555H, AAAAH, A5A5H, and 5A5AH
- 32–bit values 5555555H, AAAAAAAH, A5A5A5A5H, and 5A5A5A5AH
- Quad–word values 08040210H, 80402010H, F7FDFBFEH, and 7FDFBFEFH.

B1 The 32–bit value read from a DRAM location did not match the value written. The value written was the address of the 32–bit location.

B2 The value read from a DRAM location did not match the value written. The value written was 0.

Modified Algorithm Test Sequence DRAM Test

B3 The value read from a DRAM location did not match the value written.

Ethernet Controller Internal Self–Test

B4 The Ethernet controller self–test failed.

Ethernet Controller Reset and Initialization

B5 The Ethernet controller failed to reset properly, or the status of the channel attention line is incorrect.

Ethernet Controller Interrupt Test

B6 The Ethernet controller failed to generate an interrupt.

Set Ethernet Controller Address

B7 An attempt to set the Ethernet address in the Ethernet controller failed.

Ethernet Controller Internal Loopback

B8 The Ethernet controller internal data loopback test failed.

Ethernet Controller OnBoard Loopback

B9 The Ethernet controller onboard loopback failed.

Ethernet Controller External Loopback

BA The Ethernet controller external loopback failed.

Ethernet Controller Internal Register Dump Test

BB The Ethernet controller failed to dump its internal registers.

Memory–to–Memory DMA Test

BC Testing of the 80960CA's direct memory access (DMA) channel 1 failed.

BD An unexpected DMA interrupt occurred during DMA channel 1 testing.

Host Handshake (Polled)

C0	The host did not echo data back to the adapter within 250 ms.

Host Handshake (Interrupt)

C1	The adapter did not receive an interrupt from the host within 12 seconds to indicate that data had been received.
C2	The adapter received two interrupts from the host when only one was expected.
C4	Data received by the adapter did not match what was sent to the host.
C5	The host did not respond to the adapter's interrupt within 12 seconds.
C6	The processor received an interrupt from a source other than the host during the host handshake test.
C8	Unexpected DMAP status register value.
C9	The adapter timed out waiting for the DMAP to the host to complete.
CA	The DMAP data read back from the i82325 data register did not match that written.
CB	Unexpected direct memory access controller (DMAC) status register value.
CC	The adapter timed out waiting for the DMAC operation to complete.
E0	Electrically eraseable programmable read only memory (EEPROM) internal CRC mismatch.
E1	Could not successfully initialize EEPROM.

Checkpoints

The following list describes the checkpoints that reference self–tests performed by the adapter board along with their corresponding self–tests:

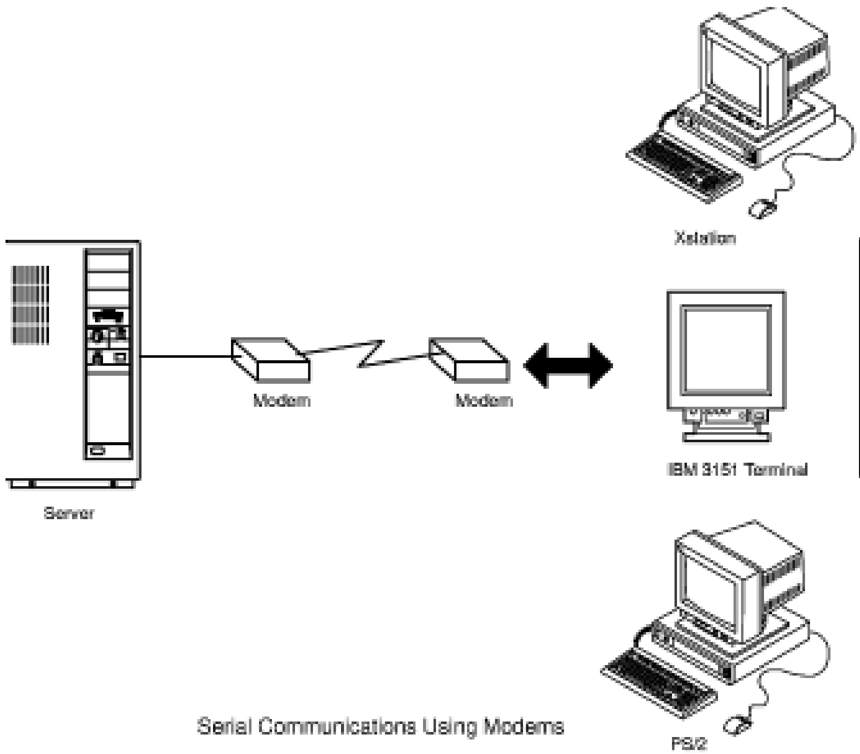
01	Internal data RAM test
02	Register cache test
03	i82325 initialization and test
04	Memory region byte–order test
06	Clear and check DRAM parity
07	PROM checksum test
05	DRAM size test
08	NINDY RAM test
09	Real–time clock period test
0A	Polled UART test
0B	Interrupt UART test
0C	Dynamic RAM test (destructive)
0D	Ethernet controller internal self–test
0E	Ethernet controller reset and initialization
0F	Ethernet controller interrupt test
10	Ethernet controller address
11	Ethernet controller internal loopback
12	Ethernet controller onboard loopback
13	Ethernet controller external loopback

14	Ethernet controller internal register dump
15	Memory–to–memory DMA test
40	Adapter waiting for the host to execute its portion of the host handshake test.
1E	Polled handshake test
1F	Interrupt host handshake test
20	Programmed I/O to host (i82325 DMAP operation)
21	DMA block of data to or from host (i82325 DMAC operation)
22	EEPROM test

Chapter 10. Modems Overview

Modems provide serial communications across ordinary telephone lines. This chapter discusses modem standards, general modem setup, and specific configuration recommendations when attaching a modem to the server family of asynchronous adapters.

A modem is a device that can connect one computer to another across ordinary telephone lines. The current telephone system is incapable of carrying the voltage changes required for a direct digital connection. A modem overcomes this limitation by modulating digital information into audio tones for transmission across the phone line and by demodulating those tones back into digital information upon reception. It is from these actions that the name MODEM is derived (MOdulate, DEModulate).

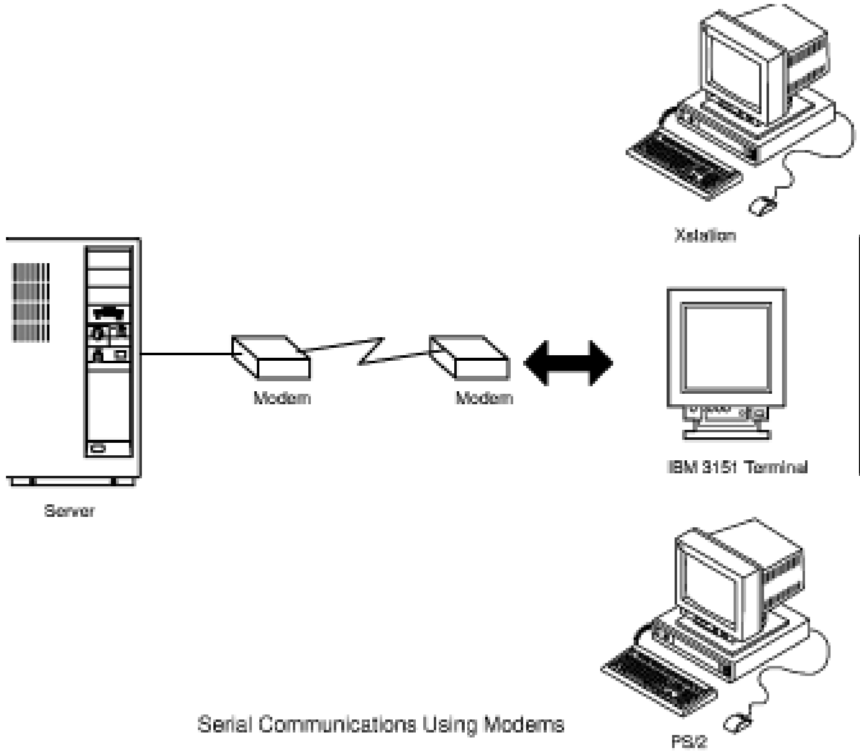


Serial Communications Using Modems

Modern modems generally communicate at speeds ranging from 2400 baud to 14,400 baud. Some modems use proprietary means to communicate at even faster rates (for example 19,200 baud). The disadvantage to these proprietary, faster modems is that in order to achieve faster speeds, identical modems (or more precisely, modems implementing the same proprietary protocol) must be used at both ends. Older modems communicate at speeds ranging from 300 baud to 1200 baud.

Often, the term baud is used to refer to a modem's speed instead of bps. Baud is actually a measurement of the modulation rate. In older modems, there was only 1 bit encoded in each signal change, so a modem's baud rate was equal to the modem's speed. Modems that operate at higher speeds, however, still generally operate at 2400 (or even 1200) baud, and encode two or more bits per signal change. A modem's bps rate is calculated by multiplying the number of data bits per signal with the baud (for example, 2400 baud x 6 bits per signal change = 14,400 bps). Most modern modems can communicate at a variety of speeds (for example: 14400, 9600, 7800, 4800, and 2400 baud). In general, the terms bps and baud are interchangeable.

Figure showing serial communications using modems:



Serial Communications Using Modems

The following sections contain detailed information about modems:

- Telecommunications Standard, on page 10-3
- Modem Considerations for AIX, on page 10-5
- Considerations for the TTY Device, on page 10-8
- Modem Attachment and Configuration, on page 10-11
- Troubleshooting Modem Problems, on page 10-17

Telecommunications Standards

The older speeds of 300, 1200, and 2400 baud were well–defined. However, as modem manufacturers began to devise methods for gaining higher speeds, each modem manufacturer started to use a proprietary method or algorithm, incompatible with modems from other manufacturers.

Section topics included are:

- Full and Half Duplex Transmission
- CCITT Communication Standard
- Microcom Networking Protocol
- MNP Communication Standards

Full and Half Duplex Transmissions

When studying telecommunications standards, it is important to understand the differences between half duplex and full duplex transmissions. In a *half duplex* (HDX) transmission, a data packet is sent by one system and received by the other. Another data packet CANNOT be sent until the receiving system sends an acknowledgement back to the sender.

In a *full duplex* (FDX) transmission, both the sending and receiving systems communicate with each other simultaneously; in other words, both modems can send and receive data at the same time. This means a modem can be receiving a data packet while acknowledging the receipt of another.

CCITT Communication Standard

Today, the United Nations Consultative Committee for International Telephony and Telegraphy (CCITT) defines standards for most high–speed communications.

Even modern high–speed modems are much slower than other methods of computer communication. A high–speed modem typically operates at 14,400 baud, while an Ethernet LAN connection operates at 10,000,000 bps. In order to boost data throughput, high–speed modems typically offer one or more data compression algorithms. These algorithms can boost the throughput of a high–speed modem to speeds above 54,000 bps.

Note that these algorithms are sensitive to the data being transmitted. If the data has already been compressed (for example, with the **compress** command), the data compression methods of high–speed modems will offer little or no benefit, and might even reduce data throughput.

When using a modem with data compression technology, the speed of the DTE/DCE connection between the computer and the modem should be greater than the nominal data rate of the connection between modems. For example, with a V.32*bis* modem with V.42*bis* data compression, the data rate of the modem (the speed at which the modem communicates across telephone lines) is 14,400 baud. When the V.42*bis* compression is active, actual data throughput can reach 57,600 bps. To accommodate the greater throughput offered by data compression, the speed of the DTE/DCE between the computer and the modem should be set to 57,600 bps (38.4Kbps).

Some modems implementing data compressions and modem modulation schemes may yield a higher data throughput than some systems and asynchronous adapters can accommodate.

Today, the CCITT defines standards for high–speed communications including data compression algorithms. CCITT standards are usually named V.*nn*, where *nn* is a number.

Following is a list of some common communications standards defined by the CCITT:

V.29 CCITT standard for half–duplex 9600 baud communications.

V.32 CCITT standard for full–duplex 9600 baud communications.

V.32*bis* CCITT standard for 14,400 communications. V.32*bis* is a revision to the V.32 standard.

V.fast Proposed CCITT standard for 28,800 baud communications. Note that this standard will achieve 28,800 bps data rates through multiple bit encoding, instead of the data compression scheme used by MNP Class 9.

V.42 CCITT data compression standard.

V.42*bis* Revised CCITT data compression standard.

Microcom Networking Protocol

Another de–facto standard is the Microcom Networking Protocol (MNP) which was originally developed by Microcom, Inc. Available in versions (called classes) 1–9, MNP is a high–performance, high–speed protocol that was available before the advent of the CCITT standards. With MNP, errors in the transmitted data packets are detected by the remote modem causing it to request a retransmission of the data packet in error. The capability to recognize and quickly correct data errors makes MNP one of today's most common protocols.

MNP Communication Standards

MNP Class 1 An asynchronous, half–duplex, byte–oriented method of transferring data realizing about 70% efficiency. This standard is uncommon in modern modems.

MNP Class 2 A full–duplex counterpart to MNP Class 1 which is also uncommon in modern modems.

MNP Class 3 A synchronous, bit–oriented full–duplex method of transferring data realizing about 108% efficiency. Efficiency greater than 100% is realized because the start/stop bits required for an asynchronous connection are eliminated. The DTE/DCE between the modem and the system are still asynchronous.

MNP Class 4 An enhancement to MNP Class 3 including a mechanism for varying the packet size (adaptive packet assembly) and a means of eliminating redundant administrative overhead (data phase optimization). An MNP Class 4 modem offers approximately 120% efficiency.

MNP Class 5 Includes data compression along with Class 4 features. An MNP Class 5 modem offers 200% efficiency.

MNP Class 6 Allows incorporation of multiple, incompatible modulation techniques into one modem (universal link negotiation). This allows MNP Class 6 modems to begin communication at a slower speed and negotiate a transition to a higher speed. Class 6 also includes a statistical duplexing scheme which dynamically allocates utilization of half–duplex modulation to simulate full–duplex service. All features of MNP Class 5 are supported.

MNP Class 7 Incorporates enhanced data compression. Combined with Class 4, efficiencies of 300% can be realized.

MNP Class 8 Not Applicable.

MNP Class 9 Combine enhanced data compression with V.32 technology to allow data rates up to 28,800 bps.

Modem Considerations for AIX

The configuration of a modem attached to AIX is different than that of a personal computer (or PC). This section discusses modem interface requirements for the general user.

Section topics include the following:

- Supported Modems
- Data Carrier Detect Handling
- Data Terminal Equipment or Data Communication Equipment Speeds
- Modem Control Signals

Supported Modems

Any modem that is EIA 232 compliant, capable of returning results in response to a command, and capable of communication at one of the following baud rates can be attached to AIX:

```
  50 134  300 1800  9600
  75 150  600 2400 19200
 110 200 1200 4800 38400
```

Data Carrier Detect Handling

The server uses the Data Carrier Detect (DCD) signal to monitor the true state of a modem. If the DCD signal on the modem's port is "high," the server believes the modem to be in use. It is therefore important to know which circumstances cause this signal to be forced into a "high" state. The DCD signal can be raised high for the following reasons:

- The use of **clocal** in the stty attributes for runtime field on the SMIT **TTY Configuration** panel.
- Having the Ignore Carrier Detect field set to **enable** on the SMIT **TTY Configuration** panel for ttys connected to a 128–port adapter.
- The modem forces DCD high with either AT commands or switches.
- The tty port is already in use by an application.

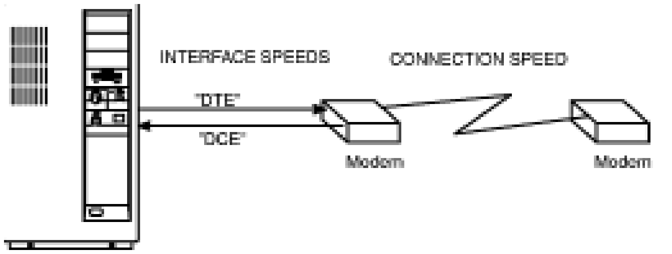
Note: When modems make a connection with another modem, the modem raises the CD. Most modem defaults settings set this signal "high" at all times even when the modem is idle. CD should not be forced "high."

Data Terminating Equipment or Data Circuit–Terminating Equipment Speeds

Data Terminating Equipment (DTE) and Data Communication Equipment (DCE) are used to describe two different hardware groups. The term DTE is used primarily for those devices that display user information. It also includes any devices that store or generate data for the user. The system units, terminals, and printers all fall into the DTE category.

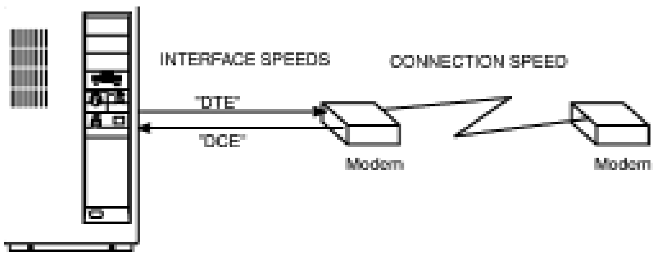
DCE includes any device which can be used to gain access to a system over telecommunication lines. The most common forms of DCEs are modems and multiplexers.

With AIX serial communications involving modems, there are three major considerations:



Modem Speed Considerations

- DTE interface speed (server to modem). This is the speed the server communicates to the modem.
- DCE interface speed (modem to server) sometimes called the ”serial port interface speed.” This is the speed at which the modem communicates to the server.
- Connection speed (modem to modem). This is the speed at which a modem communicates (or talks) to another modem. See illustration.



Modem Speed Considerations

Most modern, high–speed modems allow the DCE interface speed to be different than the connection speed. This allows the DTE speed to be locked at a single baud rate while allowing the connection speed to fluctuate, up or down as needed, for proper communication between modems.

Modern high–speed modems hold the data to be transmitted to the server in a buffer and send it when the system can accept it. They can also hold data to be transmitted to the other modem in a buffer and send it as the remote is able to accept it. This kind of data transmission requires the modem and the server to engage in *flow control*.

Modem Control Signals

Modems are often used to initiate and receive calls. It is therefore important to program the modem to negotiate a connection at the highest possible speed and to reset itself to a known state after a connection is stopped. The server will toggle the Data Terminal Ready (DTR) signal from on to off to instruct the modem to terminate the connection. Most modems can be configured to reset themselves when this on–to–off DTR transition occurs.

Note: The tty can be configured to not drop DTR by unsetting the `hupcl` flag in the stty run–time attributes.

For the connection between the server and the modem to be fully functional, the cabling must have the following qualifications:

- It must meet specifications.
- It should be properly shielded.
- The following signals should be provided: RxD, TxD, RTS, CTS, SG, DCD, and DTR.

  Note: The 16–port asynchronous adapter does not provide support for the RTS and CTS signals. It is therefore impossible to use RTS/CTS hardware flow control with this adapter.

If binary data is to be transferred using a modem on this adapter, a file transfer protocol that detects incorrect data and resends the missing data (for example, Xmodem, zmodem, Kermit, and UUCP) should be used.

The following describes the signals used by the server:

Signal	Description
FG	Frame Ground. Pin 1 of the EIA 232D specification that provides for a cable shield. Properly used, the signal is attached at pin 1 on one side of the cable only and is connected to a metal sheath around the cable.
TxD	Transmit Data. Pin 2 of the EIA 232D specification. Data is transmitted on this signal. Controlled by the server.
RxD	Receive Data. Pin 3 of the EIA 232D specification. Data is received on this signal, controlled by the modem, which is sent by the modem.
RTS	Request To Send. Pin 4 of the EIA 232D specification. Used when RTS/CTS flow control is enabled. This signal is brought high when the system is ready to send data and dropped when the system wants the modem to stop sending data.
CTS	Clear To Send. Pin 5 of the EIA 232D specification. Used when RTS/CTS flow control is enabled. This signal will be brought high when the modem is ready to send or receive data. It will be dropped when the modem wishes the server to stop sending data. Controlled by the modem.
DSR	Data Set Ready. Pin 6 of the EIA 232D specification. Signals the server that the modem is in a state where it is ready for use. Controlled by the modem.
SG	Signal Ground. Pin 7 of the EIA 232D specification. This signal provides a reference voltage for the other signals.
DCD	Data Carrier Detect. Pin 8 of the EIA 232D specification. This provides a signal to the server that the modem is connected with another modem. When this signal is brought high, programs running on the server will be able to open the port. Controlled by the modem.
DTR	Data Terminal Ready. Pin 20 of the EIA 232D specification. This provides a signal to the modem that the server is on and ready to accept a connection. This signal is dropped when the server wishes the modem to drop connection to another modem. It is brought high when the port is being opened. Controlled by the server.
RI	Ring Indicate. Pin 22 of the EIA 232D specification. This provides a signal to the server that the modem is receiving a call. It is seldom used and is not needed for common operations. Controlled by the modem.

Considerations for the TTY Device

Each tty port must be set up before attaching any input/output device. Before a tty device is created, several factors concerning its functionality should be determined. This section discusses those factors and their importance. Topics covered in this section are the:

- Physical port
- Baud rate
- Bits–per–character, parity, and stop bits
- Port startup and control
- Terminal emulation

Physical Port

Requirements for the physical port to which a modem is connected are dependent on the server and its serial adapters. The system and adapter must support (at least) the following modem control signals to be fully functional on all asynchronous software applications: RxD, TxD, RTS, CTS, SG, DCD, and DTR.

A server typically has two built–in serial ports. Additional multiport adapter cards or terminal servers can be installed on the server to allow the configuration of more serial devices. The following optional adapter cards and base units are supported on the server:

- 8–port adapter
- 16–port adapter
- 64–port adapter
- 128–port adapter
- IBM 7318 Terminal Server

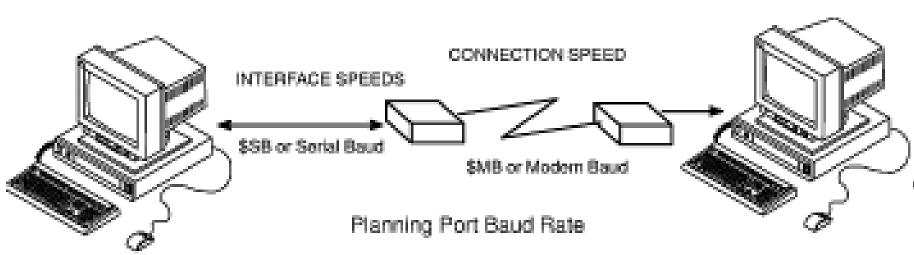
The above adapters and terminal server support the required modem control signals except the 16–port adapter. The 16–port asynchronous adapter does not provide support for the RTS and CTS signals. It is therefore impossible to use hardware flow control with this adapter.

If a binary file is to be transferred using a modem on the 16–port adapter, a transfer protocol that detects incorrect data and resends the missing data (for example, Xmodem, Zmodem, Kermit, and UUCP) should be used.

Review the associated chapter for each of the previously mentioned serial adapters for more information.

Baud Rate

There are several important factors to take into consideration when planning the port baud rate:



Planning Port Baud Rate

- Will the remote system be forced to connect at the local system's communication speed?

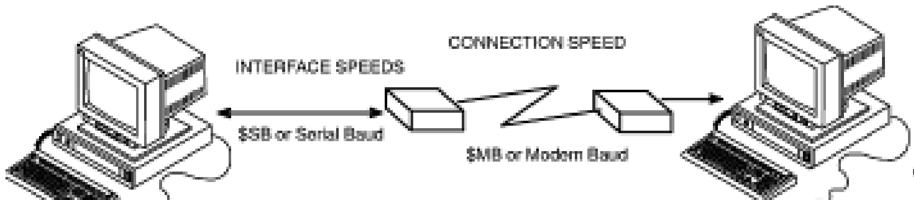
If the remote system is going to be forced to connect at a single speed, that speed should be specified in the BAUD rate field of the SMIT **TTY Setup** menu of the local system. The modem must be set to connect at that same speed.

- Will the local system adapt to the remote system's communication speed?

  The MultiTech MT932 MultiModem II has programmable variables controlling the two separate communication speeds:

  ```
  $SB=9600 $MB=38400
  ```

  See illustration Planning Port Baud Rate.



Planning Port Baud Rate

The `$SB` value is the serial baud rate between the modem and the server. The `$MB` value is the modem baud rate between the two modems themselves. With the above settings, the modem would communicate to the server at 9600 bps while at the same time communicating with another modem at 38400 bps.

Note: To avoid incompatibility problems between modems due to nonstandardized implementation of high–speed protocols, it is advisable to use the same make and model of modems at both ends of the connection.

Bits–Per–Character, Parity, and Stop–Bits

The parity, bits–per–character, and stop bits for a serial device are interrelated and require a basic knowledge of serial communications. Most modern serial communications devices (modems, terminals, etc.) use a 10–bit transmission character:

- 1 bit for the start bit
- 1 or 2 bits for the stop bits
- 0 or 1 bits for the parity setting
- 7 or 8 bits for the data character.

The following table shows all the valid combinations.

Bits–per–character	Start	Stop	Parity
8	1	1	none
7	1	1	1
7	1	2	0

Startup and Control

Local/Remote Attachment

When a port is used for bidirectional communication (both dial–in and dial–out or connect–in and connect–out), it should be set up to enable a login on the port while still allowing other users to dial–out. This bidirectional capability can be accomplished by using the System Management Interface Tool (SMIT) to change the tty's Enable LOGIN field to either `share` or `delay`.

Connection Over a Modem (Enable LOGIN=share)

With the tty's Enable LOGIN field set to `share`, the **/etc/getty** command, which controls the login process, will start up on the port and wait for the modem to turn on a carrier signal.

When the modem turns the carrier signal on, the **getty** command attempts to lock the port so that no other processes can use it. If another program (for example, **cu** or **ate**) has already locked the port for dial out, the **getty** command waits until the port is freed by that process before attempting to open it again. If the port is not locked, the **getty** command locks it and sends out a login herald, thus allowing a login to take place on the port.

