Remote Annex 6300

Supplement to the *Remote Annex Administrator's Guide for UNIX*

Part No. 166-024-028   Rev. A
January 1997

Bay Networks

The information in this document is subject to change without notice. The statements, configurations, technical data, and recommendations in this document are believed to be accurate and reliable, but are presented without express or implied warranty. Users must take full responsibility for their applications of any products specified in this document. The information in this document is proprietary to Bay Networks, Inc.

The software described in this document is furnished under a license agreement and may only be used in accordance with the terms of that license.

Restricted Rights Legend

Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Notice for All Other Executive Agencies

Notwithstanding any other license agreement that may pertain to, or accompany the delivery of, this computer software, the rights of the United States Government regarding its use, reproduction, and disclosure are as set forth in the Commercial Computer Software-Restricted Rights clause at FAR 52.227-19.

Trademarks of Bay Networks, Inc.

Annex, Remote Annex, Annex Manager, Remote Annex 2000, Remote Annex 4000, Remote Annex 6100, Remote Annex 6300, Remote Annex 5390/Async, Remote Annex 5391/CT1, Remote Annex 5393/PRI, BayStack Remote Annex 2000 Server, Quick2Config, Bay Networks, Bay Networks Press, and the Bay Networks logo are trademarks of Bay Networks, Inc.

Third Party Trademarks

All other trademarks and registered trademarks are the property of their respective owners.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Bay Networks, Inc. reserves the right to make changes to the products described in this document without notice.

Bay Networks, Inc. does not assume any liability that may occur due to the use or application of the product(s) or circuit layout(s) described herein.

In addition, the program and information contained herein are licensed only pursuant to a license agreement that contains restrictions on use and disclosure (that may incorporate by reference certain limitations and notices imposed by third parties).

Revision Level History

Revision	Description
A	Initial release.

Contents

Preface

Book A: Configuration Procedures

Chapter 1
Introduction to the Remote Annex 6300

Chapter 2
Configuring the Remote Annex 6300

Chapter 3
Configuring the PRI Interface, Global Ports, and Sessions

Chapter 4
The Port Server and Rotaries

Chapter 5
Modems

Chapter 6
Point-to-Point Protocol (PPP)

Chapter 7
Serial Line Internet Protocol (SLIP)

Chapter 8
Dial-up Networking

Chapter 9
Routing Information Protocol (RIP)

Chapter 10
Filtering

Chapter 11
Internetwork Packet Exchange (IPX) Protocol

Chapter 12
AppleTalk

Chapter 13
Printers

Chapter 14
Configuring Hosts and Servers

Chapter 15
Using RA 6300 Security

Book B: Network Management

Chapter 1
Network Administration

Chapter 2
Simple Network Management Protocol (SNMP)

Book C: Reference

Chapter 1
na Commands

Chapter 2
Configuration Parameters

Tables

Preface

This manual is intended for the person responsible for installation, configuration, and day-to-day administration of the Remote Annex 6300 (RA 6300) and/or the Remote Annex 5393/PRI. (For brevity, the manual usually refers only to the RA 6300.)

The RA 6300 operates in heterogeneous network environments providing access for remote nodes to corporate LANs, the Internet via Internet Service Providers (ISPs), and other networking resources through an ISDN Primary Rate Interface (PRI) connection to the global telephone network. The RA 6300 accepts connections from analog modems, ISDN terminal adapters (TAs) using the V.120 Rate adaptation protocol, ISDN NICs, and ISDN routers such as the Nautica Series Marlin and CLAM.

This manual assumes its readers have a basic familiarity with UNIX systems and network administration in general and with the basic concepts of Integrated Services Digital Network (ISDN). The host operating systems referenced are versions of UNIX distributed by the University of California at Berkeley, 4.2BSD and 4.3BSD, or versions of System V.

Supported Platforms

This document supports the Remote Annex 6300 and the Remote Annex 5393/PRI.

Using this Manual

This manual is a supplement to the *Remote Annex Administrator's Guide for UNIX*. It is organized into three books:

- *Book A: Configuration Procedures* presents a general introduction to the RA 6300 as well as configuration information.
- *Book B: Network Management* describes the utilities for managing RA 6300s and monitoring the network.
- *Book C: Reference* provides a reference material for the commands, parameters, utilities, and network protocols supported by the RA 6300.

This manual is called a "supplement" because several of its chapters only describe how the RA 6300 differs from other members of the Remote Annex product family.

Supported Version

This manual supports Release 4.2 of the Remote Annex 6300 software. The Engineering Release is 13.3.

Conventions

This manual uses the following conventions:

`special type` In examples, special type indicates system output.

`special type` Bold special type indicates user input.

<cr> In command examples, this notation indicates that pressing the **Return** key enters the default value.

lowercase bold Lowercase bold indicates commands, pathnames, or filenames that must be entered as displayed.

lowercase italics In the context of commands and command syntax, lowercase italics indicate variables for which the user supplies a value.

[ ] In command dialogue, square brackets indicate default values. Pressing the **Return** key selects this value. Square brackets appearing in command syntax indicate optional arguments.

{ } In command syntax, braces indicate that one, and only one, of the enclosed values *must* be entered.

| In command syntax, this character separates the different options available for a parameter.

CTRL-*X* This notation indicates a two-character sequence for control characters. To enter the control character, hold down the **Control** key (often labeled CTRL) and press the character specified by *X*.

Related Documents

Each RA 6300 or RA 5393/PRI hardware platform ships with the appropriate hardware guide. The remaining documentation is included with the software.

Book A

Chapter 1
Introduction to the Remote Annex 6300

This chapter provides overviews of the following:

- ISDN.
- The Remote Annex 6300 (RA 6300).
- The RA 6300's PRI interface.
- Typical network configurations for the three types of traffic the RA 6300 recognizes.
- RA 6300 management tools.
- The different types of parameters you use to configure the RA 6300.
- A list of general Annex capabilities supported by the RA 6300, including multiprotocol support and security systems.

The software described in this document also applies to the Remote Annex 5393/PRI.

ISDN Overview

ISDN provides a high-speed, digital link to the telecommunications network for all types of remote users, including telecommuters, mobile workers, and business personnel. ISDN replaces existing analog links and provides integrated communications services for voice, data, and video. It is an affordable alternative to leasing expensive dedicated digital lines or using lower speed analog modems.

ISDN Characteristics

Plain Old Telephone Service (POTS) allows analog modems and telephones to carry a single data transmission or voice conversation over copper lines between the user and the local telephone company. This link to the local telephone company is referred to as the local loop.

ISDN uses the same copper lines but operates digital communication paths over them. These paths are called *channels*. ISDN uses two channel types:

- B Channels (or Bearer Channels), which are circuit switched channels.
- D Channels (or Data Channels), which are packet switched channels.

ISDN Services

Local telephone companies offer ISDN services under different names and combined with custom features. However, two basic types of ISDN are available:

- Basic Rate Interface (BRI), a service used by individual users and small businesses.
- Primary Rate Interface (PRI), a service primarily used by medium-to-large businesses and other institutions.

Basic Rate Interface

The ISDN BRI service divides a single telephone line into three logical digital channels: two B channels and one D channel. BRI is also referred to a 2B+D service. A single BRI can handle multiple devices and simultaneous data, voice, and video traffic.

The BRI B channels operate at transmission rates of 64Kbps or 56Kbps. These channels usually carry data, voice, and video traffic, although some telephone companies also use B channels to carry signals normally carried by the D channel. Some companies handle signaling accounts for the lower B channel transmission rate (56Kbps vs. 64Kbps).

A BRI D channel operates at 16Kbps. It sets up and tears down calls, and handles other connection management signaling between the user's equipment and the ISDN-capable switch at the local telephone company. The D channel also manages transmission and reception of packet data over an X.25 network.

Primary Rate Interface

The ISDN PRI service provides greater B channel capacity and higher-speed D channel operation than BRI service. PRI is provided over dedicated trunk lines such as T1 and E1. Businesses and other institutions use PRI service to connect their communications resources to ISDN:

- Private Branch Exchange (PBX) switches.
- Local Area Networks (Loans).
- Wide Area Networks (WANs).
- Video conferences equipment.

In the United States, Canada, and Japan, PRI service is provided over T1 trunk lines. A single T1 line supports 23 B channels (for voice, data and video) and 1 D channel (for call set up, tear down, and connection management). All B and D channels operate at 64Kbps and provide 1.544Mbps of total bandwidth. This flavor of PRI service is referred to as 23B+D.

In Europe and the Pacific Rim, PRI service is provided over E1 trunk lines. A single E1 line supports up to 30 B channels (for voice, data, and video) and 2 D channels. One D channel handles framing and connection maintenance. The second D channel handles call setup and signaling. All B and D channels operate at 64Kbps and provide 2.048Mbps of total bandwidth. This flavor of PRI service is referred to as 30B+2D.

Incorporating ISDN Services

Users can tailor ISDN services to meet their requirements and budget constraints. Among the options are:

- Using analog devices and replacing some analog links with ISDN.
- Replacing all analog links with ISDN and integrating existing analog devices with ISDN.
- Replacing all analog links with ISDN and replacing analog devices with equivalent ISDN devices.

Using Analog Devices with Analog and ISDN Links

By incorporating ISDN with existing analog links and equipment, users can realize improved communication performance. This is a low cost option.

For example, a business can subscribe to ISDN PRI service to connect its local area network (LAN) to the local telephone company. Telecommuters and mobile workers (e.g., sales staff) can continue to use analog modem links to communicate with their company. Also, a home computer user can use an analog modem link to call a local telephone number for Internet access. The local telephone company processes the call, then directs it through ISDN PRI lines to the Internet access service provider.

Integrating Existing Analog Devices with ISDN

Users who replace their analog telephone line with ISDN BRI service do not need to give up their analog equipment (telephone, modem, fax machine, etc.). Using analog equipment on an IDSN line offers these users higher speed transmission rates while preserving their investments in analog equipment. The devices access BRI service through the use of a device called a terminal adapter (TA).

A TA performs two tasks:

- Call set up, tear down, and management signaling conversion. This allows analog devices to set up calls and connections through the ISDN BRI.
- Converts the transmission of voice and data from analog to digital. For synchronous and asynchronous data passed to it, the TA adapts the data for transmission over the ISDN B channel. Most TAs perform this task according to the ITU V.120 standard. The process is known as V.120 rate adaptation.

Replacing Analog Devices and Links with ISDN Devices and Links

BRI or PRI service users who replace their analog devices with ISDN capable ones realize full end-to-end digital communications. These connections offer the highest speed and most error free operation.

For BRI service, ISDN telephones and other devices connect directly to the ISDN BRI without an intermediate device such as a TA.

For PRI service, ISDN telephones and FAX machines might connect to PRI lines through an ISDN-capable Private Branch Exchange (PBX) switch. Computer equipment (PCs, file servers, etc.) on a LAN might connect to PRI lines through an ISDN router.

For example, a telecommuter can connect a PC to an external ISDN router through a Network Interface Card (NIC) installed in the PC. The router allows the PC to operate as a separate LAN. Connections to other networks (e.g., at a company headquarters) operate as any other routed internetwork according to the routing protocols in use (e.g., IP or IPX).

Obtaining ISDN Service

Setting up ISDN service generally involves:

- Selection of service and features available from the local telephone company. Feature selection is governed by the capabilities of the user’s ISDN equipment and those of the telephone company’s ISDN switch. This activity is referred to as provisioning and is described in the document entitled *Ordering Primary Rate ISDN for the Remote Annex 6300*.

- Identification of the switch type used by the telephone company. This information is required to set up the user's ISDN equipment properly.
- Obtaining a Service Profile Identifier (SPID) from the telephone company for each device connected to the ISDN BRI or PRI. The SPID lets the telephone company switch know which ISDN services a particular device can access. The user needs to know each device's SPID to set up that device to work on the ISDN line.

ISDN Standards Compliance

ISDN is based on a set of digital transmission protocols defined by the International Telecommunications Union (ITU).

Although ISDN standards are accepted worldwide, universal adoption is not yet a reality. There are two causes for the lack of universal standards-based service.

First, many United States telephone companies must modernize their switches to perform call signaling through Signaling System 7 (SS7). SS7 (an international standard) specifies that ISDN D channel carry all call set up, tear down, and connection management signaling exclusively. In the United States, some telephone companies with older switches must use a portion of BRI B channels to carry part of this signaling. As a result, these companies offer BRI B channel transmission rates at 56Kbps, not at the full ISDN bandwidth of 64Kbps.

Second, many digital telephone switches were installed while ISDN standards were evolving. At that time, switch vendors (e.g., AT&T and Nortel) offered switches with custom ISDN capabilities. As a result, there are differences in the ISDN service features offered by local telephone companies.

ISDN service differences are almost nonexistent outside the United States. In most other countries, regulatory bodies dictate the use of a single vendor’s switch across the entire telephone network. The United States has multiple telephone companies that make their own decisions about the switches they use.

Even though service and feature differences exist, universal access is still possible since all international carriers and local phone companies accommodate the service differences. ISDN users can still reach resources across the global telecommunications network. Further, all telephone companies and long distance carriers have committed to adopting ISDN standards.

RA 6300 Overview

The RA 6300 described in this document is a PRI ISDN server. It accepts the following kind of traffic:

- Digital data conforming to the V.120 rate adaptation protocol. The RA 6300 implements a subset of this protocol that is supported by most Terminal Adapters (TAs).
- Digital data conforming to the synchronous PPP specification.
- Analog Pulse-Code Modulated (PCM) voice data originating from a standard analog (for example, V.34) modem.