Direct Connect (Enable LOGIN=delay)

With the tty's Enable LOGIN field set to `delay`, **/etc/getty** waits until the remote system sends a character to the local tty before locking the port and sending a login herald.

Note: The `share` and `delay` settings lock the port as soon as the **getty** process starts. This is because the carrier signal is already up on a direct tty connection.

When LOGIN is set to `delay`, the **getty** command monitors the input buffer of the terminal device for a character. When a character is found, it attempts to lock the port, exactly like the shared port.

Terminal Emulation

A variety of applications and system functions are tailored to specific terminal types such as the DEC VT100, IBM 3151, or Wyse 50. Each terminal type requires its own special emulation to function properly under AIX. Without proper emulation settings, terminals may display incorrect characters or change the functions of specific keys on the terminal keyboard.

Terminal emulation can be set from the AIX command line by using the TERM= environment variable (for example, TERM=ibm3151 <enter>) or in SMIT by altering the TERMINAL type variable for a tty.

Note: For most modems used for both dial–in and dial–out connections, a TERMINAL type of `dumb` is sufficient. This value allows users dialing in to set their own terminal type (using TERM=) after logging in. If logins occur more frequently from a single, known terminal type, users can setup the TERMINAL type field when defining the tty from SMIT.

The supported terminal types are located in the **/usr/lib/terminfo** directory and subdirectories.

Modem Attachment and Configuration

The following section contains information necessary to attach and configure a modem to the native ports, 8–, 64–, and 128–port adapters on AIX.

Topics covered in this section include:

- Creating a TTY device on the server
- Modem connection
- Programming the modem
- Testing the modem

Prerequisites

- The user must have root authority.
- The serial adapter card and port must be available for use by the system.
- Proper modem cabling for serial connection must be used.

Creating a TTY Device on AIX for Modem Attachment

Use the System Management Interface Tool (SMIT) to define a tty port for the device attachment. Most fields are for the general device type. The only field that can affect the modem is the Enable LOGIN field with the following values:

DISABLE	No getty process is run on the port. Use this setting for dial–out only modem ports.
ENABLE	A getty process is run on the port. Use this setting for dial–in modems only.
SHARE	A getty process is run on the port, but the getty process allows programs to dial in and out of this port without manually changing to disable or enable. Use this setting for bidirectional port usage.
DELAY	A getty is run on the port in bi–directional mode, but no herald is sent until the getty process receives a keystroke from the user.

Fields specific to the 128–port asynchronous adapter:

Force Carrier or Ignore Carrier Detect	disable
Perform Cooked Processing in Adapter	disable
(AIX Version 3 only)	
Use Alternate RJ–45 Pinouts*	disable

Note: * This setting is set to disabled if the 10 pin RJ–45 connector is used. This setting should be enabled if the 8 pin RJ–45 connector is used.

Attaching the Modem to the Port

The next step is to physically attach the modem to the serial adapter port using appropriate cabling. If possible, avoid placing the modem and cabling near any high line–noise sources such as florescent lighting, light switches, or uninterruptable power supply (UPS) devices. Sudden, or continuous power fluctuations can cause problems with tty ports (see troubleshooting section for more information).

Programming the Modem

Most modems require customization of their default settings in order to function properly on the system unit. The commands and settings discussed in this section configure a Hayes compatible modem with the basic parameters needed for operation on the server's serial ports.

There are two methods of configuration commonly used. They are:

- Using the **cu** command.
- Using a C program.

Configuring the Modem with the cu Command

The **cu** command is the interactive and convenient way to program a modem on the server. This command is in the **/usr/bin** directory and is a part of **bosext1.uucp.obj** licensed program product.

Prerequisites:

- The Basic Networking Utilities (BNU) should be installed on the system. These utilities include the **cu** command required for configuration.
- The modem must be attached to the serial port with the correct cabling.
- The modem must be switched on.
- Some knowledge of modem configuration is assumed.
- Modem reference manual must be available for problem determination.

Procedure

1. Add the following line to the **/usr/lib/uucp/Devices** file verifying that the letter D (Direct) is in the leftmost position in the file and that the # in **tty#** is replaced with the tty device number of the modem:

   ```
   Direct tty# – Any direct
   Direct tty43 – Any direct
   ```

   Note: Any lines in this file that start with a # (pound sign) in the leftmost position are comments only.

2. Verify that the tty is disabled. Enter: `pdisable tty#`.
3. Enter: `cu –ml tty#`.
4. The message `Connected` is displayed.
5. Verify that you have the attention of the modem. Enter:

   ```
   AT
   ```

   The modem should respond with OK. If it does not respond to your command, refer to troubleshooting.

   – Some modems are case sensitive. Use uppercase letters for the **AT** commands.

   – In normal operation, it is preferable for the modem to reset when the DTR is dropped (&D3 setting). When the modem is being set up for the first time, however, it is advisable not to have the modem reset if DTR is dropped (&D2 setting). If the modem resets itself, all programmed settings which are not saved into the modem's memory will be lost.

     Not having the modem reset also protects changes when &C1 is set, because changing the Carrier Detect status may cause the carrier detect line to toggle on some modems, causing the **cu** command to drop the line. You may wish to set the modem to &D3 after the final setup is made.

   – Although these commands are standard for most Hayes–compatible modems, there is no guarantee that they are standard for your modem. Compare the commands with your modem manual before proceeding.

6. Enter the following commands. The modem should respond with OK after each line is entered.

```
AT&F    <enter>
ATE1    <enter>
AT&D2   <enter>
AT&C1   <enter>
ATS0=1  <enter>
ATS9=12 <enter>
AT&W    <enter>
~.      <enter>  (terminate connection)
```

AT Command Descriptions

`AT&F`	Recall the factory configuration as the active configuration.
`ATE1`	In command state, echo characters from the keyboard to the screen. (Make sure carrier is not ON on the port or modem.)
`AT&D2`	Monitor the DTR signal. When an on–to–off transition of the DTR signal occurs, the modem hangs up and enters the command state.
`AT&D3`	Monitor the DTR signal. When an on–to–off transition of the DTR signal occurs, the modem hangs up and resets.
`AT&W`	Write the storable parameters of the current configuration to memory.
`AT&C1`	Track the status of the carrier detect signal.
`ATS0=1`	Autoanswer.
`ATS9=12`	Carrier–detect response time. Default is 6. Possible values are 1 to 255, in tenths of seconds.

7. Enter ONE of the following commands on the AIX command line substituting the tty device number for *n*.

```
penable  ttyn
pshare   ttyn
pdelay   ttyn
pdisable ttyn
```

The modem is now configured with the basic commands needed for most AIX serial communication needs.

Configuring the Modem with a C Program

Prerequisites:

- A C Language compiler must be installed on the system.
- An ASCII text editor must be installed on the system (for example, /**bin**/**vi**).
- The modem must be attached to serial port with the correct cabling.
- The modem must be switched on.
- Some knowledge of modem configuration is assumed.
- Some knowledge of C programming is assumed.
- Modem reference manual must be available for problem determination.

Procedure

1. Using a text editor such as /**bin**/**vi**, enter the following C code: for example, **vi motalk.c**.

```
/************************************************************/
/* Motalk - A ”C” program for modem setup.                  */
/*                                                          */
/* NOTE: This program is supplied as an example only and    */
/* is not officially supported by IBM.                      */
/*                                                          */
/*                                                          */
```

```
/* To create: vi motalk.c <enter>                                    */
/* Usage: motalk /dev/tty? [speed]                                   */
/*                                                                   */
*/*****************************************************************/
#include <errno.h>
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <termio.h>
FILE *fdr, *fdw;
int fd;
struct termio term_save, stdin_save;
void Exit(int sig)
{
   if (fdr) fclose(fdr);
   if (fdw) fclose(fdw);
   ioctl(fd, TCSETA, &term_save);
   close(fd);
   ioctl(fileno(stdin), TCSETA, &stdin_save);
   exit(sig);
}
main(int argc, char *argv[])
{
   char *b, buffer[80];
   int baud = 0, num;
   struct termio term, tstdin;
   if (argc < 2 || !strcmp(argv[1], "-?"))
   {
      fprintf(stderr, "Usage: motalk /dev/tty? [speed]\n");
      exit(1);
   }
   if ((fd = open(argv[1], O_RDWR|O_NDELAY)) < 0 )
   {
    perror(argv[1]);
    exit(errno);
  }
    if (argc > 2)
  {
        switch(atoi(argv[2]))
        {
          case   300: baud = B300;
                      break;
          case  1200: baud = B1200;
                      break;
          case  2400: baud = B2400;
                      break;
          case  4800: baud = B4800;
                      break;
          case  9600: baud = B9600;
                      break;
          case 19200: baud = B19200;
                      break;
          case 38400: baud = B38400;
                      break;
          default:    baud = 0;
                      fprintf(stderr, "%s: %s is an unsupported
baud\n", argv[0], argv[2]);
                      exit(1);
          }
      }
/* Save stdin and tty state and trap some signals */
ioctl(fd, TCGETA, &term_save);
ioctl(fileno(stdin), TCGETA, &stdin_save);
signal(SIGHUP, Exit);
```

```
signal(SIGINT, Exit);
signal(SIGQUIT, Exit);
signal(SIGTERM, Exit);
/* Set stdin to raw mode, no echo */
ioctl(fileno(stdin), TCGETA, &tstdin);
tstdin.c_iflag = 0;
tstdin.c_lflag &= ~(ICANNON | ECHO);
tstdin.c_cc[VMIN] = 0;
tstdin.c_cc[VTIME] = 0;
ioctl(fileno(stdin), TCSETA, &tstdin);
/* Set tty state */
ioctl(fd, TCGETA, &term);
term.c_cflag |= CLOCAL|HUPCL;
if (baud > 0)
{
   term.c_cflag &= ~CBAUD;
   term.c_cflag |= baud;
}
term.c_lflag &= ~ECHO;
term.c_cc[VMIN] = 0;
term.c_cc[VTIME] = 10;
ioctl(fd, TCSETA, &term);
fcntl(fd, F_SETFL, fcntl(fd, F_GETFL, 0) & ~O_NDELAY);
/* Open tty for read and write */
if ((fdr = fopen(argv[1], "r")) == NULL )
{
   perror(argv[1]);
   exit(errno);
}
if ((fdw = fopen(argv[1], "w")) == NULL )
{
   perror(argv[1]);
   exit(errno);
}
/* Talk to the modem */
puts("Ready... Press <ctrl> C to exit");
while (1)
{
   if ((num = read(fileno(stdin), buffer, 80)) > 0)
                       write(fileno(fdw), buffer, num);
   if ((num = read(fileno(fdr), buffer, 80)) > 0)
                       write(fileno(stdout), buffer, num);
}
Exit(0);
}
/* ************************************************************ */
/* ******************* END OF PROGRAM ******************* */
/* ************************************************************ */
```

2. Compile the above program using the system's C compiler.

   ```
   cc -o motalk motalk.c
   ```

3. Run the program using the following syntax:

   ```
   motalk /dev/ttyn #
   ```

   Where *n* is the number of the tty port and *#* is the speed of the tty port. For example, to set port 43 for 9600 bps, the syntax would be:

   ```
   motalk /dev/tty43 9600
   ```

4. The program should run and display `Ready...press <ctrl> C to exit`. Use step 6 from Method 1 to program the modem. Press Ctrl–C to stop the program when finished.

Automated Modem Configuration

To program a modem automatically with a shell script, use the UUCP **cu** command. Create and run the following script:

```
#!/bin/ksh
tty=$1
shift
speed=$1
shift
{
       while [ –n ”$1” ];do
       echo ”$1/r”
       sleep 2
       shift
       done
       echo ’~.’
{| cu –ml $tty –s $speed
```

Troubleshooting Modem Problems

A convenient way to debug any modem problem is to remove the modem and attach an ASCII terminal (with an interposer or null modem) to the same port and cabling as the modem. Set up the terminal with the same line speed, bits per character, and parity as the modem. A login herald should be displayed on the screen. If the herald is displayed on the terminal screen, then the problem is quickly isolated to the modem configuration.

The following tips will help you isolate problems associated with modem connections:

Problem	Resolution
Respawning too rapidly	The **getty** program is respawned by **init**.
Messages on console or errpt	If **init** sees that it has to respawn any program more that five times in 225 seconds, it will display the message on the console and not respawn it for a period of time. The solution is to find out why **getty** is dying. There may be several causes: • Incorrect modem settings, usually as a result of having CD strapped high on the modem or cabling and also having either ”echo” or ”command response” turned on. (CD can also be assumed high by adding clocal in the runmodes and/or logmodes in the port configuration or also forced on the 128 port.) • Toggling of the CD signal. The **getty** process will die every time CD is toggled from an on to off state. (This action could be caused by a number of reasons. Be sure to verify that the cable is properly shielded. Logging in and out several times in rapid succession can cause this.)

No login prompt displayed after connection to modem

Make sure **getty** is running on the port. If it is, verify that the carrier detect connection to modem signal is being raised after the remote side has connected to the modem. If CD is being properly asserted, then verify the modem is connected to the right port. If you still do not see login, then attach a terminal with interposer to the cable in place of the modem and verify that a login prompt does appear. If you still don't see a prompt, try to echo characters to the terminal screen to verify the cable and hardware are functioning properly.

When a remote modem connects, it immediately disconnects

Verify that the modem is talking to the server at the same speed at which the server is listening to the modem. Try different baud rates for the tty, or program the modem to lock DTE speed to match the speed of the tty port. Verify that the modem or port is not keeping the carrier detect signal high or that the port is already being used by another process.

Getting garbage characters instead of a login prompt

This is due to a difference in protocols. Be sure to verify that the modem and the tty port agree on the same parity, baud rate, flow control, and character size.

Sometimes, after a successful session, no one is able to login

It may be that the modem does not reset after disconnect. See the modem manual to see how the modem can be set to reset after an on to off DTR transition.

Problem	Resolution
Receiver buffer overruns in errpt	The UART chip buffer is being overrun. Lower the value of the receive trigger in SMIT for the tty. This solution is only valid for the native, 8– or 16–port asynchronous adapters. Verify that the modem and tty port are using the same flow control.
ttyhog errors in errpt	Modem and tty either do not agree on flow control, or no flow control is taking place.

Where to Get Additional Assistance

- Your local area representative can assist in the modem configuration.
- Another source of assistance for customers, There are many different support options available to customers in the Support Services offered, including on–site assistance or over–the–phone support. Contact your nearest service representative for assistance.
- Perhaps an often overlooked source of help is the modem manufacturer themselves. Most manufacturers have some type of online assistance for their products.

/usr/lib/uucp/Dialers Sample File Entries

The examples in this section are supplied without any warranty and will work as is for the models mentioned, but may not meet your specific needs. Some modifications will be required to meet your individual needs. Consult your modem manual for a more detailed explanation of the settings.

To use the settings to program the modem, you need an entry in the **/usr/lib/uucp/Systems** file such as:

```
hayes Nvr HayesPRGM Any
```

The **/usr/lib/uucp/Devices** file should have an entry such as:

```
HayesPRGM tty0 - 2400 HayesProgrm2400
```

With the above two entries made, use the following **cu** command to program the modem:

```
cu -d hayes
```

```
# @(#)28 1.3 Dialers.samples, cmduucp, bos325 1/23/94 11:23:14
#
# COMPONENT_NAME: cmduucp
#
#
# (C) COPYRIGHT International Business Machines Corp. 1994
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM
# Corp.
################################################################
# Motorola UDS Modem
#
# Use udsmodemPROGRAM to program the modem.
# Port needs to have rts/cts set.
# Use uds or hayes dialer.
#
# The ”udsmodemPROGRAM” line should be a single, continuous line
#
################################################################
udsmodemPROGRAM =,-, ”” \dAT&FQ2\r\c OK
ATE0Y0&C1&D2&S1%B5%E0*LC\r\c OKAT&K3&W\r\c OK

uds =,-, ”” \dAT\r\c OK\r ATDT\T\d\r\c CONNECT
```

```
#################################################################
#
# IBM 7855 Model 10
# Use IBMProgrm to program the modem.
# This sets rts/cts flow control, turns
# off xon/xoff, and sets the DTE speed at 19,200 bps.
# The modem will connect at the appropriate speed and
# flow control with the server.
# Port needs to have rts/cts set.
#
# The ”IBMProgrm” line should be a single, continuous line
#
################################################################
IBMProgrm =,-, ”” \dATQ0\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OK AT&C1\R2\Q2\M14\r\c OK AT&B8N1L0E0\A0\r\c OK
ATS0=1\r\c OK ATQ1&W0&Y0\r\c ””

#######################################################
# The following are used for Dialing out on a 7855
# regular ACU device. We have to turn on result
# codes (Q0) because they are turned off when we
# programmed it. (Keeps all upper case login from
# happening on dial in attempts.)
# We have to have an extra ”\” before ”\N” because
# the BNU programs strips it if it’s before an ”N”.
#######################################################
ibm =,-, ”” \dATQ0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECL (No Compression)
ibmecl =,-, ”” \dAT\\N3%C0Q0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECLC (Compression)
ibmeclc =,-, ”” \dAT\\N3%C1Q0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 ECLC Compression with 256 byte block size
ibmeclc256 =,-, ”” \dAT\\N3%C1Q0\A3\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 1200bps
ibm_ne12 =,-, ”” \dATQ0\\N0&A2%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 2400bps
ibm_ne24 =,-, ”” \dATQ0\\N0&A3%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 9600bps
ibm_ne96 =,-, ”” \dATQ0\\N0&A6%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 19200bps
ibm_ne192 =,-, ”” \dATQ0\\N0%C0\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 12000bps
ibm_ne120 =,-, ”” \dATQ0\\N3%C0&AL8\r\c OK ATDT\T\d\r\c CONNECT

# IBM 7855 No Compression 1200bps (Dial Quietly)
ibmq12 =,-, ”” \dATQ0\r\c OK AT&A2M0DT\T\d\r\c CONNECT

# IBM 7855 No Compression 2400bps (Dial Quietly)
ibmq24 =,-, ”” \dATQ0\r\c OK AT&A3M0DT\T\d\r\c CONNECT

# IBM 7855 No Compression 9600bps (Dial Quietly)
ibmq96 =,-, ”” \dATQ0\r\c OK AT&A6M0DT\T\d\r\c CONNECT

# IBM 7855 No Compression 19200bps (Dial Quietly)
ibmq192 =,-, ”” \dATQ0\r\c OK ATM0DT\T\d\r\c CONNECT
```

```
################################################################
#
# Intel 9600EX Modem
# Use IntelProgram to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff.
# Port needs to have rts/cts set. (Use hayes dialer)
#
# The ”IntelProgram” line should be a single, continuous line
#
################################################################
#IntelProgram =,–, ”” \d\dAT\r\c OK AT&F\r\c OK AT&S1M1\r\c OK
AT&D3\r\c OKAT&C1\r\c OK ATL0E0Y0&Y0\X1\r\c OK ATS0=1\r\c OK
AT&W\r\c OK

################################################################
# Practical Peripherals 1440FXMT Modem
# Use PracPerProgram144 to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff. (Use hayes dialer)
# DTE speed will be locked at connect speed when
# the modem is programmed. (Suggestion: 38400 baud)
#
# The ”PracPerProgram144” line should be a single, continuous
# line
################################################################
PracPerProgram144 =,–, ”” \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1&K3\r\c OK ATQ2E1&Q9\r\c OK ATS0=1S9=20\r\c OK
AT&W\r\c OK

################################################################
# Practical Peripherals 9600 bps Modem
# Use PracPerProgram9600 to program the modem.
# This sets rts/cts flow control, and turns
# off xon/xoff. (Use hayes dialer)
#
# The ”PracPerProgram144” line should be a single, continuous
# line
################################################################
PracPerProgram9600 =,–, ”” \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1&K3\r\c OK ATL0E0\r\c OK ATS0=1S9=20\r\c OK
AT&W\r\c OK

################################################################
# Practical Peripherals 2400 bps Modem
# Use PracPerProgram to program the modem
#
# The ”PracPerProgram2400” line should be a single, continuous
# line
################################################################
PracPerProgram2400 =,–, ”” \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1\r\c OK ATL0E0\r\c OK ATS0=1S9=20\r\c OK
AT&W\r\c OK

################################################################
# Hayes 2400 bps Modem
# Use HayesProgrm2400 to program the modem.
# (Use hayes dialer to dial)
#
# The ”HayesProgrm2400” line should be a single, continuous line
#
################################################################
HayesProgrm2400 =,–, ”” \d\dAT\r\c OK AT&F\r\c OK ATM1\r\c OK
AT&D3\r\c OKAT&C1\r\c OK ATL0E0\r\c OK AT S0=1\r\c OK AT&W\r\c OK
```

```
#################################################################
# Telebit t2000 Trailblazer Plus
# Use TelebitProgrm to program the modem
# This sets rts/cts flow control, and turns
# off xon/xoff and sets the Default DTE speed at
# 19,200 bps.
# Port needs to have rts/cts set.
# This sets modem to send PEP tones last as they can
# can confuse some other modems.
#
# The "TelebitProgram" line should be a single, continuous line
#
################################################################
TelebitProgram =,-, "" \dAT&F\r\c OK
ats2=255s7=60s11=50s41=2s45=255s51=254s52=2s54=3s58=2s64=1s66=1\r
\c OK
ATs69=1s92=1s96=0s105=0s110=1s111=30s130=3s131=1F1M0Q6TV1W0X3Y0\r
\c OK
ATE0&W\r\c OK
# Telebit T2000 dialers Entries:
# Forces a PEP connection:
tbfast =,-, "" \dATs50=255s7=60\r\c OK\r ATDT\T\r\c
CONNECT-\d\c-CONNECT

# 2400bps connection:

#tb2400 =,-, "" \dATs50=3\r\c OK\r ATDT\T\r\c CONNECT

# 2400 MNP:
tb24mnp =,-, "" \dAT\r\c OK ATS0=0S95=2S50=3S41=0\r\c OK
ATDT\T\r\c CONNECT

# 1200bps connection:#tb1200 =,-, "" \dATs50=2\r\c OK\r
ATDT\T\r\c CONNECT

# 1200 MNP:
tb12mnp =,-, "" \dAT\r\c OK ATS0=0S95=2S50=2S41=0\r\c OK
ATDT\T\r\c CONNECT

################################################################
# Telebit WorldBlazer
# WORLDBLAZERProgram sets the DTE speed at 38400, but
# you could set it higher if the DTE connection can
# handle it. We answer with PEP tones last so as not
# to confuse other modems. This turns off xon/xoff
# and turns on RTS/CTS flow control. The port should
# be locked to 38400 with these settings, and needs
# to have RTS/CTS turned on.
#
# The "WORLDBLAZERProgram" line should be a single, continuous
# line
################################################################
WORLDBLAZERProgram =,-, "" \dAT\r\c AT AT&F3M0\r\c AT
ATs51=253s92=1\r\c ATAT&W\r\c AT

#######################################################
# ACU Dialers for various BAUD rates for the
# WorldBlazer - each sets the modem to attempt to
# connect at a specific speed and lower. The
# WBlazer will accept whatever the remote modem can
# do. You will want to use PEP for other Telebits,
# so use WBlazer38400 or WBlazer19200 for those
#######################################################
```

```
# WBlazer =,-, "" \dAT\r\c OK ATDT\T\d\r\c CONNECT
WBlazer38400 =,-, "" \dATs50=255\r\c OK ATDT\T\d\r\c CONNECT
WBlazer19200 =,-, "" \dATs50=255\r\c OK ATDT\T\d\r\c CONNECT
# WBlazer14400 attempts to negotiate a V.42bis connection.
WBlazer14400 =,-, "" \dATs50=7\r\c OK ATDT\T\d\r\c CONNECT

# For a V.32 connection:
WBlazer9600 =,-, "" \dATs50=6\r\c OK ATDT\T\d\r\c CONNECT

# For a V.22 connection:
WBlazer2400 =,-, "" \dATs50=3\r\c OK ATDT\T\d\r\c CONNECT

# For a 1200 bps connection:
WBlazer1200 =,-, "" \dATs50=2\r\c OK ATDT\T\d\r\c CONNECT
```

128–Port Modem Cabling Considerations

AIX does not require DSR in modem–control applications, and since almost all of today's modems have autoanswering capability, the Ring Indicator signal is generally unnecessary.

ALTPIN Modem Wiring for RJ–45 Cabling

The 10–pin RJ–45 plugs is not the predominant cabling subsystem and may be difficult to obtain in the retail market. The AIX TTY subsystem provides an optional feature called ALTPIN, which swaps the logical functions of DSR (Data Set Ready) with DCD (Data Carrier Detect) for a port. When ALTPIN is enabled, DCD becomes available on pin 1 of an 8–pin RJ–45 connector (equivalent to pin 2 of a 10–pin connector).

If you wish to build an 8–wire modem cable for the 128–port RAN, use the 8–pin RJ–45 plug wired as in the following figure:

SYSTEM END CONNECTOR 8-pin RJ-45	DEVICE END	
	RI	22
1	DSR*	6
2	RTS	4
3 ..(Chassis)..	GND	SHELL
4	TxD	2
5	RxD	3
6 ..(Signal)....	GND	7
7	CTS	5
8	DTR	20
	CD*	8

* The physical location of DSR and CD may be swapped with the ALTPIN parameter when enabled using the stty-cmxa command.

The following figure is a candid look at the asynchronous signal communication between the system unit and an attached modem. Here, data is being sent from the system unit to a remote system.

DEVICE	SIGNAL	ON/OFF	MEANING
Computer	DTR	+	Hey, modem, are you ready to connect to another system?
Modem	DSR	+	Yes, I am ready. Go ahead and dial.
Modem	DCD	+	I’ve got the other system on the phone.
Computer	RTS	+	OK, can I send data now?
Modem	CTS	+	Sure, go ahead.
Computer	TxD		Sending data out to modem.
Modem	RxD		I’ve received the data.
Modem	CTS	–	Don’t send me anymore data, I’m sending it out...
Modem	CTS	+	OK, I’m ready for more data, let me have it!