The RA 6300 supports all of the dial-in, multiprotocol, remote access features characteristic of other Remote Annex products. Moreover, ISDN calls originating from a router (such as the Clam and Marlin) have access to standard Annex IP routing capabilities.

Currently, the RA 6300 does not support dialout. This feature will be available in a future release.

The above features allow network administrators to consolidate their network needs into a single product that:

- Supports analog calls, which are currently the predominant access method for remote users.
- Supports ISDN connections for the same users in the future.

The Primary Rate Interface

The RA 6300 provides a single ISDN Primary Rate Interface (PRI). The PRI line connects to a PRI line module within the RA 6300. Two versions of this module are available, one for the United States and another for Europe:

- The U.S. PRI module uses T1 as the physical medium and supports 23 ISDN B channels and one D channel. The B channels operate at 64 Kbps or 56kbps and carry user, voice (modem), or digital data. The D channel is used for signaling.
- The European module uses E1 as the physical medium and supports 30 ISDN B channels and two D channels, one for signaling and another for framing.

The PRI and the Ethernet port (en0) are the only physical ports on RA 6300. (Three types of Ethernet connectors are provided, but you use one of them at a time.)

Typical ISDN Network Configurations

The following are examples of the three ISDN connection types supported by the RA 6300.

Voice (Modem) Calls

The example in Figure A-1 shows a PC user with a standard modem (e.g., a V.34) dialing into the RA 6300.

Other modem call configurations can be established instead of the one shown in Figure A-1.

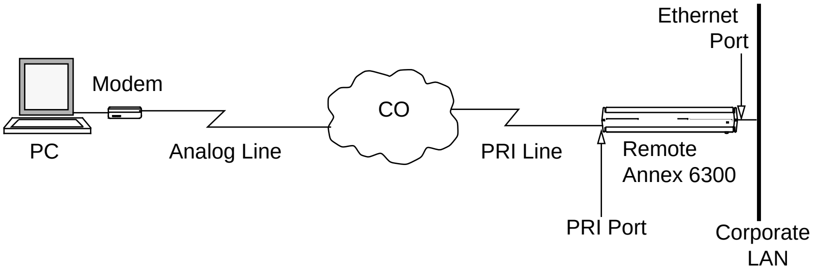


Figure A-1. Voice Call Over Analog Line

In Figure A-1, the CO of the telephone company terminates the analog line from the modem, converts the voice call into digital data, and dynamically chooses a PRI B channel to carry the data through the telephone network to the RA 6300. The CO also converts the signals describing the characteristics of the call into ISDN out-of-band signaling messages, as defined by the Q.931 standard. These signaling messages are carried to the RA 6300 on the D channel (channel 24).

The modem call is not automatically accepted by the RA 6300. First, the RA 6300 and the CO negotiate to determine whether the specified B channel is free on the RA 6300 and whether or not the RA 6300 has the resources, such as an available modem, to handle the call. RA 6300 negotiations are based not only on physical resources but also on parameters that you, as network administrator, configure. If the RA 6300 and the CO reach agreement and the parameters are set properly, the call is accepted.

Once a call is accepted (and the user is authenticated), the RA 6300 places the user in the protocol environment you have configured. Protocols supported are asynchronous PPP, SLIP, ARAP, or CLI.

Spare Modems

The domestic PRI/T1 RA 6300 contains an optional internal modem bank of up to 24 modems. Given only 23 B channels, this leaves one extra modem. Similarly, the European PRI/E1 RA 6300 can support up to 32 modems. Given only thirty B channels, this leaves two extra modems.

The extra modems can be used as spares. The RA 6300 treats the modems as a modem pool, assigning the next available modem to the next call, regardless of the B channel the call arrives on. This round-robin assignment method ensures that all the modems are used. If a PRI/T1 modem fails, the RA 6300 can continue to support the full set of 23 channels by using the spare. Even if 2 PRI/E1 modems fail, the RA 6300 can continue to support the full set of 30 channels.

V.120 Calls

The example in Figure A-2 shows a TA using the V.120 Rate Adaptation protocol to convert asynchronous data generated by a PC into V.120 frames. These frames are transmitted over the BRI line to the appropriate switch at the CO and then sent to the RA 6300 on a negotiated PRI B channel. If negotiations between the CO and the RA 6300 are successful, and if you have properly configured RA 6300 parameters for this kind of call, then the call is accepted. The RA 6300 converts the V.120 frames into an asynchronous data stream, and the RA 6300 software handles the data as if it originated at V.120 asynchronous port.

Other V. 120 call configurations can be established instead of the one shown in Figure A-1. For example, the BRI line could be a PRI line.

Once a call is accepted (and the user is authenticated), the RA 6300 places the user in the protocol environment you have configured. Protocols supported for V.120 calls are asynchronous PPP, SLIP, ARAP, or CLI.

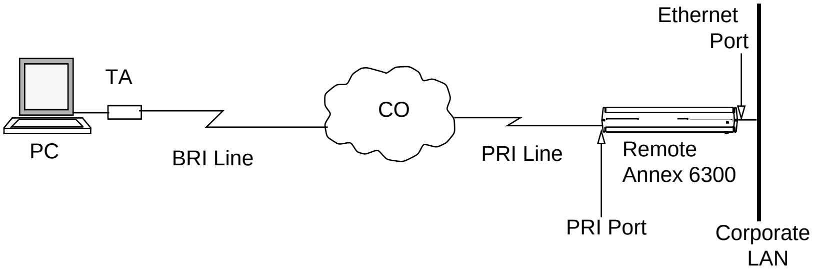


Figure A-2. V.120 Call over a BRI Line

Synchronous PPP Calls

Figure A-3 shows a sample synchronous PPP call from a user whose PC has an adapter card supporting synchronous PPP encapsulation over ISDN. The user connects to the CO via a BRI line and connects to the RA 6300 over a negotiated PRI B channel. If negotiations between the CO and the PRI are successful, and if you have properly configured RA 6300 parameters for this kind of call, the call is accepted.

Once a call is accepted (and the user is authenticated), the RA 6300 places the user in the protocol environment you have configured. Protocols supported for synchronous calls are IPCP (IP over PPP), IPXCP (IPX over PPP), and ATCP (AppleTalk over PPP).

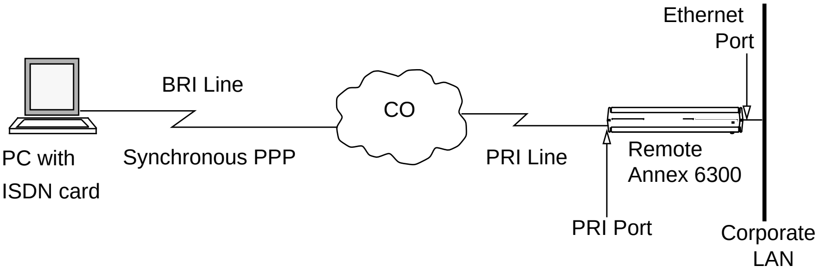


Figure A-3. Synchronous PPP Connection

Multilink PPP

The RA 6300 also supports synchronous Multilink PPP (MP). MP is a protocol standard that provides a method to adjust the bandwidth of a connection between two network devices to accommodate dynamically changing network loads. This implementation of MP is based on the RFC1990 technical specification. For information on using and configuring MP, see the *Multilink PPP Addendum to the Remote Annex Administrator's Guide*.

Mixed Annex Environment

Figure A-4 shows how an RA 6300 can fit into an environment with other types of Annexes to provide a complete enterprise solution. In the figure, other types of Annexes are unlabeled.

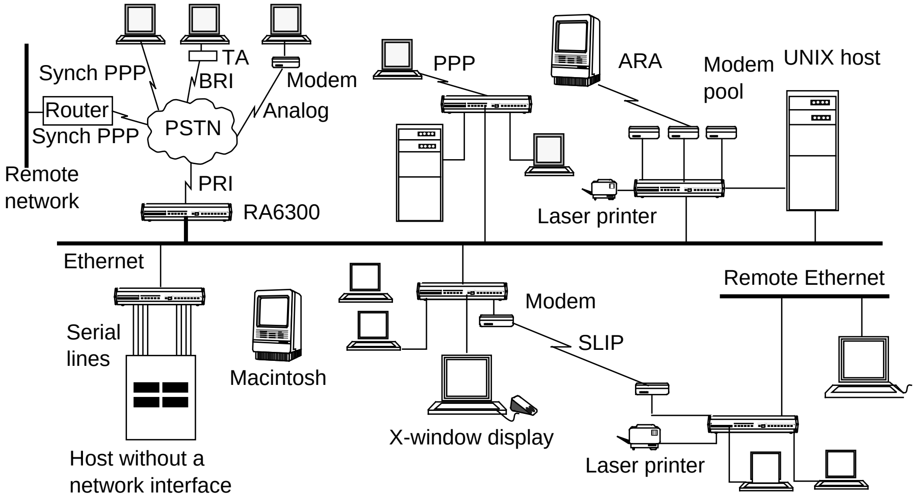


Figure A-4. Network with Mixed Annex Types

Configuring Parameters

As mentioned in the previous sections, the RA 6300 does not accept a call unless you have set certain parameters properly. You can set parameters using various RA 6300 management tools.

RA 6300 Management Tools

The RA 6300 software provides network management tools and files for setting the parameters that control the RA 6300 environment.

- The Network Administrator (**na**) program is a host-based UNIX utility. It provides commands for displaying and modifying operating characteristics of the RA 6300, its en0 or PRI port, and PRI call handling.
- The Command Line Interpreter (CLI) is the RA 6300’s command interface. Among other things, CLI commands allow users to connect to hosts and to display and change RA 6300 characteristics. Users can also display known hosts, as well as statistics for the RA 6300 and the network. The CLI also provides superuser commands for network administration and management.
- The CLI **admin** command, which you access as a superuser on a CLI connection, is a local (resides in the RA 6300) substitute for the host-resident **na** command. The **admin** command set provides a subset of the host-resident **na** commands. However, all parameters that you can set via **na** you can also set using **admin**.
- Certain host-based configuration files allow you to create entries that can control, among other things, PRI call handling and user security. The default configuration file is named **config.annex** and is located in the **/usr/annex/** directory on the UNIX load host.

Parameter Types

You configure the RA 6300 by setting the following types of parameters:

- Annex-wide parameters that apply to an entire RA 6300 or set of RA 6300s. This parameter type is described in *Configuring the Remote Annex 6300* on page A-17.

- Global port parameters that apply to all ISDN calls. You can also set these parameters for a subset of calls. To define port parameters globally, you use **na** or **admin**. To define the same parameters for a subset of calls, you enter the parameters in a Session Parameter Block (SPB) that handles the calls. SPBs are located in the Annex configuration file. For detailed information, see *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51.
    - PRI parameters, which define characteristics of the PRI line and establish remote IP and IPX addresses.
    - Interface parameters.
    - Modem parameters that apply to the internal modem set.

Loading Files

RA 6300 files can be loaded from a host using either the trivial file transfer protocol (**tftp**) or the expedited remote procedure call daemon (**erpcd**).

- The **erpcd** utility runs on a UNIX host; it listens for RA 6300 file server host requests (to download the operational code and other files).
- The **tftp** program, supplied on most hosts, is supported as an alternative to **erpcd** and as a back-up in case a UNIX host is not available to install **erpcd**.

If a software problem occurs, the RA 6300 can send a memory dump to a host. These dumps assist customer support personnel in resolving problems.

Multi-protocol Support

Once a call has been established, the RA 6300 allows use of multiple protocols, including asynchronous IP, PPP, SLIP, IPX, LAT (with limitations) and AppleTalk, as well as synchronous IP over PPP, IPX over PPP (IPXCP), and AppleTalk over PPP.

Chapter 2 Configuring the Remote Annex 6300

This chapter describes configuring Annex-wide parameters – that is., parameters that apply to an entire RA 6300 or to multiple RA 6300s. This involves defining the unit's necessary operating and administrative attributes, which include:

- The Internet addresses for the RA 6300.
- The preferred hosts for booting and dumping.
- Security for the RA 6300.
- The name servers to be used.
- Event logging.
- The local time zone for using a time server.
- A customized RA 6300 environment.
- LAT services.
- AppleTalk access.

Please note the two major differences (as described in this chapter) between configuring any other type of Remote Annexes and configuring the Remote Annex 6300:

- You cannot use SLIP to boot from or dump to a Remote Annex 6300.
- You cannot log events to a port on the Remote Annex 6300.

Configuring RA 6300 Parameters

You can configure RA 6300 parameters using the following:

- The host-based **na** utility.
- The CLI superuser **admin** command.
- Annex Manager (GUI).
- A SNMP based manager such as Sun NetManager.

To determine the current settings of RA 6300 parameters, use the **show annex all** command. The **set annex** command allows you to change any setting. All parameters have default settings. Some of these parameters must be set using the ROM Monitor before booting the RA 6300 with its operational code (see the *Remote Annex 6300 Series Hardware Installation Guide* for more details).

By default, the **show annex** command scrolls the selected parameters line by line in two-column format.

You can set up a pager as follows:

`setenv PAGER more` (BSD)

or

`set PAGER=more; export pager` (System V)

The **./src/na/README** file describes how to use a *pager* along with the **show** command.

Using the na Utility

1. **Log into a UNIX host and enter** na**:**

```
% na
Annex network administrator Rx.x January 1997
command:
```

2. **Specify one RA 6300, or specify multiple RA 6300s:**

```
command: annex 192.9.200.95
          or annex 192.9.200.95,frontlobby
          or annex
enter default annex list: 192.9.200.95,frontlobby
```

3. **Execute the** set annex **command to change parameters. The following sample command lines:**

- enable the DNS name server.
- define two name server hosts.
- enable security on the RA 6300.
- define a security server host.
- enable security for virtual CLI connections.
- define an administrative password.
- enable event logging.
- define a CLI prompt.

```
command:   set annex name_server_1 dns
command:   set annex pref_name1_addr 192.9.200.95
command:   set annex name_server_2 dns
command:   set annex pref_name2_addr 192.9.200.85
command:   set annex enable_security Y
command:   set annex vcli_security Y
command:   set annex pref_secure1_host 192.9.200.95
command:   set annex password piano
command:   set annex syslog_mask all
command:   set annex syslog_host 192.9.200.95
command:   set annex cli_prompt "%a%c"
```

4. **Execute the** show annex all **command to review your changes. Using the example in step 3, the terminal displays:**

```
command: show annex all

Annex Generic Parameters

inet_addr: 132.245.44.187          subnet_mask:255.255.255.0
pref_load_addr: 132.245.44.80      pref_dump_addr:132.245.33.8
load_broadcast: Y                  broadcast_addr:132.245.44.255
load_dump_gateway:132.245.44.22    load_dump_sequence: net
image_name: ""                     motd_file: "motd"
config_file: "config.annex"        authoritative_agent: Y
routed: Y                          server_capability: none
disabled_modules: vci              tftp_load_dir: ""
tftp_dump_name: ""                 ipencap_type: ethernet
ip_forward_broadcast: N            tcp_keepalive: 120
option_key: ""                     session_limit: 1152
output_ttl: 64


VCLI Parameters

max_vcli: unlimited                cli_prompt: "%a%c"
vcli_security: Y                   vcli_password: "<unset>"
vcli_inactivity off


Nameserver Parameters

nameserver_broadcast: N            rwhod: Y
pref_name1_addr: 192.9.200.95      name_server_1: dns
pref_name2_addr: 192.9.200.85      name_server_2: dns
host_table_size: 64                min_unique_hostnames: Y


Security Parameters

enable_security: Y                 security_broadcast: Y
pref_secure1_host:192.9.200.95     pref_secure2_host: 0.0.0.0
network_turnaround: 2              loose_source_route: Y
acp_key: "<unset>"                 password: "<set>"
allow_snmp_sets: N                 lock_enable: Y
passwd_limit: 3                    chap_auth_name: "chap"
max_chap_chall_int: 0
```

(continued on next page)

```
Time Parameters

time_broadcast: N                  daylight_savings: us
timezone_minuteswest: 300          time_server: 0.0.0.0


SysLog Parameters

syslog_mask: all                   syslog_facility: log_local7
syslog_host: 192.9.200.95          syslog_port: 0


MOP and "Login" User Parameters

pref_mop_host: 00-00-00-00-00-00
mop_password: “<unset>”            login_password: “<set>”
login_prompt: “#”                  login_timer: 30


LAT Parameters

lat_key: ""                        facility_num: 0
server_name: ""                    sys_location: ""
lat_queue_max: 4                   service_limit: 256
keep_alive_timer: 20               circuit_timer: 8
retrans_limit: 8                   group_value: none
vcli_groups: none                  multicast_timer: 30
multisessions_enable: N


AppleTalk Parameters

a_router: 00-00-00-00-00-00
default_zone_list: ""              node_id: 0.0
zone: ""


Router Parameters

rip_auth: "<unset>"                rip_routers: all


IPX Parameters

ipx_file_server: ""                ipx_frame_type: raw802_3
ipx_dump_username: ""              ipx_dump_password: "<unset>"
ipx_dump_path: ""                  ipx_do_checksum: N
```

(continued on next page)

```
TMux Parameters

tmux_enable: N                        tmux_max_host: 64
tmux_delay: 20                        tmux_max_mpx: 700


DHCP Parameters
pref_dhcp1_host: 0.0.0.0              pref_dhcp2_host: 0.0.0.0
dhcp_bcast: N
```

5. **Execute either** boot **or** reset annex all **to effect these changes at the Annex.**

You can configure more than one RA 6300 simultaneously using one of these sequences:

- Define the RA 6300 using the **annex** command. Next, use the **set annex** command to change the parameters.
- Define the parameters for one RA 6300 and use the **write** command to copy the parameters to a script file. The script file will contain all copied parameter settings with a comment character (#) at the beginning of lines defining the settings for the Internet address, administrative password, virtual CLI password, LAT key, option key, and ACP key. Remove the comment character for any of these settings you want to use.
- Next, execute the **read** command for all RA 6300s you want to configure.

Using the CLI admin Command

Entering the Annex superuser CLI **admin** command puts you in administrative mode. The *admin* prompt replaces the CLI prompt. Pressing the attention key or typing quit at the *admin* prompt terminates the **admin** session and returns you to the superuser CLI prompt.

The **admin** command functions only on the local RA 6300.

1. **At the CLI prompt, execute the** su **command and enter the superuser password:**

```
annex: su
password:
```

2. **At the superuser CLI prompt, execute the** admin **command:**

```
annex# admin

ANNEX-PRI Rx.x, 32 async, 32 sync, 32 ta, 32 modem ports

admin:
```

3. **Execute the** set annex **command to change parameters. The following sample command lines:**

   - enable the DNS name server.
   - define two name server hosts.
   - enable security on the RA 6300.

```
admin: set annex name_server_1 dns
admin: set annex pref_name1_addr 192.9.200.95
admin: set annex name_server_2 dns
admin: set annex pref_name2_addr 192.9.200.85
admin: set annex enable_security Y
```

4. **Execute the** show annex all **command to review your changes. Using the example in step 3, the terminal displays:**

```
admin: show annex all

Annex Generic Parameters

inet_addr:132.245.44.187           subnet_mask:255.255.255.0
pref_load_addr:132.245.44.80       pref_dump_addr:132.245.33.8
load_broadcast:Y                   broadcast_addr:132.245.44.255
load_dump_gateway:132.245.44.22    load_dump_sequence: net
image_name: ""                     motd_file: "motd"
config_file: "config.annex"        authoritative_agent: Y
routed: Y                          server_capability: none
disabled_modules: vci              tftp_load_dir: ""
tftp_dump_name: ""                 ipencap_type: ethernet
ip_forward_broadcast: N            tcp_keepalive: 120
option_key: ""                     session_limit: 1152
output_ttl: 64


VCLI Parameters

max_vcli: unlimited                cli_prompt: "%a%c"
vcli_security: Y                   vcli_password: "<unset>"
vcli_inactivity off


Nameserver Parameters

nameserver_broadcast: N            rwhod: Y
pref_name1_addr: 192.9.200.95      name_server_1: dns
pref_name2_addr: 192.9.200.85      name_server_2: dns
host_table_size: 64                min_unique_hostnames: Y


Security Parameters

enable_security: Y                 security_broadcast: Y
pref_secure1_host:192.9.200.95     pref_secure2_host: 0.0.0.0
network_turnaround: 2              loose_source_route: Y
acp_key: "<unset>"                 password: "<set>"
allow_snmp_sets: N                 lock_enable: Y
passwd_limit: 3                    chap_auth_name: "chap"
max_chap_chall_int: 0
```

(continued on next page)

```
Time Parameters

time_broadcast: N                  daylight_savings: us
timezone_minuteswest: 300          time_server: 0.0.0.0


SysLog Parameters

syslog_mask: all                   syslog_facility: log_local7
syslog_host: 192.9.200.95          syslog_port: 0

MOP and "Login" User Parameters

pref_mop_host: 00-00-00-00-00-00
mop_password: “<unset>”            login_password: “<set>”
login_prompt: “#”                  login_timer: 30


LAT Parameters

lat_key: ""                        facility_num: 0
server_name: ""                    sys_location: ""
lat_queue_max: 4                   service_limit: 256
keep_alive_timer: 20               circuit_timer: 8
retrans_limit: 8                   group_value: none
vcli_groups: none                  multicast_timer: 30
multisessions_enable: N


AppleTalk Parameters

a_router: 00-00-00-00-00-00
default_zone_list: ""              node_id: 0.0
zone: ""


Router Parameters

rip_auth: "<unset>"                rip_routers: all


IPX Parameters

ipx_file_server: ""                ipx_frame_type: raw802_3
ipx_dump_username: ""              ipx_dump_password: "<unset>"
ipx_dump_path: ""                  ipx_do_checksum: N


TMux Parameters

tmux_enable: N                     tmux_max_host: 64
tmux_delay: 20                     tmux_max_mpx: 700
```

(continued on next page)

```
DHCP Parameters
pref_dhcp1_host: 0.0.0.0           pref_dhcp2_host: 0.0.0.0
dhcp_bcast: N
```

5. **Execute either** boot **or** reset annex all **to effect these changes at the RA 6300.**

Local File System

The stand-alone file system allows the RA 6300 to store its configuration and message-of-the-day files in local non-volatile memory. The configuration files must have the appropriate file names for the operational image to locate and load them. These files exist in the **root** directory rather than the **/usr/spool/erpcd/bfs** directory. The files are manipulated using the CLI local file system commands.

RA 6300 Internet Addressing

The RA 6300 uses Internet addressing to communicate with hosts on the network. Internet support requires an Internet address, a broadcast address, and a subnet mask.

The Internet Address

The RA 6300's Internet address is defined in the **inet_addr** parameter. This address must be set prior to downloading the operational code to the RA 6300. To do so, use the ROM monitor **addr** command during the RA 6300's initial installation. You can reset the address at any time thereafter by changing the **inet_addr** parameter.

The CLI, **na**, and ROM Monitor commands always display the Internet address in dotted decimal notation.

The Broadcast Address

The broadcast address defines the Internet address the RA 6300 uses to broadcast. The RA 6300 will broadcast requests when it has not received a response from a server, such as file server or security server. The **broadcast_addr** parameter defines this address.

The Subnet Mask

If the network is divided into subnets, you must specify the RA 6300's Internet subnet mask using the **subnet_mask** parameter. If you do not define the subnet mask, the RA 6300 assigns one based on the network part of its Internet address. Set this parameter using the ROM Monitor **addr** command during the RA 6300's initial installation. You can reset the address at any time thereafter by changing the Annex **subnet_mask** parameter.

Certain combinations of the RA 6300 subnet mask and Internet address have special meaning:

- Setting the RA 6300 Internet address to 0.0.0.0 or 255.255.255.255 turns off all IP services, including SLIP, PPP, and IP routing. The RA 6300 continues to support non-IP services, such as ARAP and LAT, provided that they are configured properly.
- Setting the RA 6300 Internet address to a valid value and RA 6300 **subnet_mask** to 255.255.255.255 installs IP but specifies the RA 6300 does not have an Ethernet connection. IP services, including SLIP, PPP, and IP routing, are still available.
- By default, the RA 6300 acts as an authoritative agent for ICMP Address Mask Requests. If another host broadcasts this message querying for the subnet mask, the RA 6300 replies with the subnet mask. Optionally, you can prevent the RA 6300 from responding by setting the **authoritative_agent** parameter to **N**.

Booting and Dumping

The RA 6300 obtains its operational code by downloading it over the network from a UNIX host that runs RA 6300 file server software, a non-UNIX host running **tftp**, another RA 6300 configured as a boot server (running the same operational code), or the local media (self-boot). The RA 6300 boots each time it is powered up and upon receipt of a **boot** command.

The RA 6300 can dump to a file server or a host running **tftp**. The RA 6300 performs a dump upon receipt of either the **na** command **dumpboot** or the superuser CLI **boot –d** command, or automatically when it detects fatal internal errors or failures.

Setting the Preferred Load Host

The **pref_load_addr** parameter specifies the preferred load (or file server) host. This is the host from which the RA 6300 first requests a down-line load of its operational code. If this parameter is not defined or the specified host is not available, the RA 6300 broadcasts its boot request and loads operational code from the first host that responds. You can modify the **pref_load_addr** parameter using **na** or the **admin** command; specify the host by its Internet address or its name.

The **image_name** parameter specifies the name of the image file that contains the Annex’s operational code. This file resides in different host directories, depending on which transfer protocol (**tftp** or **erpcd**) is used.

If the load host has a different network or subnet address, you must define a gateway through which the Annex can reach the host. The **load_dump_gateway** parameter specifies the Internet address for the gateway.

During the initial boot of the operational code, the ROM Monitor requires the address of a gateway if the specified load host is on another network or has a different subnet address. In this case, enter the gateway’s address using the ROM Monitor **addr** command. The Annex automatically adds this gateway to its routing table.

Setting the Preferred Dump Host

The **pref_dump_addr** allows you to specify the preferred host to which the Annex performs a dump. If this parameter is not defined or the specified host is not available, the Annex broadcasts its dump request and dumps to the first host that responds.

The dump creates a file that is between one and three megabytes in size. If using **erpcd**, the Annex assigns the dump file a unique name and places it in a directory named **/usr/spool/erpcd/bfs**. If using **tftp**, the file name is defined by the **tftp_dump_name** parameter and file placement is user-defined. If the dump host has a different network or subnet address, you must define a gateway through which the Annex can reach the host. The **load_dump_gateway** parameter specifies the Internet address for the gateway.

Setting the Load-Dump Sequence

The configuration parameter **load_dump_sequence** specifies the network interface and the order to be used for a down-line load or an up-line dump. The arguments are **net** (for use with a LAN), and **self** (to boot from the local media). For more details, see *load_dump_sequence* on page C-68.