Transmit data steps may be repeated until ...

Computer	DTR	–	FINISHED! Go ahead and hang up.
Modem	DCD	–	OK.

Here is the signal communication between a RISC and a modem about to receive an incoming call from another system.

Computer	DTR	+	I’m ready and have “enabled” the port for dial-in.
Modem	DSR	+	I’m ready also but I’m just waiting around for a call.

Someone calls in!

Modem	DCD	+	Alright! Somebody called in and I’ve got them on the line.
Modem	CTS	+	I’ve got data from another box, can I send you data now?
Computer	RTS	+	I’m ready to receive...go ahead and send.
Modem	RxD		Here it comes!

Modem continues to send data to computer until ...

Computer	RTS	–	WAIT! My buffer is full, don’t send anymore data yet.
Computer	RTS	+	I’m OK now...send me more data.
Modem	DCD	–	Call has ended.
Computer	DTR	–	OK, please hang up.

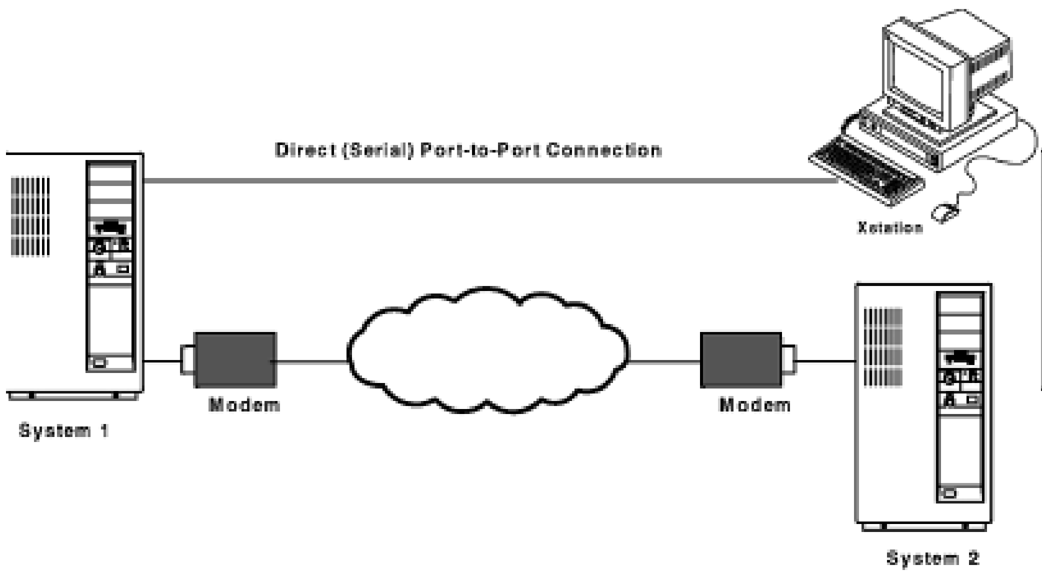
Chapter 11. Asynchronous Communication Applications

This section describes the communication for asynchronous applications.

Topics discussed are:

Serial Line Internet Protocol

Serial Line Internet Protocol (SLIP) is the protocol which TCP/IP uses when operating through a serial connection as shown in the following figure. It is commonly used on dedicated serial links and dial–up connections that operate at speeds between 1200bps and 19.2Kbps or higher.



SLIP Dedicated Serial Links

Note: To use baud rates higher than 38400, specify a baud rate of 50 in the /**etc**/**uucp**/**Devices** file for the desired tty, then change the SMIT configuration for that tty to reflect the actual baud rate desired.

For example. to run the **cu** command on tty0 with a baud rate of 115200, use the following procedure:

1. Ensure the hardware supports the baud rate.
2. Edit /**etc**/**uucp**/**Devices** to include the following line:

   ```
   Direct tty0 – 50 direct
   ```

3. Enter the **smit chtty** fast path.
4. Select tty0.
5. Change the baud rate to 115200.
6. Exit SMIT.

See the following for further discussions of SLIP:

- SLIP Configuration Steps, on page 11-3
- Modem Considerations, on page 11-3
- Manual Modem Programming Using cu, on page 11-4
- Automated Modem Configuration, on page 11-5
- Configuring SLIP over a Modem, on page 11-6
- Configuring SLIP on an Xstation 130, on page 11-8
- Temporarily Deactivating a SLIP Connection, on page 11-11
- Activating a SLIP Connection, on page 11-11

SLIP Configuration Steps

There are two recommended steps to follow during SLIP configuration. Using this two–step approach separates the hardware and machine–dependent configuration requirements from the SLIP software and command syntax problems.

Step 1

Use ATE or the **cu** utility to accomplish a successful login at the remote system. This proves the usability and correctness of the physical link.

It is important to verify the operability of any modems involved in a SLIP link as they are the most frequent cause of problems during the setup phase. For this reason, topics on modem considerations and modem programming are presented before any SLIP configuration procedures in this chapter.

Step 2

After establishing an error–free login to the remote system using ATE or the **cu** command, the user can begin the SLIP configuration.

Modem Considerations

When configuring modems for SLIP, it is important that the following changes be made on both ends of the communication link. Both the local and remote modems must be configured exactly the same.

1. The modem must acknowledge the presence of DTR.

   Referencing the local modem, if DTR is assumed or ignored, the modem can never perform a hang–up. It can only close the line or hangup when it recognizes the loss of carrier from the other end. This means that disconnects can occur only when instigated by the other end. The AT commands &D2 or &D3 are proper settings for most Hayes–compatible modems.

2. The modem must never force, assume, or ignore data carrier detect (DCD).

   DCD must follow or track the real condition. This means that carrier will exist after a bona fide connection to the other end (modem) across the switched telephone line. This also applies to a dedicated line. &C1 is the suggested setting for most Hayes–compatible modems.

3. The modem must never force, assume, or ignore a clear to send (CTS) signal.

   CTS must track or follow request To send (RTS). If CTS is forced true, the port open will fail whenever a **getty** is put upon the port or when RTS flow control protocol is added to the port.

4. Modems should be configured to turn off automatic repeat request (ARQ) codes if problems arise during **slattach** dial attempts.

   If, the modems repeatedly fail to make a connection during **slattach** dial–in attempts, the user should check the modem configurations and turn off the ARQ codes if they are currently on. In most Hayes–compatible modems, this is the &A0 setting.

   Disabling ARQ result codes does not affect error–controlled connections nor does it keep the modem from returning standard CONNECT messages (if result codes are enabled) as needed for the **slattach** dial string.

5. ECL (Error Checking on the Link) is critical.

Either BOTH modems or NEITHER modem can use it. Normally, both modems must agree on its usage during the connect session. If ECL is chosen, the physical telephone line must be good enough to allow a recovery from a data error before the TCP/IP timers expire while awaiting an acknowledge packet for the last data sent across the SLIP link.

6. Data Compression across the link.

   It is acceptable to use data compression across the link as long as it is totally handled by the modems. SLIP does not perform any type of compression. If data compression is invoked, it is much better to have two modems of the exact same type; this ensures that each will perform the compression in the same manner and same time frame.

Manual Modem Programming Using cu

Use the following procedure to manually program modems attached to the system unit.

Prerequisites

- The UNIX–to–UNIX Copy Program (UUCP) must be installed on the system. Use the **lslpp –f | grep bos.net.UUCP** command to verify installation.
- A modem must be attached to the system and powered on.
- Root user authority is needed to change the appropriate files.

Procedure

1. Add the following line to the **/etc/uucp/Devices** file if it does not already exist (replace # with the number for your port).

   ```
   Direct tty# - Any direct
   ```

   Note: Any line in the Devices file which begins with a # sign in the leftmost column is a comment.

2. Save and exit the file.
3. Enter the following command on the command line:

   ```
   cu –ml tty#
   ```

4. A connected message should appear on the screen indicating that the modem is connected and ready to be programmed.
5. Type AT and press Enter. The modem will respond with OK. If there is no response from the modem or if characters typed do not appear on the screen, check the following:
   – Verify modem cabling connections.
   – Verify that modem is powered on.
   – Observe the modem front panel lights when you press Enter. If the Receive Data (RD) and Send Data (SD) lights flash, then the modem is communicating with the system and the problem may lie with the current modem settings. If the lights do not flash, then the problem is with the modem connection.
   – Type the following and see if the condition changes:

   ```
   ATE1 <enter>
   ATQ0 <enter>
   ```

   ATE1 turns the echo mode on which displays any typed characters to the screen. ATQ0 enables the displaying of result codes.

6. Program the modem using the settings shown in the previous section, "Modem Considerations." The following example demonstrates how to program and save basic settings for a Hayes–compatible modem. Enter:

```
AT&F    <enter>
AT&D2   <enter>
ATS0=1  <enter>
ATS9=12 <enter>
AT&C1   <enter>
AT&W    <enter>
~.      <enter>
```

Where &F is used to reset the modem to factory defaults, &D2 sets DTR, S0 and S9 set register values, &C1 set carrier, and &W writes the settings to the modem. The tilde–period ends the connection.

Automated Modem Configuration

Users can customize their modems manually or use the **cu** utility with its associated files to create an automated modem configuration script.

Prerequisites

- UUCP must be installed on the system. Use the **lslpp –f | grep bos.net.UUCP** command to verify installation.
- A modem must be attached to the system and powered on.
- The modem AT command string must already exist (for example, `at&f&c1&d3`). Users should not attempt automated modem configuration until the command string has first been tried manually using the **cu** command.
- Root user authority is needed to change the appropriate files.

Procedure

The following example shows how to automatically configure a Telebit T3000 modem attached to tty0.

1. Edit the **/etc/uucp/Systems** file.
2. Add the following line at the end of the file. The entry should begin in the leftmost column of the file.

   ```
   telebit Nvr TELEPROG 19200
   ```
3. Save and exit the file.
4. Edit the **/etc/uucp/Devices** file.
5. Add the following line at the end of the file. The entry should begin in the leftmost column of the file.

   ```
   TELEPROG tty0 - 19200 TelebitProgram
   ```
6. Save and exit the file.
7. Edit the **/etc/uucp/Dialers** file.
8. Add the following lines at the end of the file. The entries should begin in the leftmost column of the file.

   Note: The following four lines should be made into one long line:

   ```
   TelebitProgram  =,-,     ”” \dAT&F\r\c OK
   ats0=1s2=255s7=60s11=50s41=2s45=255s51=252s63=1s58=2s64=1\r\c OK
   ATs69=2s105=0s111=30s255=0M0&C1Q2&D3&Q0&R3&S1&T5\r\c OK
   ATE0X12&W\r\c OK
   ```
9. Save and exit the file.
10. To begin the automated configuration, enter the following command:

    ```
    cu –d telebit
    ```

The command will fail because you are not connecting to a system. Watch the debug output of the command to see that `ATE0X12&W` is sent to the modem and that an OK is received. If so, then the modem has been successfully programmed.

Problems may arise because of incorrect values placed in the **Dialers** file or because of the modem's existing configuration. If this occurs, try programming the modem manually and enter the dialers strings (in step 8) one by one.

Configuring SLIP over a Modem

The following procedure lists the steps necessary to configure a SLIP line between two system units communicating over 9600 baud modems. For clarity, these instructions use the names `systemA` and `systemB` for the two hosts.

Prerequisites

1. You must have root user authority.
2. The modems must be physically connected to `systemA` and `systemB`.

Procedure

1. To create a tty on `systemA` through SMIT, use the **smit tty** fast path.
2. Select **Add a TTY.**
3. Select `rs232` as the type of tty you want to create.
4. Select an available serial port, for example `sa0`.
5. Select a port number for this tty. Position the cursor on the PORT number field; list and select an available port number from the list.
6. Set the BAUD rate to the baud rate of the modem.
7. Set Enable LOGIN to **disable**.
8. Set FLOW CONTROL to be used to **RTS** or **none**. This option is available only on systems with AIX Version 4.
9. Select **Do**.
10. Exit the SMIT interface.
11. To create a tty on `systemB`, repeat steps 1–10, except set Enable LOGIN to **enable**.

    Note: The remaining instructions assume that the tty number on both systems is `tty1`.
12. Test the physical modem connection with the **cu** command.
    a. On `systemA`, enter:

    ```
    cu –m1 tty1
    ```

    b. Once connected to the modem, type:
       - ATDT ###–####
       - Where ###–#### is the phone number of the remote machine.

    c. At this point, a login prompt from `systemB` should appear. If not, return to the section entitled "Configuring Modems for SLIP" and verify modem setup for both systems. *Do not continue* unless a successful login to `systemB` is achieved.

    d. Log in to `systemB`.

    e. If `systemB` is NOT at AIX Version 4.1 or greater, then once logged into `systemB`, enter **stty add rts** at the command line. This adds RTS/CTS line discipline to `tty1` which SLIP uses to operate more reliably.

    If `systemA` is also NOT at AIX Version 4.1 or greater, issue the same command on it by entering:

```
stty add rts < /dev/tty1
```

f. Now type **exit** to logoff of `system B` and type **~.** to exit **cu**.

13. Since the tty configuration for use with **cu** is slightly different from the configuration for use with SLIP, enter **smit chgtty** on `systemA`.
14. Select `tty1`.
15. Select **Change/Show TTY Program**. (This menu option is available on AIX Version 3.2.5 systems only.)
16. Remove all occurrences of ixon, ixoff and ixany in the stty attributes for RUN TIME and stty attributes for LOGIN fields. (This option is is available only on AIX Version 3.2.5 systems.)
17. Select **Do**.
18. On systemB, enter the smit chgtty fast path.
19. Select tty1.
20. Select **Change/Show TTY Program**. (This menu option is available only on AIX Version 3.2.5 systems.)
21. Set Enable LOGIN to **disable**.
22. Remove all occurrences of ixon, ixoff and ixany in the stty attributes for RUN TIME and stty attributes for LOGIN fields. (This option is is available only on AIX Version 3.2.5 systems.)
23. Select **Do**.
24. Exit the SMIT interface.
25. Add the following line to the **/etc/uucp/Devices** file on both `systemA` and `systemB`:

    ```
    Direct tty1 - 9600 direct
    ```

    This entry must precede any other entry for tty1 in the Devices file and should always begin in the leftmost column.
26. To create a SLIP network interface on `systemA`, use the **smit mkinet** fast path.
27. Select **Add a Serial Line INTERNET Network Interface**.
28. Select `tty1`.
29. Specify the INTERNET ADDRESS of `systemA`. For example:

    ```
    [130.130.130.2]
    ```
30. Specify the DESTINATION Address of `systemB`. For example:

    ```
    [130.130.130.1]
    ```

    DO NOT make entries in the BAUD RATE or DIAL STRING fields at this time. These entries can be added later after the correct operation of SLIP is verified through command line options.
31. Select **Do**.
32. To create a SLIP network interface on `systemB`, repeat steps 26–31. Change the network addresses as follows if following the examples provided:

    ```
    INTERNET ADDRESS      [130.130.130.2]
    DESTINATION Address   [130.130.130.1]
    ```
33. Add the following two entries to the **/etc/hosts** file on both `systemA` and `systemB`:

    ```
    130.130.130.1    systemA
    130.130.130.2    systemB
    ```

The name assigned should be unique. In other words, if the Token–Ring interface on `systemA` is already assigned the name `systemA`, assign the SLIP interface a name such as systemA_slip.

34. Start SLIP on `systemB` by entering:

```
slattach tty1 9600
```

35. Start SLIP on `systemA` by entering:

```
slattach tty1 9600 ' "" AT OK ATDT555-1234 CONNECT "" '
```

The command can be read as: Use `tty1` at 9600 baud. Send AT to the modem. The modem should respond with OK. Dial the phone number 555–1234. The modem should respond with CONNECT.

Users may add the number nine to the end of the **slattach** dial string in order to obtain debug information during the dial attempt. For example:

```
slattach tty1 9600 ' "" AT OK ATDT555-12349 CONNECT "" ' 9
```

This debug output is similar to information displayed when using the **cu –d** command on a tty device.

Note: The debug option is not a supported feature of **slattach** and is supplied on an as–is basis. Non–supported features may be removed at any time without notice to the users.

36. Test the SLIP connection using the **ping** command.

a. On `systemA` enter: **ping systemB**.

b. On `systemB` enter: **ping systemA**.

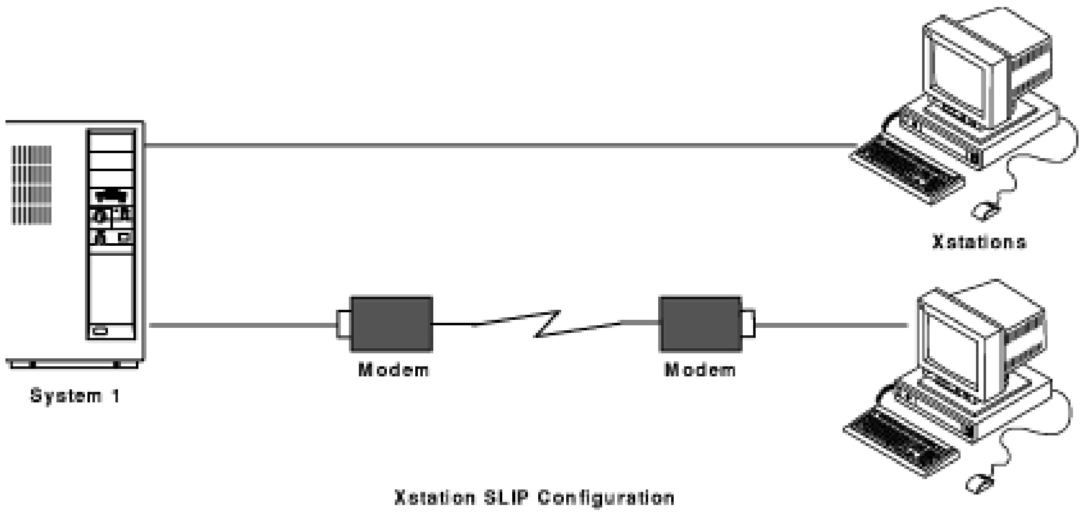
If both tests succeed, the SLIP connection is ready for use. If not, return to step 25 and verify that the configuration on both systems is correct.

Configuring SLIP on an Xstation 130

Serial Line Internet Protocol (SLIP) is a driver that provides point–to–point connection from an Xstation 130 to a host providing boot service and X protocols. This allows the Xstation 130 to use the capabilities of remote hosts not physically connected by an Ethernet or token–ring network.

The SLIP protocol recognizes Internet Protocol (IP) packets on a serial line, runs Xclients on the Xstation from a serial–line connected host, and runs some Internet Control Message Protocols (ICMPs), such as those used by the **ping** command.

Two methods of SLIP configuration from an Xstation to a host system are shown in the following figure. Xstations may be directly attached to the host system through a null modem cable or they may be attached over modems.



Xstation SLIP Configuration

Procedure

1. Connect the serial–link between the host and the Xstation 130.

   Connect one end of the EIA–232 serial cable to the selected serial port on the host, and the other end to the selected serial port (s0, s1, s2, or s3) on the Xstation 130 or a host modem. If a host modem is used, connect the selected Xstation serial port to its modem.

2. To configure the host for SLIP, enter the host Internet address, host name, Xstation Internet address, and Xstation name at the name server or in the host's /**etc**/**hosts** file.

3. To configure a tty on the host for SLIP, use the **smit tty** fast path.

4. Select **Add a TTY**.

5. Select **tty rs232 Asynchronous Terminal** as the tty type to configure.

6. Select **sa0** or **sa1** as the parent adapter.

7. Change the values, if required, for the following fields:

PORT number	Select List and choose the correct port number on the host from the displayed list.
BAUD rate	Select List and choose the baud rate that matches the host modem or Xstation 130.
PARITY	Set this field to **none**.
BITS per character	Set this field to **8**.
Number of STOP BITS	Set this field to **1**.
Enable LOGIN	Set this field to **disable**.
XON–XOFF handshaking	Set this field to **no**. (This option is not available on AIX 4.1 or higher.)
FLOW CONTROL to be used	Set this field to **rts** or **none**.

8. Select **Do** to complete the configuration of the host tty serial port.

9. To add a Communications Serial Line Interface to the host, use the **smit mkinet** fast path.

10. Select **Add a Serial Line INTERNET Network Interface**.

11. Select the previously configured tty port on the system as the SLIP connection.

12. Fill in values for the following fields:

INTERNET ADDRESS	Enter the same host address as in step 2.
DESTINATION Address	Enter the same address as in step 2.
Network MASK	If applicable, enter the network mask in dotted decimal format.
BAUD RATE	If a dial string is used, select **List** to select the correct baud rate to use for the dial string and modem.
DIAL STRING	If applicable, enter the string used to call up the Xstation 130 through the modem. **Note:** The modem may require a carriage return (a single bell note) at the end of the dial string.

13. Select **Do** to complete the configuration of the Serial Line Interface.
14. To attach a serial communications line (tty), use the **slattach** command.
15. To detach the interface, use the **ifconfig *InterfaceName* down** command after ending the **slattach** command. The *InterfaceName* parameter is the name shown by the **netstat** command.
16. Unless a network type has been defined, use the **smit x_config** fast path to display the **Xstation Configuration** menu.
17. Select **Define an Xstation Network Type**. Provide the values to define an Ethernet network for the Xstation. This network type must use **bootfile4**.
18. Select **Do** to save the defined Ethernet network.
19. **Cancel** to return to the **Xstation Configuration** menu.
20. To add an Xstation using the network type defined above, select **Add an Xstation**.
21. Select **Xstation 130** from the **Add an Xstation** menu.
22. Provide the field values. Be sure to use the Xstation name added in step 2 and the Network Type Name defined in step 17. A hardware address is not required.
23. If using modems, set them up according to their instruction manuals. If desired, the Xstation can be used as an asynchronous terminal to set up the Xstation modem.
24. Configure the Xstation 130 for SLIP at the Xstation, as follows:
    a. Power on the Xstation 130.
    b. When the LAN Statistics screen is displayed, press F12 to set the SLIP parameters.
    c. When the **Network Setup** menu is displayed, use the up and down cursor keys to move the highlighted bar. Enter the following information:

Primary Network	Select the network from which to boot the Xstation.
Enable SLIP	Select **YES**.
Serial Port	Select **s0** as the serial port where the EIA–232 cable is connected. SERIAL 1 is **s0**.
Baud rate	Select the correct baud rate. Ten values range from 110 to 38400. If modems are not used, this value must match the system value in step 2b.
Terminal Internet Address	Enter the SLIP Internet address for the Xstation 130 in dotted decimal format. This address must match the DESTINATION Address configured on the system in step 2c.
Host Internet Address	Enter the SLIP Internet Address for the system in dotted decimal format.

Subnet Mask	If applicable, enter the subnet mask in dotted decimal format.
Dial String	If applicable, enter the string used to dial the system connection through the modem.
Disable Bootp	Select **NO**.

d. Press the F12 key to save the Network Setup and return to the LAN Statistics screen.

Temporarily Deactivating a SLIP Connection

To temporarily deactivate a SLIP connection, do the following on both the local and remote systems:

1. Enter:

   ```
   ifconfig sl# down
   ```

2. List the currently running **slattach** processes using the command:

   ```
   ps –ef | grep slat
   ```

   The output may be similar to the following:

   ```
   root  1269  1    0   Jun 25  ... slattach
   ```

3. Kill the **slattach** process using its process ID. For example, to kill the **slattach** process shown above enter:

   ```
   kill 1269
   ```

   where 1269 is the **slattach** process ID. Do NOT remove the **slattach** process using the –9 flag of the **kill** command.

   The SLIP connection is now disabled.

Activating a SLIP Connection

Use the following instructions to activate a SLIP connection that is temporarily disabled using the above instructions. Run these commands on both the local and remote systems.

1. Enter:

   ```
   ifconfig sl# up
   ```

2. Re–issue the **slattach** command used initially. Review the instructions on pages 19 and 20 for additional help with this step.

Removing a SLIP Interface

Use the following instructions to completely remove a SLIP interface. Once these instructions are executed both the sl# interface and its associated **slattach** process are removed. Any entries made to the /**etc**/**hosts** file will remain and must be removed manually.

1. To remove the SLIP interface and its associated **slattach** process, use the **smit rminet** fast path to access the **Available Network Interfaces** screen.

2. Select the appropriate entry from the **Available Network Interfaces** screen and select **Do**.

   Note: Any entries made to the /**etc**/**hosts** file will remain and must be removed manually.

Debugging SLIP Problems

The following describes the commands needed to debug SLIP problems and supplies you with examples.

netstat Command

The **netstat** command works in conjunction with the **ifconfig** command to provide a status condition of the TCP/IP network interface. The command **netstat –in** for example uses the

–i flag to present information on the network interfaces while the **–n** flag prints the IP addresses instead of the host names. Use this command to verify SLIP interfaces, addresses, and hostnames. The following section describes **netstat –in** output.

Program the modem using the settings shown in the previous section, ”Modem Considerations.” The following example demonstrates how to program and save basic settings for a Hayes–compatible modem. Enter:

```
Name  Mtu   Network    Address            lpkts     Ierrs Opkts Oerrs  Col
lo0   1536  <Link>                        2462       0     2462   0     0
lo0   1536  127        localhost.austi    2462       0     2462   0     0
tr0   1492  <Link>                        1914560    0    21000   0     0
tr0   1492  129.35.16  glad.austin.ibm    1914560    0    21000   0     0
sl0   552   1.1.1.0    1.1.1.1            48035      0    54963   0     0
sl1*  552   140.252.1  140.252.1.5        48035      0    54963   0     0
```

netstat –in Command Output

Notice the * next to the sl1 interface. This shows that the network interface is down or unavailable for use. The user can correct this by issuing the **ifconfig sl1 up** command if it is a valid SLIP interface.

netstat provides statistics concerning input and output packet counts as well as input and output errors that are helpful when troubleshooting SLIP connections.