Setting an Annex as a Load Server

The **server_capability** parameter defines the Annex as a file server host. An Annex can provide operational code only for another Annex of the same type. When an Annex boots, it uses the **image** file to load the operational code, and the configuration file to initialize the routing table, rotaries, and macros. The Annex normally does not store these files because they use memory. As a file server host, the Annex uses approximately 120 Kbytes for the operational code; for the message-of-the-day (**motd**) and configuration files, it uses the amount of space relative to the size of the files.

The **server_capability** parameter defines the files that the server supplies during a boot. Table A-1 describes the arguments for **server_capability**; the default is **none**.

If you configure an Annex to supply only a copy of the operational code, the default is for the Annexes being booted to broadcast for the configuration and **motd** files. The file server Annex uses **erpcd** to serve other Annexes.

Table A-1. Arguments for the server_capability Parameter

Argument	Description
all	The Annex is a file server for the configuration, operational image, and message-of-the-day files.
config	The configuration files.
image	The operational code.
motd	The message-of-the-day file.
none	The Annex is not a file server.

Disable Broadcasting for Files During a Boot

During a boot, the Annex broadcasts for the configuration, **image**, and **motd** files if they are not available on the preferred load host. You can disable broadcasting for these files by setting the **load_broadcast** parameter to **N**.

Self Booting

The self-boot option loads and boots the operational image from local non-volatile memory. To store the image into the local media, issue the **boot –l** command from **na**, the superuser CLI, or the ROM monitor.

Only ROM revisions 0601 and greater with the self-boot option installed support the **boot –l** command.

After executing a **boot –l** command, the **ls** command may not show the newly-loaded image.

To boot the stored (local) image, set the configuration parameter **load_dump_sequence** (or the ROM monitor parameter **sequence**) to **self** and reboot. This sequence instructs the Annex to load the operational image and the configuration file from the local media.

To boot from both the local media and the network, set **load_dump_sequence** to either **self**,**net**, or **self**. The Annex will first load the files from the local media; whatever files it cannot find there it will seek from the network.

Using the Trivial File Transfer Protocol

The Trivial File Transfer Protocol (**tftp**) is a standard network interface loading program. The Annex operational code opens and reads the operational image, configuration, and **motd** files. The Annex accesses one file at a time.

The Annex initially tries to open a file using **erpcd** (except when using the self-boot option). If **erpcd** fails or times out, the Annex tries to open a file using **tftp**. If the **tftp** request fails or times out, the Annex retries opening the file using **erpcd**. This cycle continues until the Annex succeeds in opening the file or until the it reaches a maximum try count (currently 8 cycles). If the **load_broadcast** parameter is enabled and the Annex cannot open a file from the **pref_load_host**, it broadcasts the open request (this is true for both **erpcd** and **tftp**). Once a file is successfully opened, the Annex continues to read it using the protocol with which it was opened.

The protocol used to transfer one file is independent of the protocol used to transfer another file. For environments that support both **erpcd** and **tftp**, the Annex may use **tftp** to transfer one file and **erpcd** to transfer another file.

Using Annex Security

The Annex provides a security system that allows you to implement as many security measures as the network requires. You can set up the security subsystem to use host-based security, local password protection, or a combination of the two. In addition to these security mechanisms, the Annex provides an administrative password that validates access through the administrative tools.

If unauthorized users can access your Annex, we strongly suggest that you enable the security features after loading the host code and booting the unit.

For a detailed description of Annex security, see *Using RA 6300 Security* on page A-211.

Using Name Servers

Name servers allow users to enter names in place of addresses in order to access a host or other entity on the network. The Annex supports two standard types of name servers: a Domain Name System (DNS) server and IEN-116 server. In addition, the Annex can use RWHO broadcast messages to provide name-to-Internet address translation. You can configure the Annex to use one of these, a combination, or none.

The Annex supports the minimum uniqueness feature when entering host names. This feature allows users to enter the host name with a minimal string that is unique enough to identify that host from any other in the host table. If this feature is not enabled, the user must enter the complete name to access a host. Host name to Internet address translation entries can be downloaded to the Annex from the **gateway** section of the configuration file. The format is the same as in the **/etc/hosts** file, but aliasing is not permitted. To set up an Annex for use with a name server:

- Specify the name server type.
- Specify the host(s) using the name server.
- Enable or disable the **rwhod** parameter.
- Specify the host table size.
- Enable or disable the **min_unique_hostnames** parameter.

Defining Name Servers

The Annex supports two standard name server protocols: Domain Name System (DNS) server and IEN-116 server. Both of these name server protocols are available in the UNIX environment. You can use one or both on the network, and the Annex allows you to specify the preferred protocol. If you choose not to use either protocol, you can configure the Annex to build the host table by listening to RWHO broadcasts.

Domain Name System

Domain Name System (DNS) servers use a distributed database to maintain host names and Internet addresses for network hosts. DNS provides a full range of capabilities that enable its use in very large networks, such as the Internet.

Each DNS server is responsible for maintaining information on all hosts in its domain. If the server receives a request for a host that is not in its domain, the server retrieves the information from another domain server for the requesting host.

A number of DNS servers are available and the Annex can support them all. One typical DNS server is the Berkeley Internet Name Domain (BIND) server. The BIND server is a standard part of 4.3BSD (see 4.3BSD documentation for more details). DNS provides:

- address to name translation.
- multiple aliases for a host.
- multiple addresses for the same host.

Address to name translation allows a host to obtain a name for a specific Internet address, allowing an Annex to learn its name from a DNS server. The DNS' capabilities for assigning multiple aliases or multiple IP addresses to a single host allow you to assign multiple names to a rotary or multiple Annexes to the same rotary (for more details, see *The Port Server and Rotaries* on page A-77).

IEN-116 Name Server

The IEN-116 name server is a simple host-resident name server that uses the local **/etc/hosts** file as a database. One host is designated as the name server host, and other hosts query that host for an address. Using this method, every host on the network does not need its own up-to-date **/etc/hosts** file, and every host does not have to run **rwhod**. The Annex distribution medium supplies the source for IEN-116 (see *Configuring Hosts and Servers* on page A-209 for installation instructions).

IEN-116 name servers cannot do reverse address queries.

Setting Configuration Parameters

The **name_server_1** parameter defines the type of name service that the primary name server will supply when queried by the RA 6300. Valid service types are **dns**, **ien_116**, or **none**; the default is **none.** You specify the Internet address of the primary name server by setting the **pref_name1_addr** parameter.

The **name_server_2** parameter defines the type of name service that the secondary name server will supply when queried by the RA 6300. Valid service types are **dns**, **ien_116**, or **none**; the default is **none**. You specify the Internet address of the secondary name server by setting the **pref_name2_addr.** This server is queried only if **pref_name1** server does not respond.

Broadcasting for a Name Server

By default, the Annex does not broadcast for a name server if the preferred name servers do not respond. However, you can configure the Annex to broadcast requests for a name server by setting the **nameserver_broadcast** parameter to **Y**. This causes the Annex to broadcast three requests for a Domain Name Server, followed by three requests for an IEN-116 name server. You may want to use broadcast as a back-up for a name server.

Using the RWHO Protocol

Berkeley UNIX hosts use the RWHO protocol to pass information about themselves to other hosts. This information includes the host’s name, who is logged in, up time, and load factor. The RWHO daemon, **rwhod**, broadcasts this information and listens for RWHO messages from other hosts, storing what it receives in a file. The information can be displayed with the **rwho** and **ruptime** commands from a UNIX host.

The Annex uses the RWHO protocol as a name server. The Annex runs an **rwhod** that listens for broadcasts from other hosts, but does not broadcast information about itself. When the Annex receives an RWHO message, it stores the host name, status information, and the source address from the IP header as the host's Internet address in its host table.

Using only RWHO messages to build the host table is satisfactory for a small network in which all the hosts run **rwhod**. But, **rwhod** often is not used in networks primarily comprised of workstations because of the load it imposes on hosts. In large or heavily loaded networks, RWHO broadcasts can impose an excessive load on the network.

Some hosts send RWHO packets with incomplete source addresses in the IP header. The Annex is unable to store an Internet address for these hosts, causing the host table to display the host's Internet address as "_._._._".

If an **rwhod** forwards packets from one network to another, the Internet address in the IP header is that of the forwarding host, not of the host whose name is in the data packet. This results in the Annex storing the wrong Internet address for that host.

Because the Annex does not broadcast RWHO messages, Annex names never appear in host tables built exclusively from these broadcasts. In which case, the only way to access an Annex using the **telnet** command is with an Internet address.

The **rwho** parameter defines whether or not the Annex listens for RWHO broadcasts. Setting the parameter to **N** disables the Annex's **rwhod** and prevents the Annex from using RWHO messages for building the host table. The default is **Y**.

Managing the Size of the Host Table

When the host table acquires a new entry after it is full, the Annex deletes the oldest, least-used entry to make room for the new one. The Annex's use of the host table is erratic if the table size is too small. Increasing the size of the table reduces this problem.

You modify the host table size using the **host_table_size** parameter. This parameter specifies the number of entries in the host table. You can specify the size as a number from **1** to **250**. Specifying the string **"unlimited"** sets no limit other than the size of memory available in the Annex. Alternatively, you can set the size to **"none"**, which forces the Annex to query the name server for each host name.

Minimum Uniqueness

Minimum uniqueness provides an ease-of-use feature, which allows users to enter only the characters necessary to uniquely match an entry in the host table. However, users can force the Annex to select only an exactly matching host name by enclosing the name they enter in double quotes. For example:

```
annex: rlogin "widget"
```

If the host table contains the name *widgetslips*, and you want to log in to a host named *widget*, which is not in the host table, then entering *widget* without the quotes causes the Annex to select *widgetslips*. Entering the name enclosed in double quotes forces the Annex to query a name server, because an exactly matching name is not in the host table. The minimum uniqueness feature can be turned off entirely by setting the **min_unique_hostnames** parameter to **N**.

Using Event Logging

The Annex can log events to a 4.3BSD system log daemon (**syslogd**) on the system you specify via the **syslog_host** parameter. The Annex may be able to log events to a 4.2BSD system using the **syslog** daemon or to a System V if it has system logging similar to 4.3BSD syslogging.

The 4.3BSD system logging daemon provides a *facility* as an addition to the *selector* field. The selector field is a list of priorities for a message and includes a level, which indicates the severity of a message. The facility defines the part of the system that generates the message. Certain facilities are reserved, such as kernel, mail, and daemons; other facilities can be defined in the configuration file **/etc/syslog.conf**. Facilities allow you to selectively log messages by priority.

If the **syslog_host** does not have a **syslog** daemon, or if you do not specify a **syslog_host**, the Annex logs events to the RA 6300 console.

When configuring the host and the Annex for system logging, consider the following parameters: **syslog_host**, **syslog_facility**, and **syslog_mask**. (Reboot the Annex after configuring any parameters related to system logging.)

- The **syslog_host** parameter defines the Internet address of the host configured to log Annex messages. The default, **0.0.0.0**, causes the Annex to broadcast its log messages.
- The **syslog_facility** parameter defines the facility used in the syslog messages (specified as **log_local***n* where *n* is a number from 0 through 7). The default is **log_local7**.

If the host to which messages are logged does not support 4.3BSD syslogging, this parameter is ignored and messages are logged only by priority level as defined in the **syslog_mask** parameter.

- The **syslog_mask** parameter defines the priority levels for logging messages. The options are **all**, **none**, or a combination of levels. The default, **none**, disables logging. Table A-2 describes the levels in priority order.

When defining a priority level, all messages of that level or greater (i.e., of greater severity) are forwarded to **syslogd**. For example, selecting **error** logs all **error**, **critical**, **alert**, and **emergency** messages.

Table A-2. Priority Levels for the syslog_mask Parameter

Level	Description
emergency	Hardware failures.
alert	All Annex reboots.
critical	Configuration and initialization problems, such as format errors in the **gateway** section of the configuration file or lack of memory.
error	All line initialization errors, including CLI.
warning	Indications of minor problems.
notice	Time server queries and information about responses.
info	Starting and ending of CLIs and of Annex jobs created by the **rlogin** and **telnet** commands and the **ping** and **tap** superuser CLI commands.
debug	Activation and exit of all Annex processes.

Using the Time Server

The Annex maintains a UNIX-style time-of-day clock, which is based on the Internet date and time server. The Annex distribution includes source code for a time server in case one is not available on the preferred load host. The Annex synchronizes its clock by requesting the time from a time server.

The time server expresses time in the number of seconds since midnight (00:00:00), January 1, 1970, Greenwich Mean Time (GMT). The Annex converts time server time to local time and uses it to log events to **syslog** and to calculate the time of a boot and/or dump. The CLI **stats** and **who** commands display this time; the local file system **ls** command displays the time the files were last modified.

The Annex requests the time when it boots and synchronizes its clock with a server every 30 minutes. It always queries the preferred load host first if one is defined. If a time server does not respond, the Annex displays *unknown* in place of its time.

By default, if a time server is not available on the preferred load host, the Annex does not broadcast for the time. However, you can enable broadcasting for a time server by setting the **time_broadcast** parameter to **Y**. Most UNIX systems provide a time server with the **inetd** daemon.

Every host on the network that has a timer server will respond to a broadcast for the time.

The Annex does not reset its time by more than 10 minutes based on an answer to a broadcast request. If the time returned to the broadcast query was greater than 10 minutes from the Annex's current time, the Annex only resets its time by a maximum of 10 minutes. If the timer server is on the preferred load host, the Annex adjusts to the time reported by the time server, regardless of the time interval.

The **timezone_minuteswest** parameter defines the time zone in which the Annex resides. Enter a positive number of minutes for time zones west of GMT and a negative number for time zones east of GMT. For example, since U.S. Eastern Standard Time is five hours west of GMT, its value is 300 minutes; since Paris is one hour east of GMT, its value is -60 minutes.