Example

The user enters a **ping** to a remote host across a SLIP link and the **ping** command appears to hang. They quickly run a **netstat –in** command from another command shell and notice that the Opkts are increasing but that there are no Ipkts from the remote host. This indicates that the remote system is not returning (or not receiving) the information. They must run the same **netstat** command on the remote system to verify the receipt of the **ping** packets or rise in the error count.

The translation of hostnames versus Internet numbers is relative to name resolution and thus critical to proper operation of a SLIP line. To debug hostname, aliases, and routing problems, use the **netstat –rn** command. The basename of the host or hostname is the only name that should return from the /**etc**/**hosts** file. If the machine is being serviced by a nameserver (ie. /**etc**/**resolv.conf** exists), then the name–server will return the fully qualified–domain name in this command.

ifconfig Command

The **ifconfig** command is the network interface configuration tool provided with AIX. It allows the network interface STRUCTURE to be dynamically created or deleted from the kernel memory in AIX. This command accepts data from the command line, then builds a memory structure that conforms to the parameters. For debugging purposes, the **ifconfig** command is used to examine the status of a communications interface.

Example

To examine the current status of the sl1 interface:

1. Enter the **netstat –i** command and examine the output selecting the appropriate sl# interface. For example, sl0, sl1, sl2, etc.
2. Enter the **ifconfig sl#** command and examine the ifconfig output for the following key fields:

POINTTOPOINT flag	This flag should always be present on an operational SLIP link. If not, the link could be in a down or disconnected state. Try issuing the **ifconfig sl# up** and the **ifconfig sl#** commands again to see if its condition changes.
UP flag	Indicates that the network sl# interface is activated and should be operational.
RUNNING flag	Indicates that the **slattach** command was successful. In actuality, the link is accessed, a dial is completed, the other end has answered, and the remote end has returned CARRIER DETECT status. When the CD status occurs the flags are updated with the running bit.

pdisable and lsdev Commands

Any tty port that is used for SLIP connections must be in a disabled or unavailable state. To verify that the port for tty1 is disabled, obtain root user authority and enter one of the following commands:

- **lsattr –El tty1 –a login**

  This command displays the permanent state of the tty port as recorded in the system's Object Database Manager (ODM). If the output is anything other than `login disable`, use SMIT to change the enable LOGIN field to **disable**.

- **pdisable | grep tty1**

  This command, when used without parameters, displays all tty ports that are in a disabled state. In this example, **pdisable** is piped to the **grep** command to eliminate unnecessary output. If tty1 is not displayed after running this command, the port is not disabled.

ps Command

The **ps** command displays information about active processes to standard output. Use this command to verify the existence (or nonexistence) of **slattach** processes that are used to assign a tty line to network interfaces.

If **netstat –in** shows that the interface is down, the user should run the **ps –ef | grep slat** command to see if an **slattach** process is currently running on the associated tty port. Note that for a directly connected SLIP interface, broken connections are retried automatically without manual intervention. For a SLIP interface connected by modem, broken connections must be manually redialed. If a user supplies a dial string in the **slattach** command line, the user must reenter the command and dial string to restore a broken connection.

ping Command and Modem Lights

The **ping** command and modem lights are used to debug SLIP communication problems. A ping is an echo request packet, sent out of the machine, and an echo response packet is returned. This sequence of events is useful if the administrator can see the modem lights.

Example

The local system constructs the echo request packet and sends it to the remote system. The Send Data (SD) light on the local modem illuminates. This means that the local TCP/IP, **slattach**, and tty were able to group information and send it out of the modem to the remote system.

The remote modem receives the packet and the receive data light flashes but its SD light does not. This means that the remote system was not able to send (or return) the local system's ping request. As a result, the user on the local system may see the **ping** command hang, requiring a **Ctrl c** to exit the condition.

The most common cause of this problem is the use of XON/XOFF flow control in one or both modems, however, the user should not overlook the possibility of routing or address conflicts on the systems.

Common Problems and Error Messages

Message: 0821–296 Cannot set line discipline for /**dev**/**tty#** to slip.ioctl(TXSETLD). A system call received a parameter that is not valid.

Possible Causes: This type of error normally occurs when starting the **slattach** process and is attributable to incorrect configuration of SLIP. The problem is most likely caused by a mismatch between the tty device number and the sl interface number. This also explains why the system reported that ifconfig had not been run before **slattach**.

This problem may also occur when **slattach** processes are dropped or killed incorrectly or when the user attempts to move a SLIP connection to another tty port and forgets to reconfigure the sl# interface to match the tty. Check for running **slattach** processes that may still be running (for example, **ps –ef | grep slat**).

Action: The tty device for SLIP is /**dev**/**tty24** and user has created an sl0 interface. This is incorrect. The user should create an sl24 interface which matches the tty number (tty24 and sl24). If the problem continues, the user should bring down the sl interface (see "Bringing Down an SLIP Interface") and reconfigure the connection using the following commands:

```
lsdev  –Cc if  –s SL
lsattr –El sl0
```

Message:

network is not currently available

route to remote host not available

Possible Cause: These errors occur most often when a user attempts to ping a host over the SLIP link and the link has been improperly established. The most likely problem is that one or both tty ports associated with the sl# interface are in an enabled state. It is also possible that there is an address or route conflict between the host systems.

Actions:

- Remove the sl# interface using the **smit rminet** fast path This must be done on both the local and remote SLIP hosts.
- Do the following for each SLIP host:
  a. Enter **pdisable | grep tty#**.
  b. If the tty device is NOT listed in the output of the previous command, the tty is not disabled. Disable the tty either through SMIT or the command line. With tty ports disabled, use SMIT to recreate the SLIP interfaces on both systems. If problem persists, verify network addresses and routes (if any). Use the **netstat –ir** command to quickly view address, routing, and interface information.

Problem: When the remote site dials in to the local host, the modem on the local host connects but does not complete the login process.

Possible Causes: If the two modems connect and begin to handshake or exchange connection information but then disconnect, the problem may be due to modem result codes. This problem can also be caused by an improper **slattach** dial string. If the two modems ring but never begin the handshake process, the problem may be that the modem is not set for auto–answer.

Actions:

1. Test the modem connection first with the **cu** command. The modem on the remote host should allow the user to login to the system. There should not be any garbage on the screen during the login attempt; if so, it may indicate a noisy phone line which may be part of the problem. During the login, multiple login heralds should *not* scroll across the screen. If they are present, this could again indicate a problem phone line or incorrect modem settings.

2. Check the modem configurations and try turning off the ARQ codes if they are currently on. In most Hayes–compatible modems this is the &A0 setting. Disabling ARQ result codes does not affect error–controlled connections nor does it keep the modem from returning standard CONNECT messages (if result codes are enabled) as needed for the **slattach** dial string.

Problem: The user is unable to **ping** across a modem SLIP connection. The **ping** command may hang or return error messages.

Possible Causes:

1. The modems and/or tty ports may be configured to use XON/XOFF flow control.
2. The **slattach** process may have been terminated on the remote host or the modem connection dropped.
3. The addresses assigned to the SLIP hosts may be incorrect.

Actions:

1. Examine both the local and remote modem configurations. They should be set to use RTS/CTS (hardware) flow control or no flow control at all. The user should attempt to ping from each system. Ping systemA to systemB.
2. Verify that the **slattach** process is still running on both local and remote systems. Use the command: **ps –ef |grep slat**. Verify that the sl# interface is in a running state. Use the command: **ifconfig sl#**.
3. Verify that there is not a conflict between the SLIP addresses and those associated with other network interface (if any). Use the command: **netstat –ir**. If the address or address class is in question, reconfigure SLIP using a simpler address scheme such as 1.1.1.1 for the local host and 1.1.1.2 for the remote host.

   Additional help with SLIP setup is available through ConsultLine which is a fee based offering of the AIX Support Family.

   Use the SLIP questionnaire located on the next page to record information on the existing SLIP configuration before calling the help line. In this way, the user can more efficiently relay his information to help line personnel and speed up the troubleshooting process.

SLIP Questionnaire

Use this questionnaire to record data on SLIP configurations. Information collected on these sheets can be faxed to a service representative when additional assistance with SLIP configuration is required.

1. Was this SLIP configuration working previously? (Y/N) ____
2. What are the machine types? (example: AIX/PC, DOS/PC, etc.)

   Local System: ________________ Remote System: _________________

   If the host is not an AIX system, please name the type of software being used to establish the SLIP connection.

   ____
3. What versions of the AIX operating system are on each of the system units? Issue the **/bin/oslevel** command. If this command is not recognized use the following method:

   ```
   lslpp -h bos.Rte
    look for the
   active commit
    line release level.
   ```

   Local System: ________________ Remote System: _________________
4. List all interfaces available on both systems (for example, sl0, sl1). To do this, use the command: **lsdev –Cc if**

   Local System: ______________ Remote System: _____________

   ______________ _____________

   ______________ _____________

   ______________ _____________

   Note that in newer versions of AIX (for example, AIX 3.2.4 and up), the SLIP interface number should match the tty device number. For example, /**dev**/**tty53** should be used with sl53.
5. Is SLIP being configured through SMIT or with commands? _____________________

   SLIP configurations using commands are not permanent and are not present after a system reboot.
6. Is SLIP being configured over modems or a direct serial line?

   ____
7. If modems are being used, list the manufacturer and modem type for both the local and remote systems.

   ```
           TYPE      BAUD RATE  IBM CABLING If not IBM cable,
                                (Yes/No)    what type?
   Local:  _________ __________     ___     ____________________________
   Remote: _________ ___________    ___     ____________________________
   ```
8. If modems are in use, what is the phone carrier type? (leased–line or normal switched)

   ____
9. What hardware is the SLIP line being used on?

   IBM 7318 Serial Communications Network Server (model P10 or S20): ___

   128–Port Adapter (with 16–port RAN(s): ___

   64–Port Adapter (with 16–port concentrator(s)): ___

   16–Port Fanout Box (DB25 pinouts): ___

8–Port Fanout Box (DB25 pinouts): ___

Native, S1 or S2 serial ports: ___

10. Is it possible to ping from the local system to the remote system?

    `(Y/N) _______     (on the local system enter: ping <remote address> )`

11. Is it possible to ping from the remote system to the local system?

    `(Y/N) _______     (on the remote system enter: ping <local address> )`

12. Are the tty ports disabled on both the local and remote systems?

    `(Y/N) _______`

    Use the command: **pdisable | grep tty#**. Only disabled tty numbers are displayed as output of this command.

13. Are any error messages being displayed? If yes, please list them below:

____

____

____

____

____

____

____

____

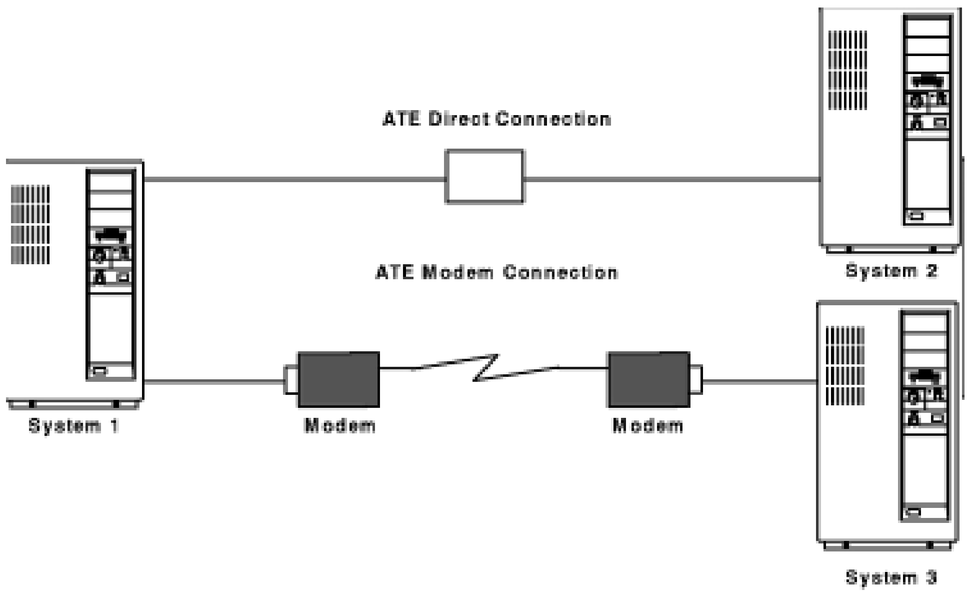
____

____

Asynchronous Terminal Emulation

The Asynchronous Terminal Emulation (ATE) program enables terminals on the AIX system to emulate a terminal, thus allowing users a means of connecting to most other systems which support asynchronous terminals. ATE accomplishes this by making the remote system see a terminal either as a system's display or as a DEC VT100 terminal. The VT100 option allows the user to log in to systems that do not support their terminal, but do support VT100 terminals.

ATE uses both direct (cabled) and modem connections to communicate between the user's system and a remote system. (See the following figure.) Depending upon the connection type used, the user can configure ATE to connect either to a system in the next room or to a system across the country.



ATE Connection Types

For a direct connection, the user must know the port to use on their system. For a modem connection, users must know the port to use on his system and the telephone number of the remote system. Users must also have a login ID and password on the remote system.

This section includes the following information about ATE:

Setting Up ATE

Before running ATE, the system administrator must install the proper software (if needed) and configure the tty ports and connections. ATE uses both direct (cabled) connections and modem connections. Local RS–232C connections allow a maximum distance of 15 meters (50 feet) between machines, and RS–422A connections allow up to 1200 meters (4000 feet) between machines.

Before using ATE to call a remote system, verify that the remote system's tty device is ready to accept a call.

Prerequisites

- ATE is an optional program product. All files necessary for operation of ATE are contained in the **bos.net.ate** program product available on the install media. Use the following commands to verify that ATE is available on your system:

  ```
  lslpp –h | more   <return>
  /bos.net.ate      <return>
  ```

  If ATE is not available on your system, install the **bos.net.ate** image from the installation media (tape, diskette, or network server).

- If ATE is installed on the system, a list of files associated with this program can be displayed using the following commands:

  ```
  lslpp –f | more   <return>
  /bos.net.ate      <return>
  ```

- The user must have root user authority to set up the port for the communications device.

Procedure

To prepare ATE to run on the system, perform the following steps:

1. Install an asynchronous adapter card in an appropriate slot in the system unit, unless the system has a built–in serial port.
2. Plug the RS–232C or RS–422A cable into the adapter card or the built–in serial port.
3. Add a tty device for the communications port using the **smit mkdev** fast path.

   Select the terminal type to emulate with ATE and make the necessary adjustments for the environment. The most common changes are line speed, parity settings, number of bits per character, and whether the line is to be driven as a remote or local line. Use bpc 8 and no parity if National Language Support (NLS) is required.
4. Set up the port for the device. To set up a port to call out with ATE, use the **pdisable** command. For example, to set up port tty1, enter:

   ```
   pdisable tty1
   ```

   To set up a port so that others can call in, use the **penable** command. For example, to let other systems call in to the tty2 port, enter:

   ```
   penable tty2
   ```
5. Ensure the device has previously been defined to the remote system. Once the device is defined, the ATE program must be customized to reflect the device settings on the remote system. Customize the default settings with the alter and modify subcommands or by editing the **ate.def** default file. To change the default settings for a telephone connection, use a dialing directory file entry.

Customizing the ATE Program

ATE creates the **ate.def** default file in the current directory the first time the user runs ATE. Edit the **ate.def** file to customize various aspects of ATE. For example, the user can change the name of the dialing directory file, the type of transfer protocols used to send and receive files from the remote system, and the baud rate ATE expects the modem to use. Refer to "Changing the Default Files" for more information on the **ate.def** file.

Users can also make temporary changes to certain aspects of ATE with the **modify** and **alter** subcommands. These subcommands can change all of the ATE default values except the control key sequences (which can only be changed by editing the default file) and the name of the dialing directory (which can be changed with the **directory** subcommand or by editing the default file). Any changes made with the **modify**, **alter**, or **directory** subcommands are effective only for that session of ATE. The next time the user runs ATE, the settings used are those defined in the default file.

When using a modem with ATE, the user can create a dialing directory of up to 20 phone numbers. The **directory** subcommand displays the telephone numbers in menu form and allows the user to select the desired system to call. Refer to ”The ATE Dialing Directory File” section for more information.

By using a dialing directory, the user avoids having to look up the telephone number when calling a particular system. The user can also specify certain data transmission characteristics in the dialing directory file. This is useful if some connections use characteristics that differ from the ATE defaults.

ate.def Configuration File

The **ate.def** file sets the defaults for use in asynchronous connections and file transfers. This file is created in the current directory during the first run of ATE. The **ate.def** file contains the default values in the ATE program uses for the following:

- Data transmission characteristics
- Local system features
- Dialing directory file
- Control keys.

The first time the ATE program is run from a particular directory, it creates an **ate.def** file in that directory.

```
LENGTH          8
STOP            1
PARITY          0
RATE            1200
DEVICE          tty0
INITIAL         ATDT
FINAL
WAIT            0
ATTEMPTS        0
TRANSFER        p
CHARACTER       0
NAME            kapture
LINEFEEDS       0
ECHO            0
VT100           0
WRITE           0
XON/XOFF        1
DIRECTORY       /usr/lib/dir
CAPTURE_KEY     002
MAINMENU_KEY    026
PREVIOUS_KEY    022
```

Edit the **ate.def** file with any ASCII text editor to permanently change the values of these characteristics. Temporarily change the values of these characteristics with the ATE **alter** and **modify** subcommands, accessible from the ATE Main Menu.

Parameters in the ate.def File

Type parameter names in uppercase letters in the **ate.def** file. Spell the parameters exactly as they appear in the original default file. Define only one parameter per line. An incorrectly defined value for a parameter causes ATE to return a system message. However, the program continues to run using the default value. These are the **ate.def** file parameters:

LENGTH Specifies the number of bits in a data character. This length must match the length expected by the remote system.

```
Options:  7 or 8
Default:  8
```

STOP Specifies the number of stop bits appended to a character to signal that character's end during data transmission. This number must match the number of stop bits used by the remote system.

```
Options:  1 or 2
Default:  1
```

PARITY Checks whether a character is successfully transmitted to or from a remote system. Must match the parity of the remote system.

For example, if the user selects even parity, when the number of 1 bits in the character is odd, the parity bit is turned on to make an even number of 1 bits.

```
Options: 0 (none), 1 (odd), or 2 (even)
Default: 0.
```

RATE Determines the baud rate, or the number of bits transmitted per second (bps). The speed must match the speed of the modem and that of the remote system.

```
Options: 50,75,110,134,150,300,600,1200,1800,2400,4800,9600,19200
Default: 1200
```

DEVICE Specifies the name of the asynchronous port used to make a connection to a remote system.

```
Options: Locally created port names.
Default: tty0.
```

INITIAL Defines the dial prefix, a string that must precede the telephone number when the user autodials with a modem. For the proper dial commands, consult the modem documentation.

```
Options: ATDT, ATDP, or others, depending on the type of modem.
Default: ATDT.
```

FINAL Defines the dial suffix, a string that must follow the telephone number when the user autodials with a modem. For the proper dial commands, consult the modem documentation.

```
Options: Blank (none) or a valid modem suffix.
Default: No default.
```

WAIT Specifies the time to wait between redialing attempts. The wait period does not begin until the connection attempt times out or until it is interrupted. If the ATTEMPTS parameter is set to 0, no redial attempt occurs.

```
Options: 0 (none) or a positive integer designating the number of
seconds to wait.
Default: 0
```

ATTEMPTS Specifies the maximum number of times the ATE program tries redial to make a connection. If the ATTEMPTS parameter is set to 0, no redial attempt occurs.

```
Options: 0 (none) or a positive integer designating the number of
attempts.
Default: 0
```

TRANSFER Defines the type of asynchronous protocol that transfers files during a connection.

p (pacing) File transfer protocol controls the data transmission rate by waiting for a specified character or for a certain number of seconds between line transmissions. This helps prevent loss of data when the transmission blocks are either too large or sent too quickly for the system to process.

x (xmodem) An 8–bit file transfer protocol to detect data transmission errors and retransmit the data.

```
Options: p (pacing), x (xmodem)
Default: p.
```

CHARACTER Specifies the type of pacing protocol to be used. Signal to transmit a line. Select one character.

When the **send** subcommand encounters a line–feed character while transmitting data, the subcommand waits to receive the pacing character before sending the next line.

When the **receive** subcommand is ready to receive data, it sends the pacing character, then waits 30 seconds to receive data. The **receive** subcommand sends a pacing character again whenever it finds a carriage return character in the data. The **receive** subcommand ends when it receives no data for 30 seconds.

```
Options: any character
Default: 0
```

Interval Number of seconds the system waits between each line it transmits. The value of the Interval variable must be an integer. The default value is 0, indicating a pacing delay of 0 seconds.

```
Default: 0.
```

NAME File name for incoming data (capture file).

```
Options: A valid file name less than 40 characters long.
Default: kapture
```

LINEFEEDS Adds a line–feed character after every carriage–return character in the incoming data stream.

```
Options: 1 (on) or 0 (off).
Default: 0.
```

ECHO Displays the user's typed input. For a remote computer that supports echoing, each character sent returns and displays on the screen. When the ECHO parameter is on, each character is displayed twice: first when it is entered, and again when it returns over a connection. When the ECHO parameter is off, each character displays only when it returns over the connection.

```
Options: 1 (on) or 0 (off).
Default: 0.
```

VT100 The local console emulates a DEC VT100 terminal so DEC VT100 code can be used with the remote system. With the VT100 parameter off, the local console functions like a workstation.

```
Options: 1 (on) or 0 (off).
Default: 0.
```

WRITE Captures incoming data and routes it to the file specified in the NAME parameter as well as to the display. Carriage–return or line–feed combinations are converted to line–feed characters before they are written to the capture file. In an existing file, data is appended to the end of the file.

The CAPTURE_KEY (usually the Ctrl–B key sequence) can be used to toggle capture mode on or off during a connection.

```
Options: 1 (on) or 0 (off).
Default: 0.
```

XON/XOFF Controls data transmission at a port as follows:

- When an XOFF signal is received, transmission stops.
- When an XON signal is received, transmission resumes.
- An XOFF signal is sent when the receive buffer is nearly full.
- An XON signal is sent when the buffer is no longer full.

```
Options: 1 (On), or 0 (Off).
Default: 1.
```

DIRECTORY Names the file that contains the user's dialing directory.

```
Default: the /usr/lib/dir file.
```

CAPTURE_KEY
Defines the control key sequence that toggles capture mode. When pressed, the CAPTURE_KEY (usually the Ctrl–B key sequence) starts or stops capturing (saving) the data that is displayed on the screen during an active connection.

```
Options: Any ASCII control character.
Default: ASCII octal 002 (STX).
```

MAINMENU_KEY
Defines the control key sequence that returns the Connected Main Menu so the user can issue a command during an active connection. The MAINMENU_KEY (usually the Ctrl–V key sequence) functions only from the connected state.

```
Options: Any ASCII control character.
Default: ASCII octal 026 (SYN).
```

PREVIOUS_KEY
Defines the control key sequence that displays the previous screen anytime during the program. The screen displayed varies, depending on the screen in use when the user presses PREVIOUS_KEY (usually the Ctrl–R key sequence).

```
Options: Any ASCII control character.
Default: ASCII octal 022 (DC2).  The ASCII control character is
mapped to the interrupt signal.
```

ATE Dialing Directory File

The ATE dialing directory file lists phone numbers that the ATE program uses to establish remote connections by modem. Users name the dialing directory file with any valid file name and place it in any directory where read and write access is owned. Edit the dialing directory file with any ASCII text editor. The default dialing directory information for the ATE program is contained in the **/usr/lib/dir** file. (See the following figure.)

```
# @(#)69   1.4  com/cmd/ate/dir, bos, bos320 4/18/91 11:00:27
#
# COMPONENT_NAME: BOS dir
#
# FUNCTIONS:
#
# ORIGINS: 27
#
# (C) COPYRIGHT International Business Machines Corp. 1985, 1989
# All Rights Reserved
# Licensed Materials - Property of IBM
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM
# Corp.
#
#
# dir -  sample dialing directory
#
#
Micom   9,555-9400 1200 7 1 2 0 0
R20     9,555-9491 1200 7 1 2 0 0
QT      9,555-8455 1200 7 1 2 0 0
Dallas1  9,555-7051 1200 8 1 0 0 0
```

Users can access the dialing directory information from within ATE by using the **directory** subcommand available in the "UNCONNECTED MAIN MENU." The following figure shows the directory information as it would appear from within the ATE program.

```
#     NAME      TELEPHONE         RATE    LEN STOP PAR ECHO LFS
        (first digits)
0     Micom     9,555-9400 1200   7  1  2  0  0
1     R20       9,555-9491 1200   7  1  2  0  0
2     QT        9,555-8455 1200   7  1  2  0  0
3     Dallas1   9,555-7051 1200   8  1  0  0  0

Enter directory entry number or e(Exit)
```

Users can have more than one dialing directory. To change the dialing directory file the ATE program uses, the user must modify the **ate.def** file in the current directory.

Note: The dialing directory file can contain up to 20 lines (one entry per line). ATE ignores subsequent lines.

Format of Dialing Directory File Entries

The dialing directory file is similar to a page in a telephone book that contains entries for the remote systems called with the ATE program. The format of a dialing directory entry is:

```
Name Phone Rate Length StopBit Parity Echo Linefeed
```

The fields must be separated by at least one space. More spaces can be used to make each entry easier to read. The fields are:

`Name` Identifies a telephone number. The name can be any combination of 20 or fewer characters. Use the _ (underscore) instead of a blank between words in a name, for example, data_bank.