The **daylight_savings** parameter defines the daylight savings time to which your geographic area adheres. The Annex uses this parameter to adjust the time display for daylight savings time. Valid arguments include: **us, australian**, **british**, **canadian**, **east_european**, **mid_european**, **west_european**, or **none**.

Customizing the Annex Environment

You can customize the following Annex attributes:

- The prompt that displays when a user accesses the CLI.
- The number of simultaneous virtual CLI connections.
- The name of the configuration file.
- The name of the message-of-the-day file.

- RIP.
- The type of IP encapsulation used by the LAN.
- TSTTY.
- TMux.
- LAT.
- AppleTalk.
- IPX.

Setting the CLI Prompt

The Annex displays a prompt when a user accesses the CLI. The **cli_prompt** parameter allows you to customize the Annex prompt. You can also customize the prompt for each serial port using the **prompt** port parameter (for more details, see *cli_prompt* on page C-47).

The values for this parameter are called prompt strings. A prompt string consists of characters and embedded formatting codes that are expanded when the prompt is displayed. The formatting codes consist of a percent character (%) followed by a single lower-case character. Each formatting code occupies one character in storage. You can also specify a string for the prompt using these codes. The default is **%a%c** (*annex:*). Table A-3 describes the codes for the prompt string.

Table A-3. Formatting Codes for Annex Prompts

Code	Expansion
%a	The string *annex*.
%c	A colon followed by a space.
%d	The current date and time in standard UNIX format, such as Mon Mar 14 13:59:42 1989.
%i	The Annex's Internet address, such as 132.245.6.40.
%j	A new line character, skip to the beginning of the next line.
%n	The Annex's name or Internet address, such as 132.245.6.40.
%r	The string *port*.
%s	A space.
%t	The current time in 24-hour format, such as 13:59:42.
%u	The user name defined for the port; if none, a null string.

If you want a prompt to appear as *date and time* (new line) *annex-name*, use the following code:

```
%d%j%n%s%c
```

For the port on the Annex named *thirdfloor*, the prompt is:

```
Mon Jan 6 11:10:25 1997
thirdfloor
```

For the superuser CLI prompt, a pound sign (#) and a space replace the code **%c**; otherwise a # is appended at the end.

Setting a Limit on Virtual CLI Connections

The number of virtual CLI connections at an Annex can affect the use of memory, as each virtual CLI connection uses memory. The **max_vcli** parameter determines the maximum number of virtual CLI connections the Annex can create at any one time. You can set the number of virtual CLI connections from an unlimited number to none. The range of values that you can enter are from **0** to **254** or **“unlimited”**. The default is **“unlimited”**. If you define this parameter as zero, users cannot create a virtual CLI connection at the Annex.

Setting Up the Configuration File

The configuration file contains all Annex configuration information. It resides either on the preferred boot host or the local media and is loaded during the Annex booting process (see *Configuring Hosts and Servers* on page A-209 for more details on creating and using the configuration file).

You can define a name for the configuration file using the configuration parameter **config_file**. The default file name is **config.annex**.

You can create these files on the local media using the superuser CLI **edit** command.

Setting the motd File

The Annex can display a message-of-the-day at the CLI prompt after it has been rebooted or reset, or the port has been reset, or each time a user accesses the Annex through a virtual CLI connection.

The default file name is **motd**. The **motd_file** parameter allows you to specify another name for this file. The Annex reads this host file each time it is booted, and when the **na** or **admin** command **reset annex motd** is issued.

Using RIP

The Annex uses a routing daemon (**routed**) for its routing services. This daemon implements Versions 1 and 2 of the Routing Information Protocol (RIP).

The **routed** parameter enables or disables RIP; the default is **enabled**. If RIP is disabled, the Annex does not listen for or transmit RIP routing updates. Instead, it depends on the routing information in the **gateway** section of the configuration file. If you disable RIP, define a default route in the configuration file.

Setting the IP Encapsulation Type

The Annex supports two types of LAN encapsulation of IP packets: Ethernet Version 2 format or IEEE 802.3 Data Link Layer format. The **ipencap_type** parameter specifies which type of encapsulation to use; the default is **ethernet**.

This parameter should be changed only at installation time using the ROM Monitor. Do not change this parameter using **na** or **admin** because the Annex cannot boot with the wrong IP encapsulation.

Using the Terminal Server TTY (TSTTY)

TSTTY is a set of independent software modules that allow a host system to connect to Annex serial ports in such a way that users appear as if they are directly connected to the host system. One module runs in the Annex and one module runs in the host. A protocol links the two modules together. When a host wishes to talk to a device attached to a port that is in slave or adaptive mode, it must first establish a connection by connecting to the appropriate TCP port on the Annex. The Annex and host can then send messages over this link to exchange data and commands. (TSTTY runs on top of any reliable byte stream protocol, e.g., TCP.)

By providing a standard *tty* interface to the host, all standard programs can access the ports through standard serial port devices, and hence perform all of the functions that a standard, directly connected port can perform.

Using the Transport Multiplexing (TMux) Protocol

The TMux protocol provides an open, standards-based solution to the CPU overload problem associated with TCP/IP terminal servers. Unlike **telnet** and **rlogin**, which generate lots of small packets, the TMux protocol multiplexes the small TCP packets generated by any number of **telnet**, **rlogin**, and TSTTY connections from an Annex to a host system into a single IP network packet. Since the system load is determined per packet, not per byte, multiplexing this single packet from one system to another significantly reduces the host overhead.

Configuring LAT Services

The Annex can display, and connect to, currently available LAT services. Initially, all LAT functions in the Annex are disabled since this feature is optional. To enable the LAT functions, the network administrator must enter the correct **lat_key** parameter value and reboot the Annex (see *Configuring Hosts and Servers* on page A-209 for more details).

The **lat_key** parameter value is unique for each Annex. If you purchased LAT, contact Xylogics to obtain your key.

Configuring the Annex for AppleTalk

Initially, all AppleTalk functions in the Annex are disabled since this feature is optional. To enable the AppleTalk functions, the network administrator must enter the correct **option_key** parameter value and reboot the Annex (for more details, see *AppleTalk* on page A-183).

The **option_key** parameter value is unique for each Annex. If you purchased AppleTalk, contact your supplier to obtain a valid key.

Configuring IPX

Initially, all IPX functions in the Remote Annex are disabled since this feature is optional. To enable the IPX functions, the network administrator must enter the correct **option_key** parameter value and reboot the Annex (see *Internetwork Packet Exchange (IPX) Protocol* on page A-153 for more details).

The **option_key** parameter value is unique for each Annex. If you purchased IPX, contact your supplier to obtain a valid key.

Chapter 3 Configuring the PRI Interface, Global Ports, and Sessions

This chapter replaces Chapter 3 in Book A of the *Remote Annex Administrator's Guide for UNIX*, which describes configuring ports for other Remote Annexes. This chapter describes:

- Delivering ISDN calls to the Remote Annex 6300.
- Configuring the PRI interface.
- Port handling on the RA 6300.
- Configuring global port parameters.
- Configuring Session Parameter Blocks (SPBs) to handle incoming calls.
- Understanding the differences between the RA 6300 as described in this chapter and other Remote Annexes (as described in the corresponding chapters of the *Remote Annex Administrator's Guides for UNIX*). This section is intended for administrators who are accustomed to (or will be) configuring other Remote Annexes.

Delivering ISDN Calls

An ISDN PRI call can arrive on any B channel on the RA 6300 PRI line. During the call SETUP process between the telephone company switch and the RA 6300, the switch dynamically assigns a B channel to the call. The SETUP process takes place on the D (signaling) channel.

Since the RA 6300 cannot predict which B channel the switch assigns to a particular call, providing the RA 6300 with a static configuration for each B channel is not useful. Instead, the RA 6300 relies on PRI parameters, global port parameters, and Session Parameter Blocks (SPBs) to dynamically obtain parameter values for a call received on a B channel.

Configuring the PRI Interface

Once you have ordered your PRI line from the telephone company and have installed the 6300 hardware and software, you are ready to set generic parameters for the PRI interface. The following superuser CLI **admin** command displays these parameters and their default values:

```
annex# admin
ANNEX-PRI Rx.x, 24 async, 32 sync, 32 ta, 24 modem ports
admin: show pri
     PRI Generic Parameters

     switch_type: ""                num_b_channels: 0
     dsx1_line_length: 0-25         buildout: 0dB
     fdl_type: att                  analog_encoding: auto
admin:
```

To reset their default values, execute the **admin** command or the host-based **na** utility, using the following syntax:

set pri *parameter value*

Table A-4 describes the parameters.

The following example uses **admin** to reset the **switch_type** parameter:

```
admin: set pri switch_type dms
```

Table A-4. Parameters for the set pri Command

Parameter	Description
switch_type	A string specifying the type of switch provided by the telephone company for your PRI line. Valid values are: AT9 (for the AT&T 5ESS switch) DMS (for Nortel’s DMS100 switch) NI2 (for a switch supporting National ISDN2) ETS (for Euro ISDN, NET5 conformance) The default is "", a blank string. The RA 6300 interprets this as a switch type appropriate for the line: With a T1 line, AT9 is used. With an E1 line, ETS is used. The switch type parameter is not case-sensitive.
dsx1_line_length	The approximate distance, in meters, from the RA 6300 PRI interface to the external CSU. Enter this as a range, e.g., **0–25** (the default), as a single number, e.g., **30**, or as a unique part of a range, e.g., **136–**. If you enter a single number, the RA 6300 picks the range that the number falls into. (You can then use the **pri show** command to determine the range assigned.) Valid ranges are **0–25**, **26–65**, **66–100**, **136–185**, **166–185**, and **186–210**.
fdl_type	The type of Facilities Data Link supported by the telephone company for your PRI line. Valid values are **att** and **ansi**. This parameter is applicable only to internal CSUs used with T1/PRI lines. To display statistics for an internal CSU, see *Displaying RA 6300 Statistics* on page B-41.

(continued on next page)

Table A-4. Parameters for the set pri Command (continued)

Parameter	Description
num_b_channels	The maximum number of B channels that can be used with the RA 6300. The default is **0**, which the RA 6300 interprets as **23** for T1/PRI connections (in the U.S.) and **30** for E1/PRI connections (in Europe and Australia). Valid values are **1–23** for T1/PRI and **1–30** for E1/PRI. The only time you might want to change the default is in Europe, where telephone company providers support PRI connections with fewer than 30 channels (which can be cheaper).
buildout	(Applies only to RA 6300s with internal CSUs). A string defining the CSU transceiver line provided by the telephone company. Valid values are: **0db** (the default) **7.5db** **15db** **22.5db**
analog_encoding	The encoding type used for modem calls. Valid values are: **a_law** (used for E1/PRI lines in Europe) **mu_law** (used for T1/PRI in the U.S.) **auto** (the default, which uses **a_law** or **mu_law** as appropriate) Typically, you do not need to change this parameter. To check that the correct value is being used, issue the **pri** command from the superuser CLI.

Port Handling

Unlike other Remote Annexes, the Remote Annex 6300 uses only two external physical interfaces – an Ethernet port (referred to as **en0**) and the PRI interface. However, the RA 6300 assigns *internal* ports to match the three types of data ISDN can carry on a B channel. You do not explicitly set port parameters for these internal ports. Instead, you configure global port parameters (call defaults) and Session Parameter Blocks (SPBs), as follows:

- The RA 6300 comes with a set of *global* port parameters set to default values. The RA 6300 applies these parameters to the appropriate internal port while a call is active on that port. You can override the global port parameter defaults by resetting them using **na** or **admin**.
- The same set of port parameters that you can modify globally can also be reset for one PRI call or subset of calls. To do this, you include the parameter definition in a Session Parameter Block (SPB) that handles the PRI call(s). Parameter values set in SPBs override the global settings. See *ISDN Sessions and Session Parameter Blocks (SPBs)* on page 3-64.

At least one SPB must be defined in order for the RA 6300 to handle any PRI calls.

Internal Ports

The internal RA 6300 ports are dynamically numbered within a given range (starting at 1) for each of the three data types. The numbers are assigned in a circular fashion. When the maximum number in the range has been assigned, the next incoming call is given the first free port number, starting over at 1. The port number maps to the internal data-handling resource; it has no *static* association with the B channel on which the data arrived.

The internal ports are of the following types:

- TA ports manage V.120 calls. The range of port numbers is 1 through 32, the ports are referred to as ta1, ta2, ..., ta32, and calls are mapped to port numbers in the circular fashion described above.
- Synchronous ports manage synchronous PPP calls. These are referred to as syn1, syn2, ..., syn32. Numbers are assigned in the circular fashion described above.
- Asynchronous ports manage voice calls. The port numbers assigned to them range from 1 to the number of installed internal modems. Numbers are assigned in the circular fashion described above.

  If no modems are installed, no internal asynchronous ports are supported. The asynchronous internal port numbers are referred to as asy1, asy2, ..., asy*n*, where *n* is the total number of installed modems. If the maximum number of modems (32) are installed, the port range is 1 to 32, even though only 23 calls (30 in Europe) can be handled simultaneously.

Global Port Parameters

All global port parameters have default values. The **na** utility and the superuser CLI **admin** command allow you to use the **set port** command to override the global defaults. When resetting the global port parameters, modify only the values:

- whose defaults differ from your requirements.
- whose differing defaults apply to all PRI calls. To specify a port parameter that applies only to a subset of calls, define that subset in a Session Parameter Block (SPB) and include the parameter setting in that SPB. Parameter values set in an SPB override global parameters set by **na** or **admin** (which in turn override the supplied defaults); see *ISDN Sessions and Session Parameter Blocks (SPBs)* on page 3-64.