`Phone`	The telephone number to be dialed. The number can be up to 40 characters. Consult the modem documentation for a list of acceptable digits and characters. For example, if a 9 must be dialed to access an outside line, include a 9, (the numeral 9 and a comma) before the telephone number as follows: 9,1112222. Although the telephone number can be up to 40 characters long, the directory subcommand displays only the first 26 characters.
`Rate`	Transmission or baud rate in bits per second (bps). Determines the number of characters transmitted per second. Select a baud rate that is compatible with the communication line being used. The following are acceptable rates: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, or 19200. For non–POSIX baud rates, setting the rate at 50 causes the ATE to use the configured baud rate set through SMIT for that device.
`Length`	Number of bits that make up a character. The entry for the Length field can be 7 or 8.
`StopBit`	Stop bits signal the end of a character. The entry for the StopBit field can be 1 or 2.
`Parity`	Checks whether a character was successfully transmitted to or from a remote system. The entry for the Parity field can be 0 (none), 1 (odd), or 2 (even).
`Echo`	Determines whether typed characters display locally. The entry for the Echo field can be 0 (off) or 1 (on).
`Linefeed`	Adds a line–feed character at the end of each line of data coming in from a remote system. The line–feed character is similar in function to the carriage–return and new–line characters. The entry for the Linefeed field can be 0 (off) or 1 (on).

Notes:

1. Changing or remapping may be necessary if control keys conflict across applications. For example, if the control keys mapped for the ATE program conflict with those in a text editor, remap the ATE control keys.
2. The ASCII control character selected may be in octal, decimal, or hexadecimal format, as follows:

octal	000 through 037. The leading zero is required.
decimal	0 through 31.
hexadecimal	0x00 through 0x1F. The leading 0x is required. The x may be uppercase or lowercase.

Example

Create an **ate.def** file that defines those characteristics to change characteristics of ATE emulation. For example, to change the RATE to 300 bps, the DEVICE to tty3, the TRANSFER mode to x (xmodem protocol), and the DIRECTORY to my.dir, create an ate.def with the following entries, in the directory running the ATE program:

```
RATE          300
DEVICE        tty3
TRANSFER      x
DIRECTORY     my.dir
```

The program uses the defined values from time the ATE program starts from that directory.

Dialing–Out with ATE

Use the following procedure to dial out of a system using ATE and a customized /**usr**/**lib**/**dir** dialing directory file.

Prerequisites

The user should verify that all of the following prerequisites and conditions are met before attempting to dial out.

- The ATE is installed on the system.
- A modem is attached, configured, and ready for use.
- The user is a member of the UUCP group (see Setting Up ATE for more information).
- The /**usr**/**lib**/**dir** dialing directory file is already customized with the correct information.
- User's present working directory (**pwd**) contains an **ate.def** file that is properly updated.
- The /**dev**/**tty** port must have its ENABLE login field in SMIT set to disable, share, or delay.

Procedure

1. Enter:

   ```
   ate
   ```

2. Enter a d for directory at the > prompt on the **UNCONNECTED MAIN MENU** screen. Refer to following figure.

```
Node: glad   UNCONNECTED MAIN MENU

COMMAND          DESCRIPTION

Connect          Make a connection.
Directory        Display a dialing directory.
Help             Get help and instructions.
Modify           Modify local settings.
Alter            Alter connection settings.
Perform          Perform an Operating System command.
Quit             Quit the program.

The following keys can be used during a connection:
ctrl b           Start or stop recording display output.
ctrl v           Display main menu to issue a command.
ctrl r           Return to a previous screen at any time.

Type the first letter of the command and press Enter.
>
```

3. Press Enter when the prompt shown in the following figure is displayed.

```
Type the file name of the directory you want to display and press Enter.
To use the current directory (usr/lib/dir) just press Enter.
```

4. To dial, enter the appropriate directory entry number from the **/usr/lib/dir** file displayed on the screen as shown in the following figure.

```
#     NAME      TELEPHONE         RATE    LEN  STOP  PAR ECHO LFS
        (first  digits)
0     Micom     9,555-9400 1200   7  1  2  0  0
1     R20       9,555-9491 1200   7  1  2  0  0
2     QT        9,555-8455 1200   7  1  2  0  0
3     Dallas1   9,555-7051 1200   8  1  0  0  0
4     Home      9,555-1324 9600   8  1  0  0  0
5     Avala     9,555-2222 9600   8  1  0  0  0


Enter directory entry number or e(Exit)
```

Transferring a File Using ATE

Use the following procedure to transfer a file from a local host to the remote system.

Prerequisites

- A connection must already be established using the **ATE** program.
- The Xmodem file transfer protocol must already exist on both the local and remote systems. On the AIX system, Xmodem is located in the /**usr**/**bin** directory.

Procedure

1. Run the following **xmodem** command on the remote system after logging in:

   `xmodem -r` *`newfile`*

   where `r` is the Xmodem flag to receive and *`newfile`* is the name of the file to be received. This name does not need to be the same as the file being transferred.
2. Press Enter.
3. The following message is displayed:

   `ate: 0828-005 The system is ready to receive file newfile. Use Ctrl-X to stop xmodem.`

   If the message is not displayed, the system may not have the **xmodem** program installed or located in its command PATH.
4. Press Ctrl–V to return to the ATE CONNECTED MAIN MENU.
5. Press `s` to send a file.
6. The following message is displayed:

Type the name of the file you wish to send and press Enter. To use the last file name (), just press Enter.

7. Enter the name and full path of the file to be transferred.
8. Press Enter.
9. ATE will display the following message and begin to transfer the file:

```
ate: 0828-024 The program is ready to send file newfile. You will
receive another message when the file transfer is complete.
ate: 0828-025 The system is sending block 1.
ate: 0828-025 The system is sending block 2.
ate: 0828-015 The file transfer is complete.
ate: 0828-040 Press Enter
```

10. Press Enter when the transfer is complete.

Receiving a File Using ATE

Use the following procedure to receive a file transferred from a remote host.

Prerequisites

- A connection must already be established using the ATE program.
- The Xmodem file transfer protocol must already exist on both the local and remote systems. On the AIX system, Xmodem is located in the /**usr**/**bin** directory.

Procedure

1. Run the following **xmodem** command on the remote system after logging in:

   ```
   xmodem -s newfile
   ```

   where `s` is the **xmodem** command to send and *`newfile`* is the name and full path of the file to be transferred.
2. Press Enter.
3. The following message is displayed:

   ```
   ate: 0828-005 The system is ready to send file newfile. Use
   ctrl-X to stop xmodem.
   ```

   If the message is not displayed, the system may not have the **xmodem** program installed or located in its command PATH.
4. Press Ctrl–V to return to the ATE CONNECTED MAIN MENU.
5. Press `r` to receive the file.
6. The following message is displayed:

   Type the name of the file you wish to store the received data in and press Enter. To use the last file name (), just press Enter.
7. Enter the name and full path of the file to be transferred.
8. Press Enter.
9. ATE will display the following message and begin to transfer the file:

   ```
   ate: 0828-020 The program is ready to receive file newfile. You
   will receive another message when the file transfer is complete.
   ate: 0828-028 The system is receiving block 1.
   ate: 0828-028 The system is receiving block 2.
   ate: 0828-040 Press Enter.
   ```

10. Press Enter when the transfer is complete.

Troubleshooting Common ATE Problems

Problem: When transferring or receiving files, the xmodem command appears to hang. A Ctrl–X corrects the problem.

Solution: Examine the Alter menu to verify that xmodem protocol (or Transfer method) is being used.

Problem: When transferring or receiving files, the file scrolls across the screen and a message is displayed stating that the transfer or receipt was complete when, in fact, it was not.

Solution: Examine the Alter menu to verify that xmodem protocol (or Transfer method) is being used.

Problem: When starting ATE, the user receives the following error:

```
ate: 0828-008 The system tried to open port /dev/tty0
but failed.  If the port name is not correct, change it
using the Alter menu.  Or, take the action indicated by
the system message shown below.

Connect: The file access permissions do not allow the
specified action.
ate: 0828-040 Press Enter.
```

Solution: The Connect: line in the error message narrows down the problem. Verify that the user attempting to run ATE is a member of the UUCP group. To check this, the user can enter *id* on the command line; **uucp** should appear in the output listing.

Problem: When attempting to make a connection with ATE, the following error is received:

```
ate: 0828-008 The system tried to open port /dev/tty0
but failed. If the port name is not correct, change it
using the Alter menu. Or, take the action indicated by
the system message shown below.

Connect: A file or directory in the path name does not
exist.
ate: 0828-040 Press Enter.
```

Solution: An incorrect or unavailable tty was selected for use by ATE. Examine the Alter screen in ATE.

Problem: The file transfers correctly, but the file size is larger than the original file.

Solution: The xmodem protocol pads the file during transfer. To avoid this, use the **tar** command to compress the file and transfer it. This is also a means of overcoming another xmodem limitation where only one file is sent at a time. The user can **tar** several files together into a single tar image and transfer it using xmodem.

Basic Network Utilities

The Basic Network Utilities (BNU) is a group of programs, directories, and files that can be used to communicate with any UNIX system on which a version of the Unix–to–Unix Copy Program (UUCP) is running.

Note: To use baud rates higher than 38400, specify a baud rate of 50 in the **/etc/uucp/Devices** file for the desired tty, then change the SMIT configuration for that tty to reflect the actual baud rate desired.

For example. to run the **cu** command on tty0 with a baud rate of 115200, use the following procedure:

1. Ensure the hardware supports the baud rate.
2. Edit **/etc/uucp/Devices** to include the following line:

```
Direct tty0 - 50 direct
```

3. Enter the **smit chtty** fast path.
4. Select tty0.
5. Change the baud rate to 115200.
6. Exit SMIT.

The following sections contain information about BNU:

- BNU Prerequisites
- How BNU Works
- National Support for BNU Commands
- BNU File and Directory Structure
- BNU Administrative Files and Directories
- BNU Security
- UUCP Login ID
- BNU Login IDs
- Adding BNU Login Shells to the Login.cfg File
- Security and the Systems and remote.unknown Files
- Security and the Permissions File
- BNU Daemons
- UUCP Configuration
- Setting Up Automatic Monitoring of BNU
- Setting Up BNU Polling Remote Systems
- Maintaining BNU
- Working with BNU Log Files
- Logging Files in the .Log and .Old Directories
- Other BNU Log Files
- Systemwide Log Files Used by BNU
- Using BNU Maintenance Commands
- Monitoring a BNU Remote Connection
- Monitoring a BNU File Transfer

- UUCP Conversation Flow Diagram
- AIX UUCP Quick Setup Guide and Information Sheet

BNU Prerequisites

Before users on the system can run BNU programs, BNU must be installed and configured.

BNU is controlled by a set of configuration files that determine whether remote systems can log in to the local system and what they can do after they log in. These configuration files must be set up according to the requirements and resources of each system.

BNU must also be maintained. To maintain BNU, users must read and remove log files periodically and check the BNU queues to ensure jobs are transferring to remote systems properly. Users must also periodically update the configuration files to reflect changes in the system or remote systems.

BNU establishes communications between computer systems on local and remote networks and is on of the Extended Services programs that can be installed with the AIX base operating system.

BNU is a version of UUCP, which was developed by AT&T and modified as part of the Berkeley Software Distribution (BSD).

BNU provides commands, processes, and a supporting database for connections to local and remote systems. Communication networks such as token–ring and Ethernet are used to connect systems on local networks. A local network can be connected to a remote system by hardwire or a telephone (modem) configuration. Commands and files can then be exchanged between the local network and the remote system.

How BNU Works

BNU uses a set of hardware connections and software programs to communicate between systems. A structure of directories and files tracks BNU activities. This structure includes a set of public directories, a group of administrative directories and files, configuration files, and lock files. Most of the directories for BNU are created during the installation process. Some of the administrative directories and files are created by various BNU programs.

With the exception of the remote login commands, BNU works as a batch system. When a user requests a job sent to a remote system, BNU stores the information needed to complete the job. This is known as *queuing* the job. At scheduled times, or when a user instructs it to do so, BNU contacts various remote systems, transfers queued work, and accepts jobs. These transfers are controlled by the configuration files on each system and those of the remote system.

National Language Support for BNU Commands

All BNU commands, except **uucpadm**, are available for National Language Support (NLS). User names need not be in ASCII characters. However, all system names must be in ASCII characters. If a user attempts to schedule a transfer or a remote command execution involving non–ASCII system names BNU returns an error message.

BNU File and Directory Structure

BNU uses a structure of directories and files to keep track of their activities. This structure includes:

- Public directories
- Configuration files
- Administrative directories and files
- Lock files.

Most of the directories for BNU are created during the installation process. Some of the administrative directories and files are created by various BNU programs as they run.

BNU Public Directories

The BNU public directory, **/var/spool/uucppublic**, stores files that have been transferred to the local system from other systems. The files wait in the public directory until users claim them with the **uupick** command. The public directory is created when BNU is installed. Within the public directory, BNU creates a subdirectory for each remote system that sends files to the local system.

BNU Configuration Files

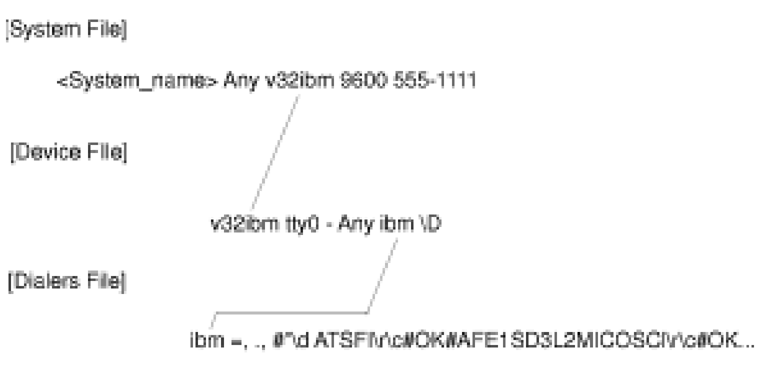
The BNU configuration files, also known as the BNU supporting database, reside in the **/etc/uucp** directory. The files must be configured specifically for your system. They are owned by the UUCP login ID and can be edited only with root authority. The configuration files contain information about:

- Accessible remote systems
- Devices for contacting the remote systems
- Times to contact the remote systems
- What the remote systems are allowed to do on the local host.

Some configuration files also specify limits on BNU activities which prevents the local host from becoming overloaded. The BNU configuration files include:

Devices	Contains information about available devices, including both modems and direct connections.
Dialcodes	Contains dialing code abbreviations, which allow the user to shorten phone numbers in the **Systems** file.
Dialers	Specifies calling command syntax for a specific modem type (dialer).
Maxuuscheds	Limits simultaneous scheduled jobs.
Maxuuxqts	Limits simultaneous remote command executions.
Permissions	Contains access permission codes. This file is the primary file for determining the security for BNU.
Poll	Specifies when the BNU program should poll remote systems to initiate tasks.
Systems	Lists accessible remote systems and information needed to contact them, including the device to use and the user name and password combinations the user needs to log in. Also specifies the times when the systems can be contacted.

The configuration files cross–reference each other when BNU is in use (see the following figure). For example the:



Correlation of BNU File

- **Systems** file contains an entry for a *Class* of device. A device of each *Class* referred to in the **Systems** file must be defined in the **Devices** file.

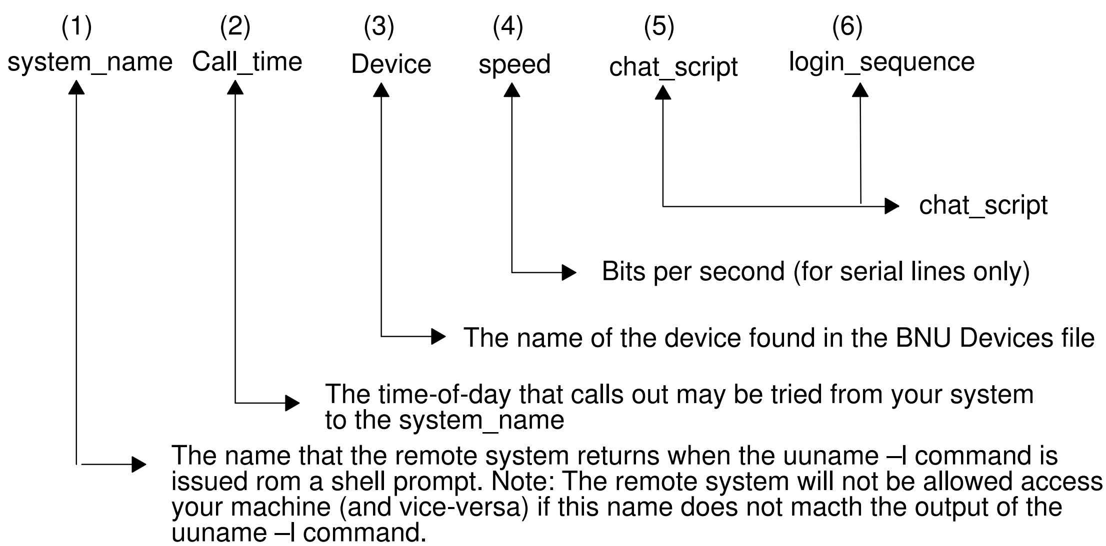
- **Devices** file contains a *Token* field that refers to entries in the **Dialers** file.
- **Poll** file contains entries for systems the host system calls. Each of these systems must be defined in the **Systems** file.

Entries in the BNU configuration files depend on the types of connections between the user's host system and each remote system. For example, special entries must be made if TCP/IP or *direct* connections are used to contact other systems. If modems are used to contact other systems, the modems must be defined in the **Dialers** file.

The **Systems**, **Devices**, and **Permissions** files must be configured on the user's host system before they can contact remote systems using BNU. Other configuration files enable the user to use BNU capabilities, such as automatic polling. Many of the configuration files must be modified periodically to reflect changes to the user's host or remote systems.

Systems File

The **/etc/uucp/Systems** file contains the primary information that is used by the **uucico** program to connect to the various systems. Users should consider this file the database of systems where their machine can connect. The following figure shows the Systems file entries and format.



The /etc/uucp/Systems File

system_name Specifies a name that the BNU caller program (**uucico**) uses to associate the remote system to the dial–up and login information from the user's **Systems** file database. The **Systems** file contains all the information needed to call a specific system, and login as a specific user.

When **uucico** calls the remote system, it will actually login (like any other user) into a special user account that will invoke as its login shell. When the remote system invokes its login shell, it will present its *uuname* (see the **uuname** command) to the system calling in.

That *uuname* must match the name in the **Systems** file of the host calling in. If they do not match, the system calling in will report a `WRONG MACHINE NAME` message and the call will fail.

Call_time Indicates the *time–of–day* that the user's system can call out to the specified remote site. Normally (for testing), this entry will read: **Any**, which means "I can call this system **Any** time of the day."

Other valid entries are:

– **Never**

– Day of the week in the format "MoTuWeThFrSaSu"

– Weekdays only, indicated by ”Wk”

– Time of day can be specified by 24 hour times 1800 (6:00pm) 0800 (8:00am), and time of day can be used in ranges by listing two times separated by a dash (–) such as: 0600–1800.

Device Specifies the name of a device that is listed in the BNU **Devices** file. There is no restriction on the name, other than it being no longer than 7 characters long and the fact that it must exist in the **Devices** file. If users do *not* wish to use the **Devices** file, they may use the dash (–) entry which indicates a valid **NULL** device.

speed Indicates the connection speed for serial line communications. If the system is using a direct connect serial line (that is, no modem just one host connected serially to the other) or modem lines, this field is required. If the system is using a **TCP** connection device type (a special **Devices** file type), this field should be a dash (–).

Chat script Indicates the BNU **Systems** file entries. The UUCP **chat** utility is what allows the user's machine to login to other systems and start a BNU connection without user intervention. This sequence is broken into *send* and *expect* fields. For every *send*, there is a corresponding *expect* (until the last one). Basically it works like this: ”I send”, ”the other end responds”; ”I send”, ”the other end responds”, etc. until I am logged in. It can be thought of in this way: ”what would I type, and what would the remote modem or system respond with, if I were going to login to that system”. Normally this sequence starts with the phone number of the remote system (if the remote system is to be connected over a modem); and is then followed by the sequence used to login to the remote system. The login sequence will look something like:

```
in:--in: username word: password
```

login_sequence

Begins: `in:--in:` which is the last 3 digits in the **login:** prompt separated by two dashes (––) used to signify:

If I don't get this within my default time out period, send the characters between the dashes (–) plus a carriage return and then wait for **in:** again. In the following example, wait for `in:`. If we don't see it, send `hello` and wait for `howdy:` in response:

```
in:-hello-howdy:
```

The next part of the chat sequence is the *username* of the remote site's UUCP account. This can be any login account the remote system administrator wants to set up to do UUCP, but the default program that it calls MUST be **/usr/sbin/uucp/uucico**.

Some sites accomplish this by creating a normal user and making the first line of the user's **.profile** call **uucico**, but it is not recommended. The default directory of this user is usually **/usr/spool/uucppublic** but it is not a requirement.

The next part of the chat sequence is `word:` which is the last five characters of the `password:` prompt. If the remote system were to say `Enter your secret code:` instead of prompting for the `password:`, this entry would be `code:` in the chat sequence.

The last entry in the chat sequence is usually the actual password of the account you will be using for UUCP on the remote system. This entry should result in a completed login and direct access to the **uucico** program. Some systems have two (or more) passwords per account, so the user would simply add more *expect – send* entries onto the chat sequence to show this.

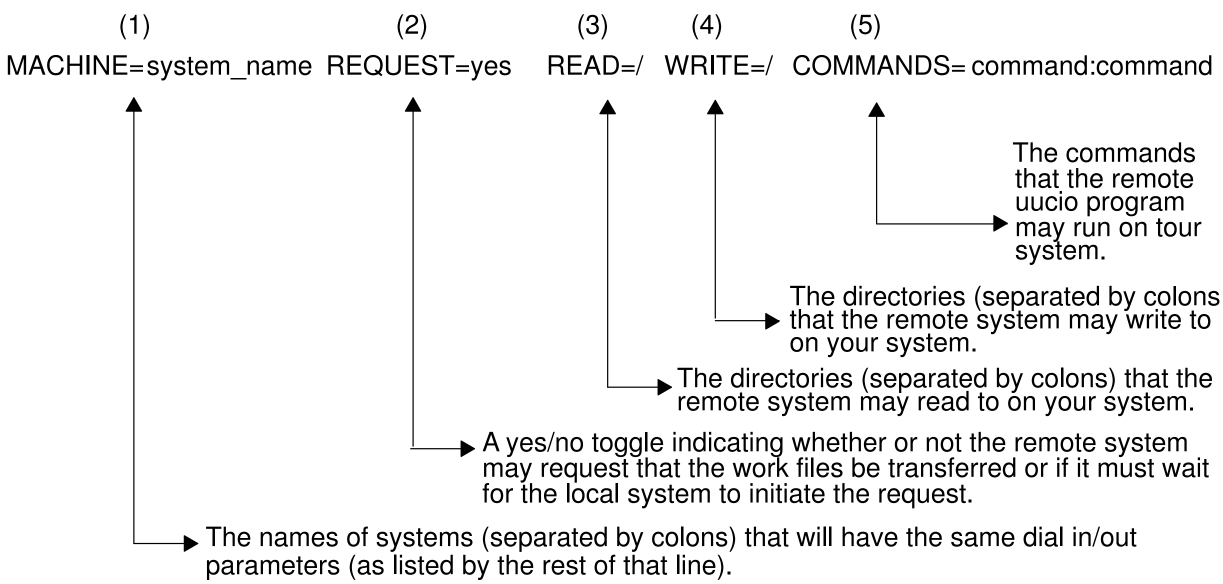
Refer to ”Security and the Systems and remote.unknown Files” , on page 11-41for more information on the **Systems** file.

Note: For quickest setup, it is probably easiest to copy the existing sample entry in the **Systems** file, and replace the system name, phone number, and account specific information (login name and password) from this default entry.

Permissions File

The **/usr/lib/uucp/Permissions** file is used to create security control over machines attempting to communicate with the local host. This file has two distinct entries each of which does a different job. It should be noted that although the entries appear together in the file, they are *not* related to one another.

The first entry in the **Permissions** file is the `MACHINE=` entry. This entry is used to establish the base permissions for a given sitename as shown in the following figure.



MACHINE= Indicates the remote systems that are allowed to access the user’s machine. This entry *must* have a corresponding *system_name* listed in the **/etc/uucp/Systems** file or it is ignored.

Several system names may be separated by colons (:) on one entry (all systems entered this way will have the same permissions, as dictated by the rest of the entry’s fields).

REQUEST= Indicates whether the users from remote systems may initiate requests to perform BNU jobs on this system. A **yes** in this field also requires a **yes** value be placed in the `REQUEST=` field for the `LOGNAME=` entry in this file (this line will be discussed later).

An answer of **yes** means that remote users may spool up jobs for this system, and request files be transferred to and from this system. For added security, it is recommended that `REQUEST=no` be used so that only this machine can initiate job requests.

READ= Indicates the directories, separated by colons (:), that the remote system may have read access to on this system. If the remote system is given access to the root directory (/), then remote users will be able to read *all* directories on this machine. The remote users will be subject only to the normal AIX file permissions

Note: Giving the remote system read access to the root directory (/) will allow it to read only those directories accessible to the UUCP account.