Displaying Global Parameters

Both **na** and **admin** provide a **show port** command to display the current global port parameter values. Table A-5 on page A-58 describes the keywords for the **show port** command. Sample displays of **show port all** and **show port sync** are shown following the table. These displays show the factory defaults.

Table A-5. Keywords for the show port Command

Keyword	Description
all	Displays all global port parameters values.
appletalk	Displays the global Appletalk parameters.
editing	Displays the global CLI line editing parameters used with terminal emulation.
flow	Displays the global flow control parameters.
generic	Displays the basic global parameters.
ipx	Displays the global IPX parameters.
lat	Displays the global LAT parameters.
ppp	Displays the global PPP parameters.
security	Displays the global security parameters.
serial	Displays the global serial parameters.
slip	Displays the global SLIP parameters.
syn	Displays the global synchronous PPP parameters.
timer	Displays the global timer parameters.
tn3270	Displays the global tn3270 parameters.
vci	Displays the global VMS command interface parameters.

The following example uses the **admin** command **show port all** to display all asynchronous and TA global port parameters.

```
annex#: admin
admin: show port all
global port:

Port Generic Parameters
mode: cli                       location: ""
term_var: ""                    prompt: ""
cli_interface: uci              data_bits: 8
stop_bits: 1                    parity: none
max_session_count: 3            allow_broadcast: Y
broadcast_direction: port       imask_7bits: N
cli_imask7: Y                   banner: Y
tcp_keepalive: 0                default_session_mode:
interactive
dedicated_arguments: ""         resolve_protocol: connect


Flow Control and Signal Parameters
input_flow_control: eia         input_start_char: ^Q
input_stop_char: ^S             output_flow_control: eia
output_start_char: ^Q           output_stop_char: ^S
ixany_flow_control: N           need_dsr: N
v120_mru: 256
Port Timers and Counters
forwarding_timer: off           forwarding_count: 0
cli_inactivity: off             inactivity_timer: off
input_is_activity: Y            output_is_activity: N
reset_idle_time_on: input       long_break: Y
short_break: Y                  autodetect_timeout: 30


Port Security Parameters
user_name: ""                   cli_security: N
connect_security: N             port_server_security: N
port_password: "<unset>"        ipso_class: none
ipx_security: N
```

(continued on next page)

```
"Login" User Parameters

login_port_password: N          login_timeout: N

CLI Line Editing Parameters

attn_string: ""                 echo: Y
telnet_escape: ^]               telnet_crlf: N
map_to_lower: N                 map_to_upper: N
char_erase: Y                   line_erase: Y
hardware_tabs: Y                erase_char: ^?
erase_word: ^W                  erase_line: ^U
redisplay_line: ^R              toggle_output: ^O
newline_terminal: N             forward_key: ""
backward_key: ""

Serial Networking Protocol Parameters

local_address: 0.0.0.0          metric: 1
slip_ppp_security: N            net_inactivity: off
do_compression: N               allow_compression: N
net_inactivity_units:minutesaddress_origin:local

SLIP Parameters

subnet_mask: 0.0.0.0            slip_mtu_size: small
slip_no_icmp: N                 slip_tos: N

PPP Parameters

ppp_mru: 1500                   ppp_acm: 0x0
ppp_security_protocol: none     ppp_username_remote: ""
ppp_password_remote:"<unset>"   ppp_ncp: all
ppp_sec_auto: N                 mp_mrru: 0
mp_endpoint_class: mac          mp_endpoint_address:
```

(continued on next page)

```
Port AppleTalk Parameters

at_guest: N                     at_nodeid: 0.0
at_security: N                  arap_v42bis: Y


Port TN3270 Parameters

printer_host: 0.0.0.0           printer_name: ""


Port LAT Parameters

authorized_groups: none         latb_enable: N
multisessions_enable: N
```

To display the synchronous PPP global port parameters via **admin**, enter the following from the superuser CLI.

```
annex3# admin
ANNEX-PRI Rx.x, 32 async, 32 sync, 32 ta, 32 modem ports
admin: show port syn
global port:
user_name: ““                   port_password: “<unset>”
ppp_username_remote: ““         ppp_password_remote: “<unset>”
slip_ppp_security: Y            ppp_security_protocol: pap
ppp_ncp: atcp,ipcp,ipxcp,mp
metric: 1                       subnet_mask: 0.0.0.0
ppp_mru: 1500                   inactivity_timer: off
input_is_activity: Y            output_is_activity: N
reset_idle_time_on: input       net_inactivity: off
net_inactivity_units:minutes    mp_mrru: 0
mp_endpoint_class: mac          mp_endpoint_address:
```

In the above sample displays, the parameters that begin with the string *mp_* are related to Multilink PPP, which is not described in this document. For information, see the *Multilink PPP Addendum to the Remote Annex Administrator's Guide for UNIX*.

Changing Global Port Parameter Defaults

To change the value of a global port parameter using **na**:

1. **Log into a UNIX host and enter:**

```
% na
Annex Network Administrator Rx.x
command:
```

2. **Specify one Remote Annex 6300 or specify multiple Remote Annex 6300s:**

```
command:annex 132.245.6.40 or
        annex 132.245.6.40,hobbes
        password:
```

3. **Set the global port parameter whose default value you wish to change.**

The following example sets the **allow_broadcast** parameter to **N** (the default is **Y**):

```
command:set port allow_broadcast N
```

4. **Execute the** following **command to review your changes.**

```
command: show port generic
```

```
global port:
```

```
mode: cli                location: ""
term_var: ""             prompt: ""
cli_interface: uci       data_bits: 8
stop_bits: 1             parity: none
max_session_count: 3     allow_broadcast: N
broadcast_direction: portimask_7bits: N
cli_imask7: Y            banner: Y
tcp_keepalive: 0         default_session_mode: interactive
dedicated_arguments: ""  resolve_protocol
```

The changed value is automatically saved in nonvolatile memory and will remain set for subsequent PRI calls and across re-boots of the RA 6300. Currently active calls are not affected.

To configure multiple global port parameters on multiple Remote Annex 6300s:

1. **Define the global port parameters.**
2. **Use the** copy port **command to copy the parameters to other RA 6300 sessions.**

The following example copies the parameter setting for **allow_broadcast** from one RA 6300 to another RA 6300:

```
command: annex 132.245.6.40
command: set port allow_broadcast N
command: copy port 1@132.245.6.40 1@132.245.6.55
```

You can also define all parameters, including global port parameters, for one RA 6300. Use the **write** command to create a script file on the specified UNIX host containing all the configuration data for that RA 6300. Finally, execute the **read** command for all RA 6300s you want to configure.

The **write** command does not write passwords.

For descriptions of the parameters you can set, see *Configuration Parameters* on page C-29.

ISDN Sessions and Session Parameter Blocks (SPBs)

An ISDN PRI session begins when the RA 6300 accepts a call and ends when the call terminates. You define how the RA 6300 handles calls (e.g., accepts or rejects them) by creating one or more *Session Parameter Blocks* (SPBs). You must define at least one SPB for any calls to be handled.

You define SPBs in the **pri** section of the Annex configuration file on the host you use to download RA 6300 software. By default, the file is named **config.annex** and is located in the **/usr/annex** directory.

Each SPB in the **pri** section must be defined within **begin_session** and **end_session** fields. The **begin_session** field allows you to name an SPB within the configuration file.

An SPB consists of three sections:

- An optional section presenting call setup criteria. If the SETUP message that starts an incoming call meets *all* of these criteria, or if no criteria are specified, the call is handled by this SPB.
- A mandatory call handling section that manages all calls meeting the SPB setup criteria. This section must at least specify an action to be taken for the call.
- An optional section containing per-session port parameter settings. All global port parameters can be overridden (as appropriate for the specific type of call) in this SPB section. If you do not need to change the value of a global port parameter, do not specify it here – i.e, if no parameter values are special to this session, you can omit this section of the SPB.

When it receives a PRI call, the RA 6300 tries to match the received SETUP information elements that accompanied the call with the values in the setup criteria section of the defined SPBs. The RA 6300 searches SPBs in the order in which they appear in the configuration file, so the sequence in which you specify SPBs is important. You should order your SPBs from the most specific to the most generic.

When it scans the SPBs, the RA 6300 uses the first SPB whose setup criteria are met by the incoming call. If no match is found, the call is rejected. All criteria in an SPB must be met by the SETUP information elements in order for the RA 6300 to consider the SPB to be a match.

Once the RA 6300 finds a matching SPB setup criteria section for a particular call, it:

- handles the call as specified in the call-handling section.
- uses the per-session port parameter settings to form the dynamic parameter values that will be applied to the call.

Configuring SPBs

Use the following format when entering an SPB into the Annex configuration file. Table A-6 describes all of the possible SPB fields.

```
# this is a comment line

begin_session                  <session_name>
calling_no                     <phone number>
called_no                      <phone number>
called_subaddress              <number>
bearer                         <voice or data>
call_action                    <action>
max_number_of_calls            <integer>
acp_log                        <yes or no>
rate56k                        <yes or no>
set                            <parameter_name setting>
end_session
```

Table A-6. SPB Field Definitions

Field	Definition
begin_session	Marks the beginning of an SPB and names it. The session name is an alphanumeric string of up to 12 characters. (The RA 6300 will accept longer strings, but 12 is the recommended limit.) You can use this string with the CLI superuser **sessions** command to display an SPB.
calling_no	(Optional) Specifies the telephone number that identifies the origin of the ISDN call. ***Specify the entire number, including the area code, even if it would not normally be required to make the call***. You can use a dash to separate the area code from the rest of the phone number, or you can include the area code in parentheses. No wild cards symbols (*) are permitted and white space is ignored. If this field is omitted, any calling number is permitted in the corresponding call SETUP message. Sometimes, the calling number is not available in the SETUP information, either because the phone company did not have the equipment to deliver it or because the caller keeps his or her number private. If *calling_no* is specified, but no number is contained in the SETUP information, the SPB is not a match.
called_no	(Optional) Specifies the number the user entered to dial into the RA 6300. ***Specify the entire number, including the area code, even if it would not normally be required***. You can use a dash to separate the area code from the rest of the phone number, or you can include the area code in parentheses. No wild cards (*) are permitted. White space is ignored.

(continued on next page)

Table A-6. SPB Field Definitions (continued)

Field	Definition
called_no (continued)	Typically, you use this number to distinguish the services required for different types of calls. For example, you can have modem users dial into one number, V.120 users dial into another number, and synchronous PPP users dial into a third number. (You would put each number in its own SPB.) Note: The ACP log file shows the called number delivered by the switch, which may only contain the final digits of the number. If this field is omitted, any called number will match this SPB.
called_subaddress	(Optional) This field is appropriate only for end-to-end ISDN calls using a PRI line that the telephone company has provisioned for subaddressing.
bearer	(Optional) Specifies the bearer capability of the call. Valid values are **voice** and **data**.
call_action	(Mandatory) Defines how to handle the call. Valid values: **reject**, which rejects the call. **modem**, which handles the call as a modem call. **v120**, which handles the call as a V.120 call. **sync**, which handles the call as a synchronous PPP call.
max_number_of_ calls	(Optional) Defines the maximum number of calls that this session handles simultaneously. Valid values are integers from 1–23 (in the U.S.) and 1–30 (in Europe). If this field is omitted, the default is the upper limit (23 or 30).
acp_log	(Optional) Specifies whether or not the RA 6300 forwards a call's SETUP information elements and status to the ACP log file. Valid values are **yes** and **no** (the default). Status is logged as *call accept, call reject,* or *call disc* (disconnect).

(continued on next page)

Table A-6. SPB Field Definitions (continued)

Field	Definition
rate56k	(Optional) If set to **yes**, specifies a data rate of 56 kilobytes for the B channel, even if the bearer information in the incoming ISDN SETUP message indicates a different rate. The default is **no**, which sets the data rate to the rate provided in the SETUP message. *Do not change this default unless you are in Europe or Australia and are having problems receiving calls from the U.S. In this situation, the phone company sometimes fails to specify the correct data rate. In all other situations, use the default setting of* **no**.
set	(Optional) Specifies a port parameter setting that will be applied to the session. The syntax is: **set** [*parameter parameter_value*] You can specify multiple *set* commands. These settings will override the values in non-volatile memory while the session is active, but they will not change the actual values in non-volatile memory. Any parameters not specified in the *set* field are determined by the actual global (non-volatile memory) settings.

Keeping the SPB format and field definitions in mind, perform the following steps to configure SPBs:

1. **Log into the load host and open the Annex configuration file.**

   The configuration file is located in the install directory on the load host (the default file is **/usr/annex/config.annex**). Use any system editor (e.g., **vi**, **textedit**) to open the file. For example, enter:

   ```
   cd /usr/annex
   vi config.annex
   ```

2. **Using the editor, locate the %pri section of the file. Following a few lines of introduction, you will see four sample SPBs. Read the explanatory text in the file to understand how these SPBs operate.**

   Note that the first and last SPBs have a comment character(#) at the beginning of each line. To enable these SPBs, remove the comment characters. The two SPBs in the middle do not start with comment characters and are therefore enabled. To disable these SPBs, add a comment character (#) at the beginning of each SPB line.

   Also note that the final SPB could contain a *called_no* field to prevent a particular user from dialing in. In this case, you would put the SPB at the beginning of the **%pri** section, since the RA 6300 searches SPBs in the order in which they appear.