If the intention is to restrict access to only the **uucppublic** directory and the **tmp** directory, the `READ=` field would resemble the following:

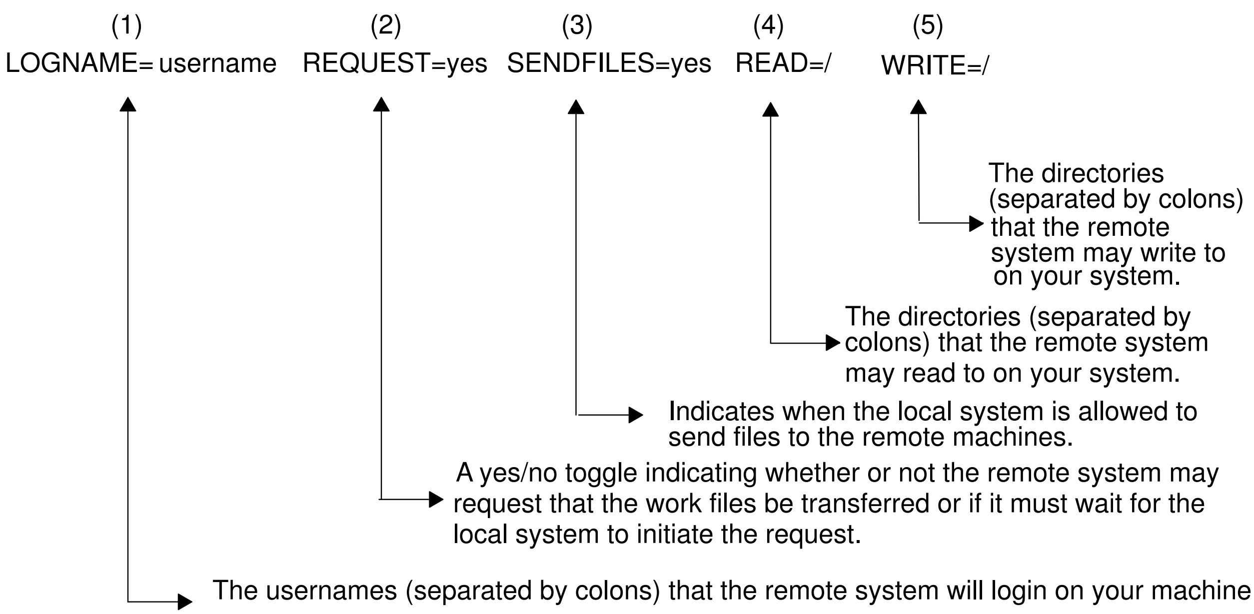
```
READ=/var/spool/uucppublic:/tmp
```

WRITE= Indicates which directories (separated by colons (:)) that the remote system may write to on this system. This works the same as the `READ=` field above, but determines write access.

COMMANDS= Indicates the command that the remote system may execute on this system, separated by colons (:). These commands can be the command names alone or the complete path to the commands. To execute the commands listed here, however, the UUCP user account on this system must have execute permission. This field does note override normal file permissions permitting execution on this system.

The second line type in the **Permissions** file is used to establish the permissions for the login account that the remote system is using. Several of the fields for this line are the same as the `MACHINE=` fields of the same name.

The primary difference is that these entries are specific to the login account that the remote system is using, while the `MACHINE=` fields are specific to the remote system's *uuname*. This field is the `LOGNAME=` entry and it has the format as shown in the following figure.



LOGNAME= Indicates the username, separated by colons (:), which remote systems will use when logging in to this machine. When a remote system logs into this machine, its username will be checked against the **Permissions** file for an entry that has `LOGNAME=` as the first characters on the line. After **uucico** checks the `LOGNAME=` line, the permissions for that remotely logged in system will be set to the the permissions defined on the rest of the `LOGNAME=` line.

REQUEST= Specifies permissions based on the login name, as opposed to the system name. It is exactly the same as the `MACHINE=` line entry of the same name.

SENDFILES= Indicates when the local system may send files to a remote system. It is only used on the `LOGNAME=` line and can be set to either **yes** or **call**. When the field is set to **yes**, the local system may send files (and/or jobs) to the remote system when either system initiates the call.

When the field is set to **call**, the local system may only send files or jobs when the local system initiates the call.

In other words with `SENDFILES=` set to **call** this system can only receive files when the remote machine calls in. With this setting the local system can send and receive files only when it initiates the call to the remote machine.

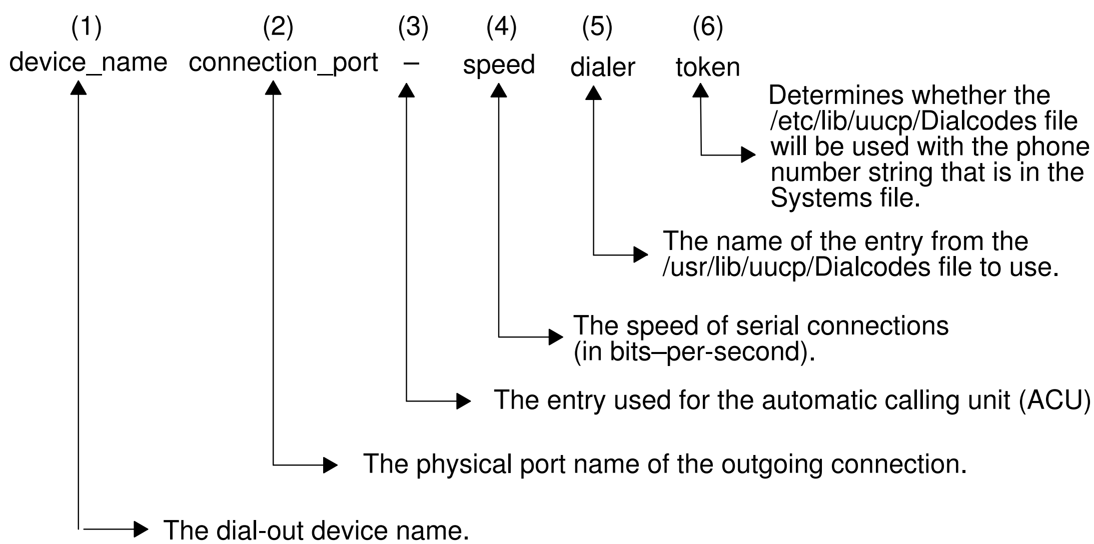
READ= Indicates the same as the `MACHINE=` field of the same name, but it sets permissions based on the login name, as opposed to the system name.

WRITE= Indicates the same as the `MACHINE=` field of the same name, but it sets permissions based on the login name, as opposed to the system name.

Devices File

The **/etc/uucp/Devices** file is used by several programs (including **uucico**) to determine which device on the system to use for a given connection attempt. For example, the **cu** program uses entries in this file for *Direct* access to the tty port, Serial Line Internet Protocol (SLIP) uses the **Devices** file for determining access to tty's for TCP/IP connections over serial line, and ECS (the IBMLINK connection facility) uses the **Devices** file to make modem connections to IBMLINK.

The basic purpose of the **Devices** file is to define the device type, location, speed, and other communication parameters necessary for the dial–out programs (it is only used for dial–out connections). The following figure shows the **Devices** file format.



device_name Specifies a *user specified* name for the device used in dial out connections. Any name may be used in this field except for the reserved words **Direct** and **TCP** which are used for direct connect serial lines (hard–wired; no modems) and connections over a network involving TCP/IP, respectively.

By standardization the *device_name* **ACU** is used for modem dial–out connections. The name ACU stands for **Automatic Calling Unit**. An ACU device used to be required for autodialing the telephone numbers for modems. The device would be attached to a second line and would dial the phone for the modem, and then attach the telephone line to the modem for communication with the remote modem/system.

connection_port

Specifies the physical device to be used by UUCP. For serial communications, the entry is the name of a tty device (for example, tty2). If a *device_name* of **TCP** is specified, the `connection_port` entry should be a dash (–).

– Early versions of UUCP used this field to indicate which tty the autodialer, or ACU, was attached to. This entry became less critical however, as modem technology progressed. Now that modems can dial without the use of ACUs, the entry is no longer needed. A dash (–) is used here only as a place–holder.

speed Indicates the speed in bits per second (bps) of a serial line connection. If the *device_name* is **TCP** this field will be a dash (–). If the user does not wish to limit the speed of this device to a specific bps rate, they may specify the speed in the **Systems** file and set this field (in the **Devices** file) to **Any**. This allows the same device to be used for several different systems that need to connect at different speeds.

dialer Indicates the type of *dialer* (as specified in the **/etc/uucp/Dialers** file) that this device will use. Several default dialers are specified in the **Dialers** file, they are:

```
hayes, penril, ventel, rixon, vadic, micom, TCP, and
direct
```

Each dialer specifies a different command set to use when attempting to dial the modem. Although several dialers are listed, the most commonly used entries are: **hayes**, **direct**, and **TCP**.

The default **hayes** entry, shown below, seems cryptic at first, but it can easily be translated using the information provided.

```
hayes =,-, "" \dAT\r\c OK \pATDT\T\r\c CONNECT
```

Escape Characters for the UUCP Dialers File

\c	No carriage return or newline.
\d	Delay (1 to 2 seconds depends on UUCP version – BNU or BSD).
\D	Telephone number (do not use Dialcodes translation).
\e	Disable echo checking.
\E	Enable echo checking (use for slower devices).
\K	Insert a Break character (BNU only).
\n	Send a newline.
\p	Pause.
\r	Send carriage return.
\s	Send a space character.
\T	Send telephone number without Dialcodes translation.
\\	Send a backslash (\) character.
=,–	Change an equal (=) or dash (–) in UUCP dial string to comma (,). Hayes uses a double quotation mark (") for *pause*.
"	Expect nothing (null string).
EOT	Send an end–of–transmission character.

System administrators who are familiar with modem configuration could use these escape characters to build a command string capable of programming even the most complicated modems. The following Dialers entry is an example of one such string used to program a Hayes–compatible, 9600 baud modem.

Note: The following lines should be combined into a single line:

```
HayesProgrm9600 =,-, "" \d\dAT\r\c OK AT&F\r\c OK
ATM1\r\c OK AT&D3\r\c OK AT&K3&C1\r\c OK ATL0E0Q2\r\c
OK
ATS0=1\r\c OK AT&W\r\c OK
```

The **direct** dialer is only a **dummy** entry, in that it does not specify any command set. Use of the direct dialer is limited to direct (non–modem) serial connections only.

The **TCP** dialer is effectively the same as the **direct** dialer, but is a keyword that TCP/IP and UUCP connections look for when making a call.

token This field indicates whether the **Devices** file should send the phone number *as it is listed* in the **Systems** file, or whether it should interpret the number through the /**usr**/**lib**/**uucp**/**Dialcodes** file first.

D Indicates that the number should be checked with the **Dialcodes** file first.

T Indicates that the number should be passed straight through to the **Dialers** file without interpretation.

The **Dialcodes** file is used to make *standardized* names for certain parts of a phone number. For example, if you made a lot of calls to a certain area code in San Francisco, you could create a Dialcodes entry that reads: `SFO 9,1415`. Which would be used in the phone number dial string like this: instead of – `9,14155551111`, you would use – `SFO5551111`. The `SFO` would be changed by the **Dialcodes** file for automatically.

It should be noted that users often include the **D** token even though the **Dialcodes** file is rarely used by UUCP administrators.

BNU Administrative Files and Directories

The BNU administrative directories and files are in subdirectories of the /**var**/**spool**/**uucp** directory. These directories and files contain two types of information:

- Data waiting to be transferred to other systems.
- Log and error information about BNU activities.

Under the **var**/**spool**/**uucp** directory, BNU creates the following seven directories:

Admin Contains the following four administrative files that retain error and log information about BNU activities:

– audit
– Foreign
– errors
– xferstats.

Corrupt Contains copies of files that cannot be processed by the BNU program.

Log Contain log files from BNU transactions.

Old Contain log files from BNU transactions.

Status Stores the last time the daemon tried to contact remote systems.

Workspace Holds temporary files that the file transport programs use internally.

Xqtdir Contains execute files with lists of commands that remote systems can run.

Directories are also created under the /**var**/**spool**/**uucp** directory which contain the system_name of each remote system the host contacts. Note that the names of all the above directories start with a period (.). This means that they cannot be found with an **ls** or **li** command unless the **–a** flag is used. When the daemon is started, it searches the /**var**/**spool**/**uucp** directory for work files and transfers the files from any directory that is not hidden. The daemon sees only the system_name directories, not the other administrative directories.

The files in the hidden directories are owned by the UUCP login ID. These files can be accessed only with root authority or with a login ID which has a user ID (UID) of 5.

BNU Lock Files

The BNU lock files are stored in the **/etc/locks** directory. When BNU uses a device to connect to a remote computer, it places a lock file for that device in the **/etc/locks** directory. When another BNU program or any other program needs the device, that program checks the **/etc/locks** directory for a lock file. If a lock file exists, the program waits until the device is available or uses another device for the communication.

In addition, the daemon places lock files for remote systems in the **/etc/locks** directory. Before contacting a remote system, the daemon checks the **/etc/locks** directory for a lock file for that system. These files prevent other instances of the daemon from establishing duplicate connections to the same remote system.

BNU Security

Because other systems contact the user's system to login, transfer files, and execute commands, BNU provides a means to establish security. BNU security enables the user to restrict what users of remote systems can do on the local host. BNU runs several daemons to complete its activities and uses administrative directories to store the files it needs. BNU also keeps a log of its own activities.

BNU security works on several levels. When the user configures BNU, they can determine:

- Who on this system has access to BNU files.
- What remote systems this system can contact.
- How users on remote systems login to this system.
- What users on the remote systems can do on this system once logged in.

UUCP Login ID

When BNU is installed, all of the configuration files, daemons, and many of the commands and shell procedures are owned by the UUCP login ID. The UUCP login ID has a user ID (UID) of 5 and a group IDD (GID) of 5. The **cron** daemon reads the **/var/spool/cron/crontabs/uucp** file to schedule automatic jobs for BNU.

Usually, logging in as user UUCP is not allowed. To change files that are owned by the UUCP login ID, log in with root authority.

Note: Allowing remote systems to log in to the local system with the UUCP login ID seriously jeopardizes the security of the local system. Remote systems logged in with the UUCP ID can display and possibly modify the local **Systems** and **Permissions** files depending on the other permissions specified in the `LOGNAME` entry. It is strongly recommended that the UUCP administrator create other BNU login IDs for remote systems and reserve the UUCP login ID for the person responsible for administering BNU on the local system. For the best security, each remote system that contacts the local system should have a unique login ID with a unique UID number.

BNU Login IDs

The startup shell for BNU login IDs is the daemon **/usr/sbin/uucp/uucico**. When remote systems call your system, they automatically start the daemon on your system. Login IDs for BNU have a UUCP group ID of 5.

Login IDs used by remote systems need passwords. To prevent security from prompting a new BNU login ID for a new password when the remote system logs in, you must set the password as soon as you create the account. To do this, use the **passwd** command followed by the **pwdadm** command.

To set a password for the login ID `nuucp`, log in as the root user and enter the following commands:

```
passwd nuucp
```

```
pwadm –f NOCHECK nuucp
```

The system prompts you for a password for the `nuucp` login ID. Completing these steps allows the remote system to log in without being immediately prompted for a new password (which the batch–oriented `nuucp` login cannot provide).

After creating the login ID for a remote system, notify that system's BNU administrator and give them the login ID and password to access your system.

Creating a BNU Administrative Login ID

A user with root authority can set up a BNU administrative login ID. This is useful if you wish to delegate BNU administration duties to a user without root authority. The BNU administrative login ID should have password security, a UID of 5, and be in a UUCP group with an ID of 5.

The login shell for the administrative login should be the /**usr**/**bin**/**sh** program (instead of the daemon). Giving the BNU administrative login a UID of 5 causes it to have the same privileges as the UUCP login ID. Thus, for security, remote systems should not be allowed to login as the BNU administrator.

Adding BNU Login Shells to the login.cfg File

User configuration stanzas in the /**etc**/**security**/**login.cfg** file provide configuration information for programs that change user attributes or add new users. There are two stanzas of this type:

```
pw_restrictions
```

```
usw
```

Note: In AIX Version 4 the `pw_restrictions` stanza is no longer available in the **login.cfg** file.

The shells attribute in the `usw` stanza defines the valid shells on the system. The value is a list of comma–separated full path names such as:

```
/usr/bin/sh,/usr/bin/bsh,/usr/bin/csh,/usr/bin/ksh
```

Before using the System Management Interface Tool (SMIT) to add a new BNU user, add the program name /**usr**/**sbin**/**uucp**/**uucico** to the `usw` shells stanza. The new program name should be separated from the last entry by a comma and no blanks.

```
/usr/bin/sh,/usr/bin/bsh,/usr/bin/csh,/usr/bin/ksh,/usr/sbin/uucp
/uucico
```

Note: SMIT will fail when specifying /**usr**/**sbin**/**uucp**/**uucico** as a user's login shell if the program name is not added to the **login.cfg** file located in the /**etc**/**security** directory.

Security and the Systems and remote.unknown Files

On most BNU systems, only remote systems listed in the /**etc**/**uucp**/**Systems** file can log in to the local system. The /**usr**/**sbin**/**uucp**/**remote.unknown** script is executed whenever an unknown system attempts to call the local system. This script refuses to let the unknown system log in, and makes an entry in the /**var**/**spool**/**uucp**/**.Admin**/**Foreign** file recording the time of the login attempt.

With root authority, or as a BNU administrator, you can modify the **remote.unknown** shell procedure to log more information about the remote system or to store the information in a different file. For example, you can modify the shell procedure to send mail to the BNU administrator whenever an unknown system tries to log in.

By taking away execute permissions on the **remote.unknown** shell procedure you enable unknown machines to log in. In this case, you should add a `MACHINE=OTHER` entry to the **/etc/uucp/Permissions** file to establish permissions for the unknown machines.

Your system can contact only remote systems listed in the **Systems** file. This prevents users on your system from contacting unknown systems.

Security and the Permissions File

The **/etc/uucp/Permissions** file determines:

- Remote login user names for logging in to the local system.
- Approved commands and privileges for remote systems logging in to the local system.

The **/etc/uucp/Permissions** file contains two type of entries:

`MACHINE`	Defines machine names and the privileges associated with them. `MACHINE` entries take effect when the local system contacts a remote system.
`LOGNAME`	Defines login names and the privileges associated with them. `LOGNAME` entries take effect when a remote system calls the local system and attempts to log in.

Options in the **Permissions** file enable you to establish various levels of security for each remote system. For example, if many remote systems share one login ID on the local system, use the `VALIDATE` options to require each remote system to use a unique login ID. The `SENDFILES`, `REQUEST`, and `CALLBACK` options specify which system has control, keeping the local system in control of transactions if necessary.

The `READ`, `WRITE`, `NOREAD`, and `NOWRITE` options define access to specific directories on the local system. These options also control where on your system remote users can place data. The `COMMANDS` option limits the number of commands users on remote systems can execute on the local system. The `COMMANDS=ALL` option allows total privileges to remote systems but seriously jeopardizes the security of your system; it should therefore be used only when necessary.

BNU Daemons

The BNU software includes four daemons stored in the **/usr/sbin/uucp** directory:

uucico	Facilitates file transfers.
uusched	Facilitates work request scheduling of files queued in the local spooling directory.
uuxqt	Facilitates remote command executions.
uucpd	Facilitates communications using TCP/IP.

uucico, **uusched**, and **uuxqt** are started by the **cron** daemon according to a schedule set by the BNU administrator. With root authority, you can also start these daemons manually. The **uucpd** daemon should be started by the TCP/IP **inetd** daemon.

Using the uucico Daemon

The **uucico** daemon transports the files required to send data from one system to another. The **uucp** and **uux** commands start the daemon to transfer command, data, and execute files to the designated system. The **uucico** daemon is also started periodically by the BNU scheduler, the **uusched** daemon. When started by the **uusched** daemon, the daemon attempts to contact other systems and execute the instructions in the command files.

How the Daemon Process Begins

To run the instructions in the command files, the **uucico** daemon first checks the **/etc/uucp/Systems** file for the system to be called. The daemon then checks the **Systems** file entry for a valid time to call. If the time is valid, the daemon checks the *Type* and *Class* fields and accesses the **/etc/uucp/Devices** file for a device that matches.

After finding a device, the **uucico** daemon checks the **/etc/locks** directory for a lock file for the device. If one exists, the daemon checks for another device of the requested type and speed.

When no device is available, the **uucico** daemon returns to the **Systems** file for another entry for the remote system. If one exists, the daemon repeats the process of searching for a device. If another entry is not found, the daemon makes an entry in the **/var/spool/uucp/.Status/***SystemName* file for that remote system and goes on to the next request. The command file remains in the queue. The daemon attempts the transfer again at a later time. The later attempt is called a *retry*.

When the Daemon Reaches the Remote System

When the **uucico** daemon reaches the remote system, it uses the instructions in the **Systems** file to log in. This causes an instance of the daemon to be invoked on the remote system as well.

The two daemons, one on each system, work together to make the transfer. The daemon on the calling system controls the link, specifying the requests to be performed. The daemon on the remote system checks the local permissions for whether they allow the request to be performed. If so, the file transfer starts.

After the daemon on the calling system has finished transferring all requests it has for the remote system, it sends a hangup request. When the remote daemon has transactions to send to the calling system, it denies the hangup request, and the two daemons reverse roles.

Note: Either the **/etc/uucp/Permissions** file on the local system or the **/etc/uucp/Permissions** file on the remote system can forbid the daemons to reverse roles, In this case, the remote system must wait to transfer files until it calls the local system.

When nothing is left to be transferred in either direction, the two daemons hang up. At this point, the **uuxqt** daemon is called to execute remote command requests.

Throughout the transfer process, the daemons on both systems log messages in the BNU log and error files.

Using the uusched Daemon

The **uusched** daemon schedules the transfer of files that are queued in the spooling directory on the local system. The spooling directory is **/var/spool/uucppublic**. When the **uusched** daemon is invoked, it scans the spooling directory for command files, then randomizes the files and starts the daemon. The daemon transfers the files.

Using the uuxqt Daemon

When a user issues the **uux** command to run a specified command on a designated system, the **uuxqt** daemon executes the command. After creating the necessary files, the **uux** command starts the daemon, which transfers those files to the public spooling directory on the specified system.

The **uuxqt** daemon periodically searches the spooling directory for command–execution requests on every connected system. When it locates such a request, the **uuxqt** daemon checks for necessary files and permissions. Then, if permitted, the daemon executes the specified command.

Using the uucpd Daemon

The **uucpd** daemon must be runnable on the remote system before BNU can establish communications with a remote computer with Transmission Control Protocol/Internet Protocol (TCP/IP). The **uucpd** daemon is a subserver of the TCP/IP **inetd** daemon and is started by the **inetd** daemon.

By default, the **inetd** daemon is configured to start the **uucpd** daemon. However, if this has been changed on your system, you may need to reconfigure the **inetd** daemon to start the **uucpd** daemon.

UUCP Configuration

Prerequisites

- BNU must be installed on the system. You must have root authority to edit the BNU configuration files.
- If you are using direct connections for BNU communications, the appropriate hard–wired connections between your system and the remote systems must be set up.
- If modems are used for the BNU connection, they must be installed and configured.
- If the connection uses TCP/IP, then TCP/IP must be running between your system and the remote site.

Configuring UUCP

To configure UUCP on your system, perform the following steps:

1. Edit the **/etc/security/login.cfg** file making the following changes:
   - Find the `shells =` variable in the `usw:` stanza.
   - Add `,/usr/lib/uucp/uucico` after the last entry.

   User configuration stanzas in the **/etc/security/login.cfg** file provide configuration information for programs that change user attributes or add new users. This entry must be made before attempting to add or change a user's login shell to `uucico`.

   ALTERNATE METHOD:

   The UUCP login herald may also be customized for a specific tty port. The system's default herald contains control sequences which may cause a **uucico** process to terminate login attempt. UUCP errors such as `Enough already` may indicate such a problem.

   To avoid this situation the administrator should make the herald as short as possible. Use the following steps to accomplish this:
   - In the **/etc/security/login.cfg** file, comment out the *default* stanza and its herald by placing an (*) in the leftmost column of each line.
   - Add the following entries before the *default* stanza, substituting the correct tty number in place of `tty0`:

   ```
   /dev/tty0: herald = "\nuucp login: "
   ```

2. Create a UUCP administrator account. Edit the **/etc/passwd** file adding the following line after the **uucp** entry:

   ```
   uucpadm:!:5:5::/usr/lib/uucp:/bin/ksh
   ```

   This creates a user with the same group and user ID as UUCP which is helpful for BNU administration and debugging.

3. Add other machine logins. You have the option of maintaining separate logins, or having one login for all UUCP connections. The rule of thumb here is that if you need to maintain complete control over access by each individual machine, you must create

separate login IDs as well as combine the `MACHINE` and `LOGNAME` entries in the **Permissions** file. A few examples are shown below:

```
Umicrtk:!:105:5:micrtk
uucp:/usr/spool/uucppublic:/usr/lib/uucp/uucico
Ufloydl:!:106:5:floydl
uucp:/usr/spool/uucppublic:/usr/lib/uucp/uucico
Uicus:!:107:5:icus
uucp:/usr/spool/uucppublic:/usr/lib/uucp/uucico
Urisctkr:!:108:5::/usr/spool/uucppublic:/usr/lib/uucp/uucico
```

If you want to have one set of permissions and do not want to maintain separate control for any of the UUCP connections, you can have a single login for all the machines. For example:

`nuucp:!:6:5::/usr/spool/uucppublic:/usr/lib/uucp/uucico` **Field requirements:**

The user ID (the third colon–separated field) must be unique to avoid would be a security risk. The group ID (the fourth colon–separated field) *must* be a 5, which is the same group as UUCP. You can change the home directory (the sixth field) to any valid directory, but the login shell (the seventh field) *must* be **/usr/lib/uucp/uucico**.

4. Make sure that the **/etc/group** file contains the new users added in step 3. For example:

   ```
   uucp:!:5:uucp,uucpadm,nuucp,luucp,Umicrtk,Ufloydl,Uicus,
   Urisctakr,ayalad
   ```

   Take this time to add any other users to the UUCP group who will be using modems to dial out with programs other than the **cu** command. The user name `ayalad` was added at the end of the UUCP group as an example. The user name `luucp` is added as a login name for systems being granted *limited* machine access.

5. After editing these files as **root**, set up a password for the new users with the command:

   ```
   passwd UserName
   ```

6. After changing the passwords, it is necessary to edit the `flags =` entry in the **/etc/security/passwd** file for each new UUCP user. Locate the following `flags` entry for each UUCP user:

   ```
   flags = ADMCHG
   ```

   Change it to:

   ```
   flags =
   ```

   Otherwise, when the remote **uucico** logs in, it will be prompted to enter a new password, which it cannot do. Hence the login will fail.