   The default **%pri** section is as follows:

```
# Sample session parameter blocks (SPBs) for the Primary Rate
# ISDN (PRI)Remote Annex RA6300.

# These set configuration parameters for sessions (calls)
# based on dialed number, calling number, and call type.
# Each incoming call is compared against each SPB, in order,
# until there is a match; if no match is found, the call is
# rejected.  See the accompanying documentation for more
# information.

%pri

# NOTE: The SPB below will cause the RA-6300 to answer all
# "voice" bearer calls with a modem.

begin_session modem
bearer voice
call_action modem
set mode auto_detect
end_session

#

# NOTE: The SPBs below are suggested templates for handling
# V.120 and sync PPP calls.  To enable these SPBs, edit the
# "called_no" line in each.  Change it to include the telephone
# numbers specific to your PRI line, using different numbers
# for each service.  You must also remove the comment ("#")
# characters in front of each line.
```

(continued on next page)

```
#

# It is not always necessary to discriminate calls based on
# called number.  If all data calls will be V.120, for
# instance, and never sync PPP, then such a distinction is
# unnecessary.

#

#begin_session v120
#called_no
Replace_this_string_with_your_V.120_telephone_number
#bearer data
#call_action v120
#set mode auto_detect
#end_session

#begin_session sync
#called_no
Replace_this_string_with_your_SYNC_PPP_telephone_number
#call_action sync
#set mode ppp
#end_session

# This SPB isn't strictly necessary, but illustrates how to
# arbitrarily disconnect undesired calls.  Any call-matching
# information that can be used in a normal SPB could be used
# here, so, for instance, an SPB like this could be used to
# reject calls from a particular phone number if desired.

begin_session unmatched
call_action reject
end_session
```

3. **Modify the sample SPBs and/or add any you wish to.**
4. **Once you have created or changed all the SPBs you need, save the config.annex file and exit the editor.**
5. **Activate your changes.**

   To activate changes, issue the following **na** command:

   `command:` **`reset annex session`**

   This causes the Annex to re-read the Session Parameter Blocks from the configuration file. Existing calls are not reset. No new calls are answered while the reset is in progress.

Setting the Mode Parameter

One of the parameters you typically set in an SPB is the **mode** parameter, which dictates the protocol mode in which the session will operate. Valid modes ar **arap**, **cli**, **ipx, ppp**, **slip**, **connect**, **rlogin**, **telnet** and **auto_detect**.

A session set to **arap** mode supports the AppleTalk Remote Access Protocol (ARAP).

A session set to **cli** allows access to the RA 6300's Command Line Interface, which, in turn, provides access to multiple hosts. From the CLI, users can also issue the **ipx** or **ppp** command to switch to **ipx** or (asynchronous) **ppp** mode. By configuring a port this way, you can enforce CLI security before a user's session is placed in one of the two protocol modes.

A session defined as **ipx** supports the Internet Packet Exchange protocol (IPX over PPP) used by Novell Netware networks.

A session set to **ppp** mode supports the synchronous Point-to-Point Protocol when the **call_action** SPB field is set to **sync**. If the **call_action** is set to modem or V120 and the mode is set to **ppp**, the session supports asynchronous PPP.

A session set to **slip** mode supports the Serial Line Internet Protocol.

A session defined as **connect** allows the user to communicate with a LAT host via the **connect** command. Use this option in conjunction with the asynchronous port parameter **dedicated_arguments**.

A session defined as **telnet** communicates via the **telnet** command. Use this option in conjunction with the asynchronous port parameter **dedicated_arguments**.

For V.120 and modem calls, a session set to **auto_detect** mode automatically determines the protocol of the incoming data stream and handles the call accordingly. The protocols detected are CLI, SLIP, PPP (asynchronous), IPX, and ARAP.

Although you usually set the **mode** parameter within an SPB, you can also set it globally via **na** or **admin**. For example, if all your users are connecting to the RA 6300 via asynchronous PPP lines, you could issue the **na** command:

```
command: set port mode ppp
```

For information on setting port parameters globally, see *Global Port Parameters* on page 3-57.

Sample SPBs

The following are sample SPBs. Together, these SPBs handle all call types.

```
%pri

begin_session modem
bearer voice
call_action modem
set mode auto_detect
end_session

begin_session v120
called_no <nudmber1>
bearer data
call_action v120
set mode auto_detect
end_session

begin_session sync
called_no <number2>
call_action sync
set mode ppp
end_session
```

The first sample SPB handles all modem calls. Since no calling or called number is specified, this SPB will apply to any voice call the RA 6300 receives. Setting the mode parameter to *auto_detect* supports CLI, SLIP, PPP (asynchronous), IPX, and ARAP connections.

The second SPB handles all V.120 users. A called number must be included, since there are two types of data calls – V.120 (TA) and synchronous – and this is the only way to distinguish between them.

The third SPB handles all synchronous PPP calls. Again, a called number (different from the one you would use in the second SPB) is required to distinguish synchronous PPP calls from V.120 calls. The mode setting of PPP is mandatory, since the only synchronous calls the RA 6300 can receive are those using the PPP protocol.

CLI Sessions

A Remote Annex 6300 CLI session has the global or SPB **mode** parameter set to **cli**. When the session begins, the CLI prompt is displayed. At this point, the user has access to all permissible CLI commands. You can configure several options for a CLI session. When configuring CLI global or SPB parameters, consider the following:

- For modem initiated calls, the **data_bits**, **stop_bits**, and **parity** parameters are set by default to 8, 1, and none, respectively. If the modems used to dial in have characteristics different from these values, change the parameter settings accordingly.
- To limit the number of connections to other hosts that a user can activate simultaneously, set the **max_session_count**. Setting the value to 1 limits the user to one connection at a time. The default is three (with a maximum of 16).

Do not confuse the sessions (host connections) specified by **max_session_count** with the ISDN sessions configured via SPBs. The two types of sessions are unrelated.

- Set the **allow_broadcast** parameter to **N** if you want to disable the display of administrative messages generated with the **na** command **broadcast**.
- The **user_name** and **location** parameters are used for administrative information. The CLI **who** command displays this information. Also, the **user_name** is passed in the **rlogin** command's connection request. If the user does not have an account on the host under the same user name you defined with the **user_name** parameter, the user must issue the **rlogin –l** command.
- The **term_var** parameter is a string identifying the terminal emulation type. Any value defined for this parameter is passed with both the **telnet** and **rlogin** connection requests. If you define a terminal emulation type, it must be one that is valid for the host to which the user is connecting. The RA 6300 uses this parameter internally for the **edit** command only.
- CLI activity timers provide simple security by resetting idle user connections. Limited resources, like dial-in modems, are released when not in use.

  The **inactivity_timer** specifies the amount of time in minutes that the remote user can be inactive before the RA 6300 resets the connection. When this timer expires, all of the user's CLI connections to hosts are terminated. Allowable values for this parameter are **0** to **255**. The default is **0** (displays as **off**).

  Activity can be set to input (data received from the user), output (data sent to the user), or both. Set the **input_is_activity** parameter to **Y** and/or the **output_is_activity** parameter to **Y**.
- The **cli_inactivity** timer specifies the amount of time in minutes that a CLI connection to a host can remain inactive before the RA 6300 drops the connection. Allowable values for this parameter are **0** to **255**, and **immediate**. The **immediate** setting directs the RA 6300 to drop the connection immediately after the last host connection closes. The default is **0**.

- Pressing an attention key notifies the RA 6300 that the user wants to suspend an ongoing connection to a host and return to the CLI. The RA 6300 provides three parameters for defining attention keys: **short_break** enables the **Break** key on many terminals; **long_break** enables a key that generates a long break; and **attn_string** defines an attention character, or string, that produces a break.

- Typically, parameters that display with the **show port editing** command define characters that provide CLI line editing functions. Some of these characters are passed as Telnet special characters with CLI-connected devices.

Configuration Differences

When configuring the RA 6300, please note the following:

- You cannot use the **port** command to define one or more ports, although you do use the **set port** command to set parameters globally (across PRI calls).
- You cannot set the following generic port parameters:
  - **type** (always set to dial-in)
  - **type_of_modem** (all of the internal modems are Microcom modems)
  - **speed** (always set to 115200)
  - **autobaud** (always set to N)
  - **ps_history_buffer**
  - **dedicated_address**
  - **dedicated_port**

- You cannot set the **mode** parameter to **dedicated**.
- You cannot set the following flow control parameter:
  - **input_buffer_size**
- You cannot set the following serial networking protocol parameters:
  - **phone_number**
  - **slip_load_dump_host**
  - **slip_allow_dump** (always set to N)

Chapter 4
The Port Server and Rotaries

This chapter in the *Remote Annex Administrator's Guide for UNIX* describes the Port Server and Rotaries for Remote Annexes other than the RA 6300. When using that chapter for an RA 6300, note the following:

- Like other Remote Annexes, the RA 6300 accepts **telnet** and **rlogin** connection requests from users and applications on the network. However, once connected to the RA 6300, a user must specify a rotary (which has been predefined in the Annex configuration file) or use the CLI. The RA 6300 does not support devices attached to ports, so users cannot attach to a port to configure a device.
- A destination phone number is mandatory for each RA 6300 rotary. Once a user picks a rotary, the RA 6300 automatically dials that phone number. This is not the case with other Remote Annexes, for which phone numbers in rotaries are optional. On non-RA 6300 Remote Annexes, the user can dial out using the standard **atd** modem command set.
- An RA6300 has only two physical ports, the Ethernet interface (**en0**) and the PRI interface. Consequently, a rotary is not a set of physical ports, as it is on other Remote Annexes. Instead, an RA 6300 rotary is a group of virtual ports of the same type. You specify the port type as the *ports* argument in the rotary definition, which has the following syntax:

  *rotary_name***:** [*keyword*] *ports@location* [**;** *ports@location*] ...

  Table A-7 gives the valid values for *ports* and *location*.

Table A-7. Valid values for *ports* Arguments in RA 6300 rotaries.

Argument	Description
ports	One of the following values: **asy**, for asynchronous modem calls. **ta**, for V120 calls operating at 64 kilobytes. **ta_56**, for V120 calls operating at 56 kilobytes. **ta_64**, for V120 calls operating at 64 kilobytes (the same as **ta**). **ta_voice,** for V120 data-over-voice calls. This is referred to as Dial Out Speech Bearer Service.

The *location* argument and *keywords* are the same as those defined in the *Remote Annex Administrator's Guide for UNIX*, Chapter A-14. The only difference is that, as mentioned, the *phone* = keyword is mandatory for RA 6300s and optional for other Remote Annexes.

- The following is an example of two simple rotary definitions in the Annex configuration file for an RA 6300:

```
%rotary
modems: phone=5551212 asy@123.456.789.1
ta_service: phone=5557777 ta@123.456.789.1
```

The first entry defines a rotary named *modems* that handles asynchronous modem calls on the 32-modem RA 6300 at Internet address 123.456.789.1.

The second entry defines a rotary to handle V.120 calls on the same RA 6300.

The following example shows a user accessing this RA 6300 via a **telnet** command and choosing the *modems* rotary:

```
telnet 123.456.789.1
Trying 123.456.789.1...
Connected to 123.456.789.1.
Escape character is ‘^]’.
<cr>
Rotaries Defined:
     modems:              asy1-32
     ta_service:          ta1-32
     cli                  -


Enter Annex port name or number: modems
Attached to port asy1
```

If the RA 6300 in the previous example had 24 internal modems instead of 32, the *modems* rotary would be displayed as *asy1-24*; if there were only four internal modems, *asy1–4* would be displayed. Rotaries for *ta* (V.120) calls always display as *ta1–32*, since there are 32 virtual *ta* ports.

- On an RA 6300, **telnet** commands can include TCP port numbers as described in Chapter 4 of the *Remote Annex Administrator’s Guide for UNIX*, with one restriction. To **telnet** to an RA 6300 directly (without being prompted for a rotary or the CLI), you issue the **telnet** command as follows:

  ```
  telnet 123.456.789.1 5000
  ```

  This takes you directly to the CLI on the RA 6300. You cannot specify any other TCP port numbers in the 5000 range. This is unlike other Remote Annexes, on which specifying the TCP port number 5006, for example, would connect you to port 6.

Chapter 5 Modems

This chapter discusses modem support for the Remote Annex 6300 (RA 6300) and covers the following topics:

- Modem Communication Sessions
- Analog Modem Support
- Modem Administration

Modem Communication Sessions

Modem Assignments

Modems are assigned dynamically to incoming calls in rotation, moving sequentially through all modems. If no modem is available for an incoming call, the RA 6300 rejects the call with cause code 17 “user busy.” The user hears a busy tone.

Analog Modem Support

The RA 6300 supports 8, 16, 24, or 32 V.34 modems. Internally, the RA 6300 uses the Microcom Quad V.34 modem.

In a typical PRI/T1 application, the USA PRI/T1 will be configured with 24 modems. Since only 23 B channels are available for call traffic, the extra modem is a spare. In a typical European configuration, the RA 6300 will support 32 modems. Since only 30 B channels are available for call traffic, the extra modems are spares.

Spare Modems

Any available modem may be assigned to any call, regardless of the B channel that the call arrives on. Since all available modems are used in rotation, all modems in the RA 6300 are used. If a modem in a PRI/T1 environment fails, it is removed from the rotation and the RA 6300 continues to support a full PRI/T1 of 23 channels. If two modems fail in an E1 environment, the RA 6300 continues to support a full PRI/E1 of 30 channels.