7. Activate **cron**. After logging in as **uucpad**, run the following command to read the current crontab for UUCP into a temporary file:

   ```
   crontab –l > /tmp/cron.uucp
   ```

   Next, edit **/tmp/cron.uucp** to uncomment entries. They should look like following:

   ```
   20,50 * * * /bin/bsh –c ”/usr/lib/uucp/uudemon.poll >/dev/null”
   25,55 * * * /bin/bsh –c ”/usr/lib/uucp/uudemon.hour> /dev/null”
   45,23 * * /bin/bsh –c ”/usr/lib/uucp/uudemon.cleanu > /dev/null”
   48,8,12,16 * * /bin/bsh –c ”/usr/lib/uucp/uudemon.admin >
   /dev/null”
   ```

   Changed the entries to suit your needs. Afterwards, read in the edited version into UUCP's crontab with this command:

   ```
   crontab /tmp/cron.uucp
   ```

8. Verify that your changes took effect by running the command:

   ```
   crontab –l
   ```

Refer to *AIX 4.3 System Management Guide: Operating System and Devices* for more information on **cron**.

9. Set up the BNU data files: **Systems**, **Devices**, **Permissions**, and **Dialers**. You can use the **/usr/sbin/uucp/uucpadm** command to initially set up the files, and then edit them to suit your exact needs.

10. Use the **uucheck** command to verify that everything is in place:

```
/usr/sbin/uucp/uucheck -v
```

The **uucheck** command verifies that the directories, programs, and support files are set up properly and that the **Permissions** file entries are consistent. Correct any errors reported by the **uucheck** command.

BNU Example

The following BNU file entries were made using the examples shown in this chapter. An optional entry is included to show how to program a modem (a Telebit T3000 in this case).

The **Systems** file:

```
Umicrtk Any ACU 9600 9,15551111 ”” \d\r in:--in: nuucp word:
nuucp
Ufloydl Any ACU 9600 9,15551234 ”” \d\r in:--in: nuucp word:
nuucp
Uicus Any ACU 9600 9,15554321 ”” \d\r in:--in: luucp word: luucp
Urisctkr Any ACU 2400 9,15559876 ””\d\r in:--in: luucp word:
luucp
telebit Nvr TELEPROG 19200
```

The **Devices** file:

```
ACU tty1 - 9600 hayes \D
ACU tty1 - 2400 hayes \D
TELEPROG tty1 - 19200 TelebitProgram
```

The **Dialers** file:

```
hayes =, -, ”” \dAT\r\c OK \pATDT\T\r\c CONNECT
# NOTE: The following 4 lines should be made into one long line:
TelebitProgram =,-, ”” \dAT&F\r\c OK
ats0=1s2=255s7=60s11=50s41=2s45=255s51=252s63=1s58=2s64=1\r\c OK
ATs69=2s105=0s111=30s255=0M0&C1Q2&D3&Q0&R3&S1&T5\r\c OK
ATE0X12&W\r\c OK 00
```

The **Permissions** file:

```
MACHINE=Umicrtk:Ufloydl REQUEST=yes READ=/ WRITE=/ COMMANDS=ALL
MACHINE=Uicus:Urisctkr REQUEST=yes READ=/ WRITE=/  \
COMMANDS=vi:ls:rm
LOGNAME=nuucp REQUEST=yes SENDFILES=yes READ=/ WRITE=/
LOGNAME=luucp REQUEST=yes SENDFILES=yes  \
READ=/usr/spool/uucppublic:/tmp  \
WRITE=/usr/spool/uucppublic:/tmp
```

Setting Up Automatic Monitoring of BNU

Prerequisites

- BNU users and files must be correctly configured.
- Root authority is required to edit the **/var/spool/cron/crontabs/uucp** file.

Configuration

BNU uses the **cron** daemon to start BNU daemons and to monitor BNU activity. The **cron** daemon reads the **/var/spool/cron/crontabs/uucp** file for instructions about when to start BNU procedures.

1. Log in as a user with root authority.
2. Using an ASCII text editor, edit the **/var/spool/cron/crontabs/uucp** file.
3. Uncomment the lines for BNU maintenance procedures, **uudemon.admin** and **uudemon.cleanup**. You can change the times these procedures are run if the system needs maintenance at more or less frequent intervals. It is best to run the **uudemon.admin** command at least once a day and the **uudemon.cleanup** command at least once a week.
4. You can use the **crontabs/uucp** file to schedule other BNU maintenance commands, such as the **uulog**, **uuclean**, or **uucleanup** commands. In addition, you can use the **/var/spool/cron/crontabs/uucp** file to instruct the **cron** daemon to start the **uucico**, **uuxqt**, or **uusched** daemons at specific times.

Setting Up BNU Polling of Remote Systems

Prerequisites

- BNU users and files must be correctly configured.
- Root authority is required to edit the **/var/spool/cron/crontabs/uucp** file and the **/etc/uucp/Poll** file.

Configuration

To enable BNU to poll remote systems for jobs, list the systems in the **/etc/uucp/Poll** file. In addition, run the **uudemon.hour** and **uudemon. poll** commands periodically.

1. Decide which remote systems to automatically poll. Decide how often you want to poll each one. Specify times for each system with the **Poll** file as seldom as once a day or as often as you wish.
2. Log in as a user with root authority.
3. Using an ASCII text editor or the **uucpadm** command, edit the **Poll** file. Add an entry for each system your system will poll. Any systems listed in the **Poll** file *must* also be listed in the **/etc/uucp/Systems** file.
4. Using an ASCII text editor, edit the **/var/spool/cron/crontabs/uucp** file. Remove the comment characters (#) from the lines that run the **uudemon.hour** and **uudemon.poll** commands. You can change the times these commands are run; however, be sure to schedule the **uudemon.poll** command approximately five minutes *before* you schedule the **uudemon.hour** command.

BNU will now automatically poll the systems listed in the **Poll** file at the times you have specified.

Maintaining BNU

BNU must be maintained to work properly on your system. The following checks should be performed in order to maintain the Basic Network Utilities:

- Read and remove log files periodically.
- Use the **uuq** and **uustat** commands to check the BNU queues to ensure jobs are transferring to remote systems properly.
- Schedule automatic commands which poll remote systems for jobs, return unsent files to user, and send you periodic messages about BNU status.
- Periodically update the configuration files to reflect changes in your system.

In addition, occasionally check with administrators of remote systems to keep up with changes on their systems that may affect your configuration. For example, if the supervisor of system `venus` changes your system's password, you will need to put the new password in the **/etc/uucp/Systems** file before your system can log in to system `venus`.

Working with BNU Log Files

BNU creates log files and error files to track its own activities. These files must be checked and removed periodically to keep them from filling the storage space on your system. BNU provides several commands for use in cleaning log files:

- **uulog**
- **uuclean**
- **uucleanup**
- **uudemon.cleanu**

Run these commands manually or use entries in the **/var/spool/cron/crontabs/uucp** files to run the commands by the **cron** daemon.

Logging Files in the .Log and .Old Directories

BNU creates individual log files in the **/var/spool/uucp/.Log** directory. A log file is created for each accessible remote system, using the **uucp**, **uuto**, or **uux** command. BNU places status information about each transaction in the appropriate log file each time someone on the system uses BNU. When more than one BNU process is running the system cannot access the log file. Instead, it places the status information in a separate file with a **.Log** prefix.

The **uulog** command displays a summary of **uucp** or **uux** requests, by user or by system. The uulog command displays the files. However, you can also have BNU automatically combine the log files into a primary log file. This is called *compacting* the log files, and can be done with the **/usr/lib/uucp/uudemon.cleanu** command (usually run by the **cron** daemon).

The **cron** daemon runs the **uudemon.cleanu** command, which combines the uucico and uuxqt log files on the local system and stores them in the **/var/spool/uucp/.Old** directory. By default, the **uudemon.cleanu** command saves log files that are two days old.

If storage space is a problem, consider reducing the number of days that files are kept. To track BNU transactions over a longer period of time, consider increasing the number of days that files are kept. To change the default time for saving log files, modify the shell procedure for the **uudemin.cleanu** command. This script is stored in the **/usr/sbin/uucp** directory and can be modified with root authority.

Other BNU Log Files

BNU also collects information and stores it in the **/var/spool/uucp/.Admin** directory. This directory contains the **errors**, **xferstats**, **Foreign**, and **audit** files. These files must be checked and removed occasionally to save storage space. BNU creates each file when it is needed.

When another system contacts your system with the **uucico** daemon's debugging mode on, it invokes the **uucico** daemon on your system with debugging turned on. The debugging messages generated by the daemon on the local system are stored in the **audit** file. This file can get quite large. Check and remove the **audit** file often.

The **errors** file records errors encountered by the **uucico** daemon. Checking this file can help you correct problems such as incorrect permissions on BNU work files.

The **xferstats** file contains information about the status of every file transfer. Check and remove this file occasionally.

The **Foreign** file is important to the security of your system. Whenever an unknown system attempts to log in to the local system, BNU calls the **remote.unknown** shell procedure. This shell procedure logs the attempt in the **Foreign** file. The **Foreign** file contains the names of the systems that have attempted to call the local system and been refused. If a system has been attempting frequent calls, use this information when considering whether to allow that system access.

Systemwide Log Files Used by BNU

Because many BNU processes need root authority to complete their tasks, BNU creates frequent entries in the /**var**/**spool**/**sulog** log file. Similarly, using the **cron** daemon to schedule BNU tasks creates multiple entries in the /**var**/**spool**/**cron**/**log** file. When using BNU, check and clean these files.

Using BNU Maintenance Commands

The Basic Networking Utilities contain several commands for monitoring BNU activities and cleaning BNU directories and files.

Cleanup Commands

BNU contains three commands that clean directories and remove files that have not been sent:

- **uuclean**

  Deletes all files older than a specified number of hours, from the BNU administrative directories. Use the **uuclean** command to specify a directory to be cleaned or a type of file to be deleted. You can also instruct the command to notify the owners of the deleted files. The **uuclean** command is the Berkeley equivalent of the **uucleanup** command.

- **uucleanup**

  Performs functions similar to the **uuclean** command. However, the **uucleanup** command checks the age of files based on days rather than hours. Use the **uucleanup** command to send a warning message to users whose files have not been transferred, notifying them that the files are still in the queue. The **uucleanup** command also removes files relating to a specified remote system.

- **uudemon.cleanu**

  A shell procedure that issues the **uulog** and **uucleanup** commands to compress the BNU log files and remove log and work files over three days old. The **uudemon.cleanu** command is run by the **cron** daemon.

Status–Checking Commands

BNU also provides five commands for checking the status of transfers and log files:

- **uuq**

  Displays jobs currently in the BNU job queue. Use the **uuq** command to display the status of a specified job or of all jobs. With root authority, you can use the **uuq** command to delete a job from the queue.

- **uustat**

  Provides information similar to that provided by the **uuq** command, but in a different format. Use the **uustat** to check the status of jobs and delete jobs you own. With root authority, you can also delete jobs belonging to other users.

- **uulog**

  Displays a summary of **uucp** or **uux** requests, by user or by system. The **uulog** command displays the file names. See ”Working with BNU Log Files”, on page 11-48.

- **uupoll**

  Forces a poll of a remote system. This is helpful when work for that system is waiting in the queue and needs to be transferred, before the system is scheduled to be called automatically.

- **uusnap**

  Displays a very brief summary of BNU status. For each remote system, this command shows the number of files awaiting transfer. However, it does not show how long they have been waiting. The **uusnap** command is the Berkeley equivalent of the **uustat** command.

Shell Procedures

BNU is delivered with two shell procedures used for maintenance:

- **uudemon.cleanu**

  Is discussed under ”Cleanup Commands” , on page 11-49

- **uudemon.admin**

  Issues the **uustat** command. The uustat command reports the status of BNU jobs. It sends the results to the UUCP login ID as mail. You can modify the **uudemon.admin** shell procedure to send the mail elsewhere, or use a mail program to reroute all mail for the UUCP login ID to the user responsible for BNU administration.

These shell procedures are stored in the **/usr/sbin/uucp** directory. If you wish to change what they do, copy the procedures and modify the copy. Run the procedures from the command line or schedule them to be run by the **cron** daemon.

To automatically run the **uudemon.cleanu** and **uudemon.admin** commands, remove the comment characters (#) from the beginning of the relevant lines in the **/var/spool/cron/crontabs/uucp** file.

Monitoring a BNU Remote Connection

Prerequisites

- The BNU program must be installed on the system.
- A link (hard–wired, modem, or TCP/IP) must be set up between the local and remote systems.
- The BNU configuration files, including the **Systems** file, **Permissions** file, **Devices** file, and **Dialers** file, must be set up for communications between the local and remote systems.

Procedure

The **Uutry** command can help monitor the **uucico** daemon process if users at the local site report file–transfer problems.

1. Issue the **uustat** command to determine the status of all the transfer jobs in the current queue as follows:

   ```
   uustat –q
   ```

   A status report similar to the following is displayed:

   ```
   venus 3C (2) 05/09–11:02 CAN'T ACCESS DEVICE
   hera 1C 05/09–11:12 SUCCESSFUL
   merlin 2C 5/09–10:54 NO DEVICES AVAILABLE
   ```

   This report indicates that three command (C.*) files intended for remote system `venus` have been in the queue for two days. There could be several reasons for this delay. For example, perhaps system `venus` has been shut down for maintenance or the modem has been turned off.

2. Before beginning more extensive troubleshooting activities, issue the **Uutry** command as follows to determine whether the local system can contact the remote system `venus` now:

   ```
   /usr/sbin/uucp/Uutry –r venus
   ```

   This command starts the **uucico** daemon with a moderate amount of debugging and the instruction to override the default retry time. The **Uutry** command directs the debugging output to a temporary file, **/tmp/venus**.

3. If the local system succeeds in establishing a connection to system `venus`, the debugging output contains a good deal of information. However, the final line in this script, which follows, is the most important:

   ```
   Conversation Complete: Status SUCCEEDED”
   ```

   If the connection is successful, assume that the temporary file transfer problems are now resolved. Issue the **uustat** command again to make certain that the files in the spooling directory have been transferred successfully to the remote system. If they have not, use the steps in Monitoring a BNU File Transfer to check for file transfer problems between the local system and the remote system.

4. If the local system cannot contact the remote system, the debugging output generated by the **Uutry** command contains the following type of information (the exact form of the output may vary):

   ```
   mchFind called (venus)
   conn (venus)
   getto ret –1
   Call Failed: CAN’T ACCESS DEVICE
   exit code 101
   Conversation Complete: Status FAILED
   ```

   Check the physical connections between the local and remote systems. Make sure that the remote computer is turned on and that all cables are properly connected, that the ports are enabled or disabled (as needed) on both systems, and that the modems (if any) are working.

   If the physical connections are correct and secure, then verify all the relevant configuration files on both the local and remote systems, including the following:

   – Verify that entries in the **Devices**, **Systems**, and **Permissions** files (located in the **/etc/uucp** directory) are correct on both systems.

   – If a modem is being used, make sure the **/etc/uucp/Dialers** file contains the proper entry. If you are using dial–code abbreviations, be sure the abbreviations are defined in the **/etc/uucp/Dialcodes** file.

   – If a TCP/IP connection is being used, make sure that the **uucpd** daemon can be run on the remote system and that the configuration files contain the correct TCP/IP entries.

5. Once the physical connections and configuration files have been checked, issue the **Uutry** command again. If the debugging output still reports that the connection failed, it may be necessary to contact a local service representative for additional assistance. Save the debugging output produced by the **Uutry** command. This may prove helpful in diagnosing the problem.

Monitoring a BNU File Transfer

Use this procedure to monitor a file transfer to a remote system. Monitoring a file transfer is useful when transmission of files to the remote system is failing for unknown reasons. The debugging information produced by the **uucico** daemon (called by the **Uutry** command) can help determine the point of failure.

Prerequisites

- The BNU program must be installed and configured on the system.
- A connection to the remote system must already exist.

Procedure

Use the following procedure to monitor file transfers.

1. Prepare a file for transfer using the **uucp** command with the **–r** flag, by entering:

   ```
   uucp -r test1 venus!~/test2
   ```

   The **–r** flag instructs the BNU program to place the **test1** file in the queue but *not* to start the **uucico** daemon.

2. Issue the **Uutry** command with the **–r** flag to start the **uucico** daemon with debugging turned on by entering:

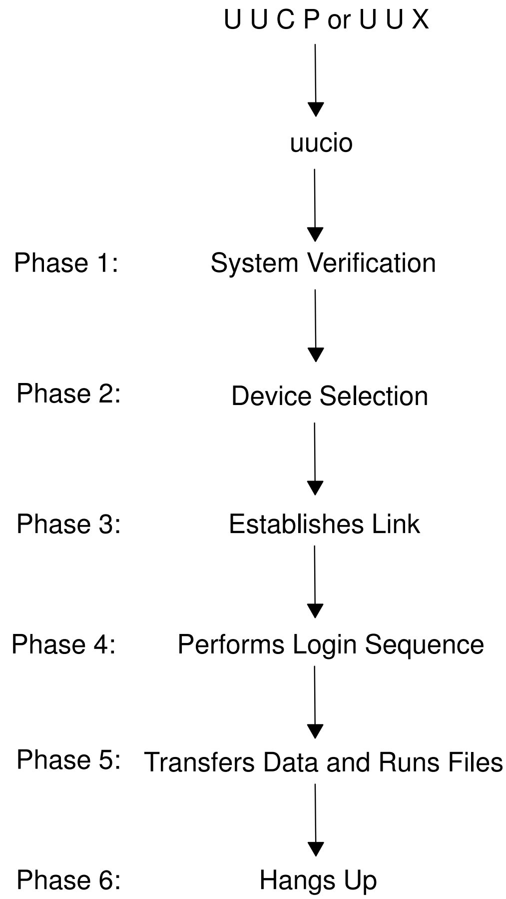
   ```
   /usr/sbin/uucp/Uutry -r venus
   ```

   This instructs the **uucico** daemon to contact the remote system `venus` overriding the default retry time. The daemon contacts system `venus`, logs in, and transfers the file while the **Uutry** command produces debugging output that monitors the uucico process. Press the interrupt key sequence to stop the debugging output and return to the command prompt.

   The **Uutry** command also stores the debugging output in the /**tmp**/*SystemName* file.

UUCP Conversation Flow Diagram

UUCP error messages can be linked to a specific phase in the conversation flow. Use the following diagram and the message descriptions on the following pages to help diagnose your UUCP problems. Some of the messages described may not be sent from AIX's UUCP (BNU) version but are included in case another UUCP version is in use on a different system platform.



PHASE 1 Status Messages

Assert Error The system is having problems. Check the error report for possible causes (for example, `errpt -a | pg`).

System not in Systems If you supply a remote system name that is not found in the **Systems** file, this status message is created. UUCP will terminate. Use the **uuname** command to check the system name again.

Wrong time to call The **Systems** file has restrictions on times to allow outgoing calls. UUCP will keep trying until the time is right. Check the **Systems** file.

Callback required The network has restricted usage either for security or economic reasons and access is denied at this time.

Cannot call or No call UUCP recently tried to call the remote system and failed. It will not immediately try again. Can also be caused by an old system status file being left around thus keeping uucico from trying again.

PHASE 2 Status Messages

Dialer Script Failed
Your **Dialers** file script did not complete successfully.

No Device Available or Can't Access Device
The modem or the outgoing phone line from your system is busy. Check for an error in the device entry of the **Systems** file. Also, check the **Devices** and **Dialers** files to be sure logical devices have physical devices associated with them. Is the device in use by some other program? Check the **/etc/locks** directory for lock on port. If a lock file exists (for example, `LCK..TTY0`), remove it (for example, `rm /etc/locks/LCK..TTY0`). Also check permissions on the port.

Dial Failed or Failed (call to system)
Appears when your system dials another successfully but the other system does not answer. It may also indicate a problem in the **Devices** file.

Enter the `uucico -r1 -x6 -s` *`system name`* command. It could be that UUCP is expecting some string that it is not receiving. Make connection by hand to find out what needs to be incorporated into the **Systems** file entry to satisfy the request. Please keep timing in mind; perhaps some delays are needed. This could also mean that the port is busy, you dialed a wrong number, or UUCP lost ownership of port.

OK or Auto Dial
These are information messages only.

PHASE 3 Status Messages

Handshake Failed (LCK)
The device is being used by someone else; the process could not create the **LCK** file. Sometimes **LCK** files must be manually removed by the administrator. After a number of retries, see your system administrator. See if another process has control of the port (for example, another uucico).

Login Failed or Timeout
Login failed due to a bad connection or slow machine. Similarly, you might see Timeout, which means the remote system did not respond within a set period of time. This could also indicate a problem with the chat script.

Succeeded (Call to System) or BNU
These are informational messages only and do not indicate a problem.

PHASE 4 Status Messages

Startup Failed or Remote reject after login
After login, uucico is started on the remote system. If there is a problem in initiating a conversation between the two systems, these messages are created. You may also have logged in using the wrong UUCP account. Initial **shere** handshake failed.

Wrong machine name
You called the wrong machine or the machine's name was changed.

Bad login/machine combination
Login for remote system failed. Problem could be a wrong phone number, login/password, or chat script.

Remote has a LCK file for me
Remote was trying to call at same time and your request will fail temporarily.

You are unknown to me
System or **Permission** file may have wrong permission or owner or may not have correct system name within. Use **uuname** on both machines.

OK or Talking These are informational messages only and do not indicate a problem.

Get the login or password prompt but it is in all capital letters

Modem may be in echo mode (E1 on Hayes compatibles). This causes the modem to echo back, or send, a RING to your system when an incoming call is received. The getty process receives the string and accordingly changes the `login:` or `password:` into all caps.

To fix this problem, change echo mode on modem to off (**ATE0** for Hayes compatibles).

Note: Keep in mind that once this change is made, you should use *ATE1* in your **Dialers** file chat script or you will not get the expected **OK** back from the modem.

Remote port set for ”delay or ”getty –r and chat script expects key input

Ports set for delay are expecting one or more carriage returns before proceeding with the login.

To fix this problem, try beginning the chat script on dialing system to the following:

```
"" \r\d\r\d\r\d\r in:--in: ...
```

Interpreted, the above reads to expect nothing and send return, delay, return, delay, return, delay, return.

PHASE 5 Status Messages

Alarm **uucico** is having trouble with the connection. Either the connection is bad or xon/xoff is set to yes on modem.

Remote access to path/file denied or copy (failed)

Permission problem; check file/path permissions. Try giving read, write, and execute permission to the receiving directory (`chmod 777`) as a test.

Bad read Remote system ran out of space, most likely in the spool area. Can also mean uucico could not read or write to device.

Conversation failed The modem is powered off, its cable pulled or loose, or the remote system crashed or shutdown. A telephone disconnection can also cause this error.

Requested or Copy (succeeded) These are informational messages only and do not indicate a problem.

PHASE 6 Status Messages

OK (Conversation Complete) The remote can deny the hangup request and reverse the roles (meaning the remote has work for the local system to do). Once the remote uucico and the local uucico agree that no more work exists, they hang up.

Conversation succeeded These are informational messages only and do not indicate a problem.

AIX UUCP Quick Setup Guide and Information Sheet

(H) Denotes HoneyDanBer versions only (or BNU).

(B) Denotes Berkeley (BSD) versions only.

( ) Denotes common across all versions.

Chat Script Parameters		
Parameters	**Version Type**	**Description**
” ”	( )	Null string.
\N	(H)	Null character.
\b	(H)	Backspace character.
\c	( )	Suppress carriage return sent at the end of every send string.
\d	(H)	Delay 2 seconds.
\d	(B)	Delay 1 second.
\e	( )	Disable echo checking.
\E	( )	Enable echo checking.
\p	(H)	Pause for approximately 1/4 second.
\K	(H)	Send a break.
\n	( )	Newline character.
\N	(H)	Send a null character. Use \0 on the other UUCP versions.
\p	(B)	
\r	( )	Carriage return character.
\s	( )	Space character.
\t	(H)	Tab character.
\ \	( )	Backslash character.
ABORT	(B)	Arm abort trap in expect string.
BREAK	( )	Generate break signal (approximately 1/4 second).
BREAK*n*	(B)	Generate *n*/10 seconds break signal.
CR	(B)	Carriage return character.
EOT	(H)	Send Ctrl–D newline Ctrl–D newline.
EOT	(B)	Send Ctrl–D newline.
NL	(B)	Newline character.
PAUSE	(B)	Delay for 3 seconds.
PAUSE*n*	(B)	Delay for *n* seconds.
P_ODD	(B)	Use odd parity for future send strings.
P_ONE	(B)	Use parity 1 for future send strings.
P_EVEN	(B)	Use even parity for future send strings.
P_ZERO	(B)	Use parity 0 for future send strings.
\ddd	( )	Send octal character.
\M	(H)	Open dial–out port with O_NDELAY (clocal on).
\m	(H)	Reopen port with clocal off.

Appendix A: 128–Port Async Adapter Configuration Worksheet

The following worksheet is designed to help you plan for your 128–port async controller configuration. You will need one copy of the worksheet for *each* line.

128-Port Async Adapter Configuration Worksheet				
Adapter:				
Line:				
Line Speed:				
Cable Type 8/4:				
Node ID	**1n**	**2n**	**3n**	**4n**
Connection Type (comm mode)				
RAN Port	**Device Type**			
1 **2** **3** **4**				
5 **6** **7** **8**				
9 **10** **11** **12**				
13 **14** **15** **16**				

Sample Worksheet

The following is a sample of a partially completed worksheet.

128-Port Async Adapter Configuration Worksheet

Adapter: 1 - cmxa0
Line: 1
Line Speed: 56Kbps
Cable Type 8/4: 8-wire

Node ID	1n	2n	3n	4n
Connection Type (comm mode)	direct	232_modem	232_modem (virtual RAN)	direct (virtual Ran)
RAN Port	**Device Type**			
1	BQ306 Terminal	BQ306 Terminal		
2	BQ306 Terminal	Modem		
3	Printer	BQ3102 Terminal		
4		Plotter		
5	BQ306 Terminal			
6	BQ306 Terminal			
7	Printer			
8	Modem			
9	BQ306 Terminal			
10	Printer			
11	BQ306 Terminal			
12	BQ306 Terminal			
13	BQ306 Terminal			
14	BQ306 Terminal			
15	BQ306 Terminal			
16	BQ306 Terminal			

Appendix B: Connector Pinouts

This appendix contains pinout information for the following ports:

- RS–423 Serial Ports
- Parallel Ports, on page B-6
- 10Base–T Ports, on page B-7
- AUI Port, on page B-8

This appendix also contains cable pinout information for the following:

- DB–25 or DB–9 Terminal, on page B-2
- MMJ Connectors, on page B-2
- Modems, on page B-3
- Extended RS–232D Modem Control Cable, on page B-4
- RS–422 Device Cables, on page B-5
- Macintosh Computers, on page B-6

RS–423 Serial Ports

This interface conforms to the electrical specifications of EIA RS–423A and has a minimum driven voltage swing of +5.6V to –5.6V in a 3K load to ground. The interface is wired as a data terminal equipment (DTE) driving TXD and receiving RXD. The interface is the same on all 16 serial ports of the concentrator. The following tables show the configuration of the interface.

Normal Mode Pinout			
Pin	**Designator**	**Circuit Name**	**CCITT Number**
1	DCD	Received Line Signal Detector	109
2	DTR	DTE Ready	108/2
3	RTS	Request to Send	105/133
4	RD	Received Data	104A
5	RD REF	DCE Common Return	102b/104B
6	CTS	Clear to Send	106
7	TD REF	Signal Ground	102
8	TD	Transmitted Data	103

Note: The input circuits are biased for fail–safe operation so that if no connection is attached, the circuits will read as follows:

Disconnected State			
Pin	**Designator**	**Circuit Name**	**Disconnect State**
1	DCD	Received Line Signal Detector	False
4	RD	Received Data	Break
6	CTS	Clear to Send	False

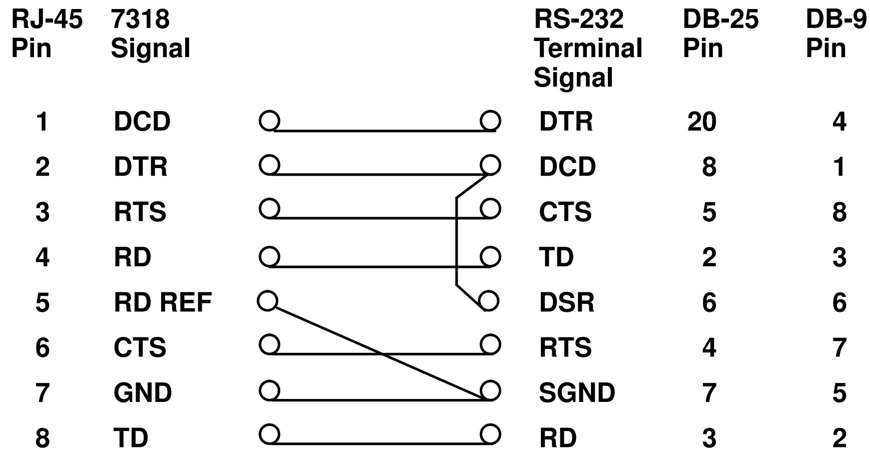
TD REF provides a common ground reference for TXD+, DTR, and RTS. It is connected to the system ground. RD REF provides a common reference for RXD+, DCD, and CTS but is isolated with respect to the system ground.

Cable and Adapter Pinouts

The following information describes the necessary wiring for cables that connect between the 7318 and various serial hardware devices.

DB–25 or DB–9 Terminal Adapter

The DB–25 or DB–9 Pinout figure depicts an expanded null modem that translates the appropriate control signals for connecting a terminal to the 7318. It also shows how the terminal signals relate to pins on the DB–25 and DB–9 connectors.

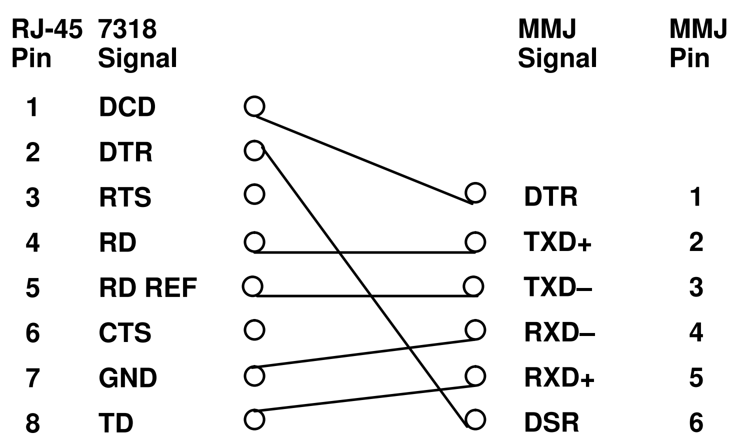


DB-25 and DB-9 Pinout

Order the DB–25 terminal adapter as Feature Code 7904 and the DB–9 terminal adapter as Feature Code 7905.

MMJ Connectors

Recent Digital Equipment Corporation (DEC) terminals (and clones) use a connector that DEC calls a modified modular jack (MMJ) connector. To attach a terminal with this type of connector, you need a special adapter. The MMJ Connector figure depicts the MMJ circuits and pins.

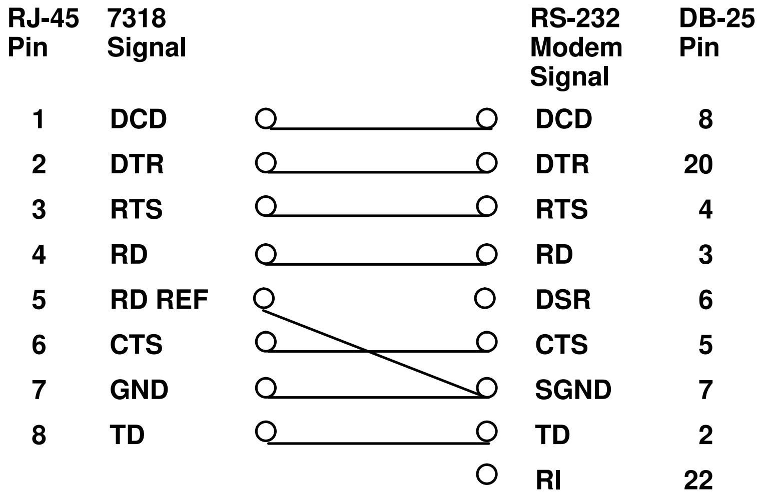


MMJ Connector

Order the RJ45 to MMJ cable as Feature Code 7907.

Modems

The Modem Pinout figure maps the control signal between an RS–232 modem and the 7318 to preserve modem control functions. The figure also depicts how the modem signals relate to pins on a DB–25 connector.

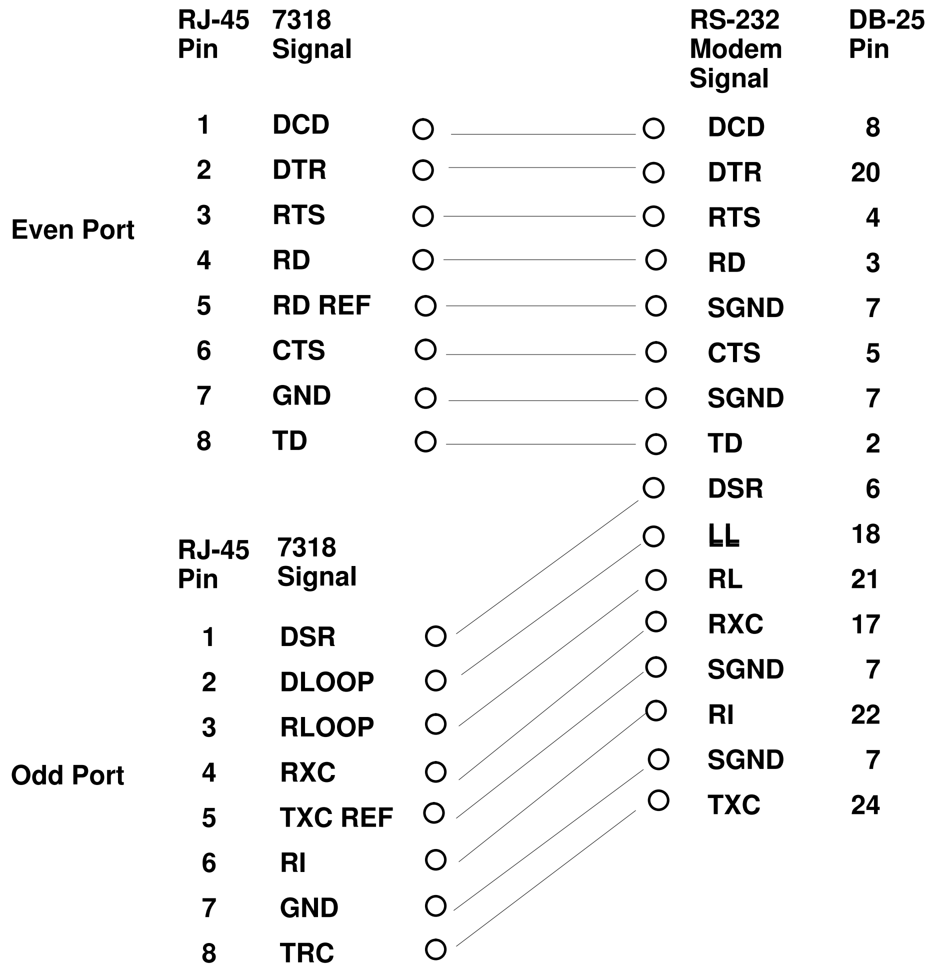


Modem Pinout

Order the DB–25 modem adapter as Feature Code 7903.

Extended RS–232D Modem Control Cable

The Extended RS–232D Modem Control Cable Pinout figure includes the DSR and RI signals. Order the Extended RS–232D Modem Control Cable as Feature Code 7902.



Extended RS–232D Modem Control Cable Pinout

General RS–422 Device Cable

The RS–422 Device Pinout figure depicts the pinout for cabling an RS–422 serial printer and the 7318. The figure also depicts how the RS–422 signals relate to the printer's DB–25 connector. When using RS–422, refer to the printer/terminal documentation for the correct pin settings. The RS–422 Device Pinout figure is a sample diagram. This is a customer–supplied part.

RJ-45 Pin	7318 Signal	RS-422 Signal	IBM 4039 Printer DB-25 Pin	IBM 3151 IBM 3161 DB-25 Pin
1	DCD			
2	DTR			
3	RTS			
4	RD	SDA	25	19
5	RD REF	SDB	19	25
6	CTS			
7	GND	RDB	15	17
8	*TD	RDA	17	15

***If more than the odd-numbered ports are used for RS-422 devices, a 100-phm current-limiting resistor must be placed in series at the device end of the cable on the TD signal.**

RS-422 Device Pinout

P10 Serial Port RS–422 Printer Cabling

The figure below is applicable to 7318 P10–style ports that are connected to RS–422 printers and are configured as native LP devices. You will notice that this configuration required additional wiring to connect RTS to CTS and DTR to DCD. These jumpers are required by AIX for serial printers and can be made at either end of the cable.

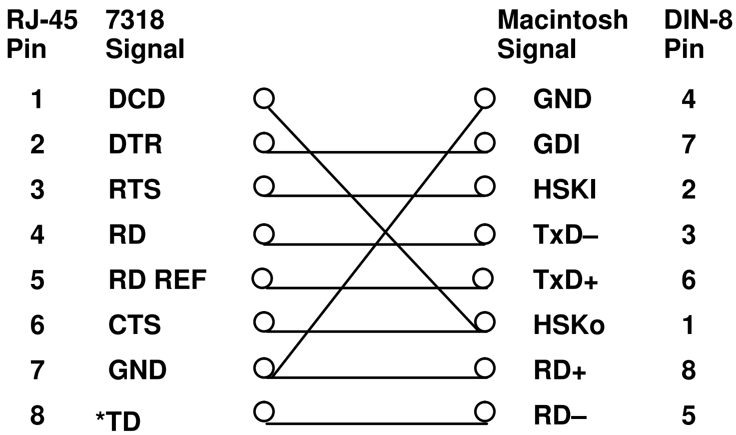
RJ-45 Pin	7318 Signal	RS-422 Signal
1	DCD	
2	DTR	
3	RTS	
4	CTS	
5	RD	SDA
6	RD REF	SDB
7	GND	RDB
8	*TD	RDA

***If more than the odd-numbered ports are used for RS-422 devices, a 100-ohm current-limiting resistor must be placed in series at the device end of the cable on the TD signal.**

P10 Port RS-422 Printer Cabling

Macintosh Computers

The Macintosh DIN–8 Port figure depicts the pinout for cabling a 7318 to Macintosh computers with DIN–8 style communications ports.



***If more than the odd-numbered ports are used for RS-422 devices, a 100-phm current-limiting resistor must be placed in series at the device end of the cable on the TD signal.**

Macintosh DIN-8 Port

Order the Macintosh DIN 8 cable as Feature Code 7906.

Parallel Ports

The 7318 has a standard DB–25F parallel interface.

Parallel Port Pinout		
Pin	**Signal**	**Direction**
1	–Strobe	Out
2-9	Data	Out
10	–Ack 1	In
11	Busy	In
12	PE	In
13	Slct	In
14	–Autofd	Out
15	–Error	In
16	–Init	Out
17	–Slct in	Out
18-25	Ground	

10Base–T Ports

The following two tables describe the 10Base–T pinouts:

T Port Pinout

The T port uses four of the eight available pins to connect to a 10Base–T hub, as shown in the following table:

T Port Pinout	
Connector Pin	**Signal Name**
1	TD+
2	TD–
3	RD+
4	
5	
6	RD–
7	
8	

T–X Port Pinout

The T–X port also uses four of the eight possible pins but crosses the RD and TD pins to directly connect to a host instead of a 10Base–T hub, as shown in the following table:

T-X Port Pinout	
Connector Pin	**Signal Name**
1	RD+
2	RD–
3	TD+
4	
5	
6	TD–
7	
8	

AUI Port

The following table describes the adapter unit interface (AUI) port pinout:

AUI Port Pinout		
Pin	**Circuit**	**Use**
3	DO-A	Data Out circuit A
10	DO-B	Data Out circuit B
11	DO-S	Data Out circuit shield
5	DI-A	Data In circuit A
12	DI-B	Data In circuit B
4	DI-S	Data In circuit shield
7	Unused	
15	Unused	
8	Unused	
2	CI-A	Control In circuit A
9	CI-B	Control In circuit B
1	CI-S	Control In circuit shield
6	Vc	Voltage Common
13	VP	Voltage Plus
14	VS	Voltage Shield
Shell	PG	Protective Ground

Appendix C: Data Transmission

Data transmission refers to the structure and timing of data, and to flow control between the 7318 and the terminal or modem. It includes the following parameters:

- Data rate
- Parity
- Stop bits
- Flow control
- Modems
- Serial Printers

Data Rate

Data rate is the speed at which data travels between devices, measured in bits per second. In order to successfully transmit data, the two devices must transmit at the same data rate; otherwise you may get distorted data or no data at all.

The physics of cables dictates that the faster you transmit data on a cable, the less reliable is its reception at the other end. Many other factors also reduce data integrity. These include:

- Cable type
- Cable length
- Electronics in the devices
- Interference from other electrical devices

Since you generally have little control over these factors, you can minimize their effect by using a slower data rate.

Terminal Data Rates

A terminal interface usually supports higher communications data rates than the speed at which the terminal processes data. Therefore, if you set the line speed to a substantially higher data rate than the terminal can process, you reduce the reliability of your connection without improving the observed response time.

To choose the optimal speed for the terminal, transmit data at the highest possible rate without introducing errors, but no faster than the terminal can actually process. You can use parity checking to verify the integrity of the data.

Slew Rate

Slew rate is the rate of change on a communications line as it transitions from 1 to 0 and from 0 to 1.

Note: Data rate specifies the duration of the ones and zeroes, not the transition time.

The 7318 allows you to program the slew rate in groups of four ports (0–3, 4–7, 8–11, 12–15) under software control. If all your devices operate at speeds of 38.4 Kbps or below, you don’t need to set the slew rate, because the default values of the 7318 are appropriate. If, however, you want to run at high speeds, such as 115.2 Kbps, you will need to increase the slew rate.

Slew rate is not directly related to data rate, since theoretically a line should be able to transition from 0 to 1 infinitely fast regardless of the actual data rate.

In practice, however, there is a relationship. Signals with a fast slew rate tend to generate interference with other signals, and this interference can limit the transmission distance of a communications signal. If the slew rate is too slow relative to the data rate, the signal does not reach a sufficiently high voltage to be recognized by the receiver on a 1 bit before it starts transitioning to a 0 bit. Consequently, there is an optimal slew rate for each data rate.

Parity

Parity is a coding scheme for checking the validity of data characters. An extra bit transmitted with each character indicates whether the sum of the other bits in the character is even or odd. You can configure most terminals to transmit data, using even, odd, or no parity bits (none).

Parity checking detects reliability problems when transferring data from the terminal to the 7318. However, because most terminals ignore parity errors on data sent to the terminal, parity checking on data sent to a terminal is useless.

The 7318 monitors and logs parity errors on data it receives. You can use the **cnsview** command to report on the number of errors observed on a port. If you observe parity errors on a port, you should take corrective action, such as:

- Lowering the data rate on the line.
- Improving the cabling on the line.
- Using a limited distance modem on the line.

Stop Bits

Stop bits delineate the end of a character in the data stream. Supported stop bits are one and two.

Flow Control

Flow control is the process of pacing data transmission so that the receiver has a chance to process all incoming characters before the transmitter sends additional data. Terminals usually require flow control to display data as fast as possible because some operations like clearing the screen take longer than simply adding a character to the screen. If you set the speed slow enough to accommodate the slowest operation, you unnecessarily delay more common operations. If you set the speed to match the fastest operation, the terminal may drop characters if it encounters a series of slow operations. Flow control enables you to set a high speed without dropping characters when the terminal is doing slow operations.

You can manage flow control using either software or hardware. Software flow control uses ASCII control characters to start and stop character transmission. Hardware flow control uses modem control signals to start and stop transmission.

Flow Control	
Advantages	**Disadvantages**
Software flow control uses only transmit and receive data circuits and does not require circuits in your cable.	Software flow control uses the same Ctrl–S and Ctrl–Q characters that some applications use.
Hardware flow control is completely independent of any application or interpretation of the data stream.	Hardware flow control requires additional wires and cables.

Modems

A modem is a data communication equipment (DCE) that uses RS–232. RS–232 defines many different circuits as part of the interface between the DCE and the data terminal equipment (DTE). Modems commonly use the following:

Modems			
Symbol	**Name**	**Direction**	**Function**
TD TxD XMT	Transmit Data	To modem	The modem receives information on this circuit.
RD RxD RCV	Receive Data	From modem	The modem sends information on this circuit.
DTR	Data Terminal Ready	To modem	The 7318 asserts DTR when it wants the modem to answer or dial out and lowers DTR when it wants the modem to hang up.
DSR	Data Set Ready	From modem	The modem asserts DSR when it is on. The 7318 assumes the modem is always on; therefore this circuit is not used (except in buddy mode).
RTS	Request to Send	To modem	The modem uses RTS for flow control. The 7318 asserts RTS when it is ready to receive more data.
CTS	Clear to Send	From modem	The modem also uses CTS for flow control. The modem asserts CTS when it is ready to send more data.
DCD	Data Carrier Detect	From modem	Indicates when a phone call connects. The modem lowers DCD when a dial–in connection hangs up or a dial–out connection is lost.
RI	Ring Indicator	From modem	Not used on Hayes–style modems and not supported by the 7318 (except in buddy mode).
GND SGND	Signal Ground		Common reference point for all data and control circuits.
RD REF RXD	Receive Reference		A 7318 circuit that attaches to a signal ground on the modem.
PGND	Protective Ground		Used for the shield of a shielded cable.

Control signals provide additional functionality, which is useful with some applications. ”Connector Pinouts”, on page B-1 explains the pinouts for the control signals.

Modems use DB–25 connectors. Using a modem adapter (DB–25 to RJ–45) to convert the DB–25 connector on your modem to an RJ–45, enables you to use twisted–pair cable between the modem and the 7318.

Serial Printers

Serial printers are normally configured as DTEs; that is, they expect to receive data on the receive data line and transmit data on the transmit data line. Like terminals, DTE–configured serial printers require a null modem cable to attach to a 7318.

Serial printers default to RS–232 connections and use DB–25 D–type connectors. Many printers also support RS–422 connections.

Appendix D: BIOS Command Summary

This appendix contains a command summary for the BIOS console. Refer to the Index for specific pages on which to find more information about the BIOS console commands.

BIOS Console Commands

Keyword	Parameter	Value	Remarks
set	Various	Various	Sets NVRAM parameter.
show	Various		Shows NVRAM parameter.
load	Path		Starts 7318 download.
biosload	Path		Reprograms BIOS NVRAM image.
save	Path		Saves NVRAM changes.
admin	Password		Sets password level.
stats			Shows LAN interface statistics.
reboot			Resets and restarts the 7318.
default			Sets BIOS NVRAM to predefined state.
diag	Various		Performs diagnostics.
ping	Address		Sends diagnostic packet echo.

The following **set** commands affect NVRAM.

Set Commands

Key	Item	Value	Default	Bounds	Remarks
set	loadimage	String	null	64 chars	Must be AIX path name.
	config	String	*xxxxxxxxxxxx*.**cfg**	64 chars	Must be AIX file name.
	inet	Address	0.0.0.0		*xxx.xxx.xxx.xxx* format.
	password	String	null		Set password.
	interface	Decimal	auto	0–4	0 = Auto LAN select. 1 = 10Base–T. 2 = AUI. 3 = Reserved. 4 = Disable loading.
	protocol	Decimal	auto	0–2	0 = Auto protocol. 1 = IPX protocol. 2 = UDP protocol.
	netmask	Address	0.0.0.0		*xxx.xxx.xxx.xxx* format.
	gateway	Address	0.0.0.0		*xxx.xxx.xxx.xxx* format.
	name–server	Address	0.0.0.0		*xxx.xxx.xxx.xxx* format.
	domain	String			
	host1	Address	auto		*nnnnnnnn*:*dddddddddddd* or *xxx.xxx.xxx.xxx* format.
	host2	Address	auto		*nnnnnnnn*:*dddddddddddd* or *xxx.xxx.xxx.xxx* format.

	name	String	CNS*Ethernet#*	16 chars	Set server name.
	log	BOOLEAN	disable	0–1	Disable/enable verbose log.
	console	0–15	disable		0–15 = Ports.
	upload	BOOLEAN	enable	0–1	Upload control.
show	self				Shows self–test result.
	version				Shows BIOS version.
	pmem				Shows physical memory.

The following **set** commands affect the **diag** command.

Set Commands

Key	**Item**	**Value**	**Default**	**Bounds**	**Remarks**
set	iterations	Decimal	1	10000	
	device	String	cns		cns, com*xx*, lpt*x*, lan
	pair	String	None		cns, com*xx*, lpt*x*, lan
	size	Decimal	512	2–910	
	pattern	String	LN		
	rate	String	9600		

Index

Symbols

Numbers

A

L

M

N

O

P

V

X

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull AIX Versions 3.2 and 4 Asynchronous Communications Guide

Nº Reférence / Reference Nº : 86 A2 26AQ 05

Daté / Dated : December 1998

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : ______________________________ Date : __________

SOCIETE / COMPANY : ______________________________

ADRESSE / ADDRESS : ______________________________

____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Technical Publications Ordering Form

Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:
Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

BULL ELECTRONICS ANGERS
CEDOC
ATTN / MME DUMOULIN
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

Managers / Gestionnaires :
Mrs. / Mme : **C. DUMOULIN** +33 (0) 2 41 73 76 65
Mr. / M : **L. CHERUBIN** +33 (0) 2 41 73 63 96

FAX : +33 (0) 2 41 73 60 19
E–Mail / Courrier Electronique : srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:
`http://www-frec.bull.com` (PUBLICATIONS, Technical Literature, Ordering Form)

CEDOC Reference # Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté	**CEDOC Reference #** Nº Référence CEDOC	**Qty** Qté
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
__ __ ____ _ [__]		__ __ ____ _ [__]		__ __ ____ _ [__]	
[_ _] : **no revision number means latest revision** / pas de numéro de révision signifie révision la plus récente					

NOM / NAME : ______________________________ Date : __________

SOCIETE / COMPANY : ______________________________

ADRESSE / ADDRESS : ______________________________

____

PHONE / TELEPHONE : ____________________ FAX : ____________________

E–MAIL : ______________________________

For Bull Subsidiaries / Pour les Filiales Bull :

Identification: ______________________________

For Bull Affiliated Customers / Pour les Clients Affiliés Bull :

Customer Code / Code Client : ______________________________

For Bull Internal Customers / Pour les Clients Internes Bull :

Budgetary Section / Section Budgétaire : ______________________________

For Others / Pour les Autres :

Please ask your Bull representative. / Merci de demander à votre contact Bull.

PLACE BAR CODE IN LOWER LEFT CORNER

BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 26AQ 05

Utiliser les marques de découpe pour obtenir les étiquettes.

Use the cut marks to get the labels.

AIX

AIX Versions 3.2
and 4
Asynchronous
Communications
Guide

86 A2 26AQ 05

AIX

AIX Versions 3.2
and 4
Asynchronous
Communications
Guide

86 A2 26AQ 05

AIX

AIX Versions 3.2
and 4
Asynchronous
Communications
Guide

86 A2 26AQ 05