Call Filtering

The RA 6300 uses call filtering to decide whether to handle the call as a modem call. Information contained in the ISDN D channel call setup messages determines whether the RA 6300 will accept a call; treat the call as Sync PPP, V.120, or an analog modem call; and start administrator-specified processes, for example, CLI, PPP, SLIP, or ARA. The following elements determine whether or not the RA 6300 accepts a call and treats it as a modem call:

- **Calling Party Number**. This element identifies the originating telephone number. The number may not be present due to user specification or telephone network limitations. Therefore, a determination based on this element may not always be possible.

- **Called Party Number**. This element identifies the number dialed by the user. The PRI line may support multiple phone numbers. Users of analog modems call a different number from users of V.120 and Sync PPP. Similarly, users of V.120 call a different number from users of modems or Sync PPP.

- **Called Party Subaddress**. This element identifies the ISDN subaddress of the dialed number and is used in the same way as the Called Party Number. This element is available only for end-to-end ISDN calls, with the PRI line provisioned for subaddressing.
- **Bearer Capability**. This element indicates whether the call contains voice or data traffic, as indicated by the originating equipment. This element is always present, but it may not be accurate. If the originating equipment utilizes a Data Over Voice Bearer Service (DOVBS), it identifies the call to its local switch as a voice call and then transmits user data such as V.120 or Sync. Many areas of the United States offer tariff advantages to voice traffic over data traffic. If the bearer capability is identified as data, the data rate is also identified (56KB or 64KB).

The RA 6300 does not use any of these call elements to invoke security features such as ACP or SecureID. Once the call is answered, security features are available by employing user data such as user name and password.

Modem Administration

Modem Status

You may display the status of all modems, busy-out a specific modem, and display call-associated data. You may configure modem status (busy-out or not busy-out). The status is preserved during a reboot of the RA 6300.

If you issue a request to busy-out a modem, the modem will not accept additional PRI calls. If you attempt to busy-out a modem that is currently handling a PRI call, the request will not be honored until the call terminates. If required, you can terminate the call immediately by issuing a reset modem request after issuing the busy-out request.

Set Modem Busy Procedure

To configure a modem to busy-out or not busy-out, follow the procedure below, using the indicated **admin** commands (you can also use **na**):

```
admin: modem modem_list
admin: set modem busy [yes|no]
admin: show modem
```

The first command defines a set of one or more modems, by number, e.g., 1-5 or 1,2,3,4,5.

The second command busies-out or does not busy-out the defined modem set, which means that it either removes the modems from or restores them to the pool of modems available for allocation.

The third command displays the status of the defined modem set.

After you complete this procedure, you will see a message indicating that you should reset the modem. At this time, you need not reset the modem, as the configuration will be taken care of the next time a call comes in. You can ignore this message in this situation.

Display Modem Status Procedure

To display the status of each modem on the RA 6300 via the CLI, use the following superuser CLI command:

annex# modem [-m[*number_range*]]

number_range is an individual modem number or a range of modem numbers separated by a hyphen. This command displays detailed information for the modem(s) specified.

The following possible status conditions exist:

Unallocated - The display indicates that this modem is available to handle a call.

Busied Out - The modem has been busied out.

Failed - The display indicates that boot-up diagnostics marked this modem as failed.

Allocated - This condition indicates the call is being handled by the modem.

Issued without any arguments, the **modem** command displays the type of modem in use.

Make Modem Available Procedure

To make a failed modem available, use the following CLI superuser command:

```
annex# modem -unumber_range
```

where *number_range* is an integer specifying the number assigned to an individual modem, or a range of integers separated by a hyphen.

This command changes the status of the modem, but has no effect if the modem is already available, is being used for a call, or has been busied out.

Chapter 6
Point-to-Point Protocol (PPP)

This chapter describes how to configure the Remote Annex 6300 (RA 6300) for access by remote nodes using the Point-to-Point Protocol (PPP). PPP lets you use the RA 6300 to provide access for remote nodes through the telephone network, using RA 6300 PRI B channels in combination with the following:

- Analog modems (e.g., V.34).
- ISDN BRI lines with terminal adapters using V.120 rate adaptation protocol.
- ISDN BRI lines using ISDN adapter cards or personal ISDN routers (e.g., Nautica Series CLAM).
- ISDN PRI lines using ISDN routers (e.g., Nautica Series Marlin).

The Point-to-Point Protocol

PPP is a standardized method for transmitting data from multiple protocols over asynchronous and synchronous point-to-point links. Data transmission and reception takes place only between the nodes at the ends of the link. PPP provides three functions:

- Asynchronous High-level Data Link Control (HDLC) to encapsulate the packets.
- Link Control Protocol (LCP) to establish the connection between peers.
- A family of Network Control Protocols (NCPs) to configure network interfaces.

The RA 6300 also supports Multilink PPP (MP), which is not described in this document. For information, see the *Multilink PPP Addendum to the Remote Annex Administrator’s Guide for UNIX*.

PPP features, as implemented in the RA 6300, include:

- FCS error checking.
- Agreement by each end of the connection on a mutually acceptable set of features for that connection.
- Large Maximum Receive Unit (MRU) size -- negotiations start at 1500.

PPP Configuration Overview

To configure the RA 6300 for PPP sessions:

1. **Decide how IP addressing will be handled.**
2. **Edit the config.annex file to define SPBs.**
3. **Review the default global port parameters, then reset the parameters you need for the PPP configuration.**

Step 1: Decide How to Assign IP Addressing

All IP addressing for PPP links is based on the value of the **address_origin** parameter, which determines the method that the RA 6300 uses to assign IP addresses. The addressing methods and their corresponding **address_origin** values are as follows:

- Setting the **address_origin** parameter to **dhcp**. This enables dynamic IP addressing, using the Dynamic Host Configuration Protocol (DHCP). Refer to *Dynamic Allocation of Network Addresses* on page A-274 for a complete description of dynamic addressing.

- Using the **acp_dialup** file. Setting the **address_origin** parameter to **acp** causes the IP addressing for individual users to be determined by the **acp_dialup** file. This method may also be used to enable dynamic IP addressing via DHCP. Refer to *Creating the acp_dialup File* on page A-276 for a complete description of the **acp_dialup** file.

  In this book, this method is also referred to as *dial-up addressing*.

- Using the asynchronous port parameters **local_address** and **remote_address**. Setting the **address_origin** parameter to **local** (the default) causes IP addresses to be assigned according to the values of the port parameters. Refer to *Configuration Parameters* on page C-29 for a description of these parameters' use.

  In this book, this method is also referred to as *fixed addressing*.

You can choose to configure the RA 6300 for any one of the methods, but setting **address_origin** to **dhcp** has priority over addressing using the **acp_dialup** file, which has priority over addressing using the asynchronous port parameters.

For information about how the RA 6300 operates when both dial-up and fixed addressing are enabled, see *Determining Dial-up Addresses using the acp_dialup File* on page A-278.

Setting address_origin to dhcp

Setting the **address_origin** parameter to **dhcp**, enables dynamic IP addressing. Dynamic IP addressing eliminates the need to assign remote client IP addresses manually (and the subsequent need to reconfigure and reboot) each time a host is added or moved to a new subnet. Addresses assigned in this way are used only as long as the remote client connection is active.

When DHCP is enabled, the RA 6300 acts as a DHCP client-by-proxy, requesting a remote PPP client address first from the DHCP server specified by the **pref_dhcp1_addr** parameter, then, if that server does not respond, from the DHCP server specified by the **pref_dhcp2_addr** parameter. For more information on DHCP, see *Dynamic Allocation of Network Addresses* on page A-274.

About Addressing Via the acp_dialup File

Addressing using the **acp_dialup** file offers the ability to assign IP addresses to individual users. When the **address_origin** parameter is set to **acp**, the RA 6300 uses the host-resident **acp_dialup** file to handle IP addressing. The file resides in the RA 6300 install directory. For information on making entries into the **acp_dialup** file, see *Creating the acp_dialup File* on page A-276.

Any ACP address request that comes from the RA 6300 includes the RA 6300 address and an associated user name, which are used as keys in this file. Once the keys are matched, the corresponding user addresses are returned to the caller on the RA 6300.

As mentioned earlier, you can also use the **acp_dialup** file to enable dynamic IP addressing for individual users, by setting the remote address field of the file to **dhcp**. For complete information, see *Dynamic Allocation of Network Addresses* on page A-274.

About Addressing Using Asynchronous Port Parameters

Setting the **address_origin** parameter to **local** causes IP addressing for the RA 6300 to be controlled by the values of two asynchronous port parameters, **local_address** and **remote_address**. This method of fixed IP addressing associates IP addresses with specific ports. For complete information on the use of the **local_address** and **remote_address** parameters, refer to *Determining Dial-up Addresses using the acp_dialup File* on page A-278.

.

Step 2: Edit the Annex Configuration File

Session Parameter Blocks (SPBs) are structures within the Annex configuration file. SPBs enable an RA 6300 to handle calls properly.

Before editing the file, determine the following:

- The type of calls that will be made to the RA 6300 (e.g., modem, V.120, and sync PPP).
- If more than one type of call will be made to the RA 6300, then the telephone number associated with the type of call.
- Whether you intend to use one or all of the default SPBs provided in the PRI section of the configuration file, disable one or all of the default SPBs, or write your own set of SPBs.

See *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51, for detailed information on SPBs.

The following instructions describe how to enable and disable the default SPBs that exist within the configuration file. These default SPBs will handle modem, V.120, and PPP calls made to the RA 6300.

To edit the configuration file:

1. **Open the configuration file.**

   The configuration file is located in the install directory on the UNIX load host (default filename is **/usr/spool/erpcd/bfs/config.annex**). Use any system editor (e.g., **vi**, **textedit**) to edit the file.

2. **Go to the** pri **section in the file. The section begins with the percent symbol** (%) **and the name** pri**. Do the following:**

   - Read the information that precedes each default SPB provided in the configuration file. You should be familiar with this information before you enable or disable a default SPB.
   - Remove the comment character (#) from the beginning of each line of each SPB you want to enable.
   - Enter a comment character (#) at the beginning of each line of each SPB you want to disable.
   - Following the called_number field in an SPB that has one, replace the string with the telephone number callers will use from remote nodes that will use this SPB. For example, to enable the default SPB that handles PPP calls, do the following:

```
#begin_session sync
#called_no Replace_this_string_with_your_SYNC_PPP_telephone_number
#call_action sync
#set mode ppp
#end_session
```

Remove the comment character (#) from the beginning of each line.

Replace this string with the telephone number callers using SYNC PPP will use.

3. **Save the file.**

4. **Issue a** reset annex session **command from** na **or** admin**.**

Step 3: Review and Reset Global Port Parameters

The RA 6300 ships with a set of default global port parameters already stored in non-volatile RAM. Review the defaults to determine which ones you need to change to satisfy your configuration requirements for PPP, security, etc.

The remainder of this section provides the following information:

- A list of the default settings for the Serial Networking and PPP global port parameter groups.
- Instructions for changing a global port parameter setting. Instructions for using the **set pri b** command to associate IP addresses with RA 6300 PRI B channels.

To view the entire set of default global port parameters use **na** or **admin** to issue the **show port all** command.

Default PPP-Related Global Port Parameters

Table A-8 lists the default parameters related to the PPP protocol stored in the RA 6300 nonvolatile memory when shipped. You can view these PPP-specific parameters through the **show port ppp** command issued from the **na** or **admin** utility.

Table A-8. Default PPP-related Global Port Parameters Settings

Parameter	Default Setting
local_address	0.0.0.0
metric	1
net_inactivity	off

(continued on next page)

Table A-8. Default PPP-related Global Port Parameters Settings (continued)

Parameter	Default Setting
allow_compression	N
address_origin	local
slip_ppp_security	N
do_compression	N
net_inactivity_units	minutes
ppp_mru	1500
ppp_security_protocol	none
ppp_password_remote	“<unset>”
ppp_ipx_network	00000000
ppp_ipx_node	00-00-00-00-00-00
ppp_acm	0x0
ppp_username_remote	““
ppp_sec_auto	N
ppp_ncp	all

How to Change a Global Port Parameter Setting

To change a global port parameter setting using **na**:

1. **At a terminal connected to a UNIX host, enter:**

   ```
   % na
   ```

   The following prompt displays on the screen:

   ```
   Annex network administrator Rx.x January 1997
   COMMAND:
   ```

2. **Specify the RA 6300 on which you intend to change global port parameter settings at the** COMMAND: **prompt. Specify the administrative password for host at the** password: **prompt.**

   You can specify the RA 6300 by its IP addresses or name. If you intend to change global port parameter settings on more than one RA 6300, separate their IP addresses or names using a comma (,). If prompted for a password, the password is the administrative password for the RA 6300..

   For example:

   ```
   COMMAND:annex 132.245.6.40 or
           annex 132.245.6.40,132.245.6.45
           password:
   ```

3. **Specify a new setting for the global port parameter at the** COMMAND: **prompt.**

   For example, to change the default setting of the **address_origin** parameter (**local**) to enable IP addressing through the **acp_dialup** file, enter the following:

   ```
   COMMAND:set port address_origin acp
   ```

   The new parameter setting is stored automatically in non-volatile RAM.

4. **To review your changes, issue the** show port all **command at the** COMMAND: **prompt.**

   This command displays all of the global port parameter settings. To locate the parameters you changed, press the **Return** key (only is pager is set), which allows you to scroll down through the file.

   ```
   COMMAND: show port all
   ```

5. **Enter** quit **at the** COMMAND: **prompt to exit** na**.**

   ```
   COMMAND: quit
   ```

How to Assign IP Addresses to PRI B Channels

Use the set **pri b** command (issued with the remote_address global port parameter) associates IP addresses with PRI B channels. This lets you configure fixed IP addressing for RA 6300 sessions. You can use **na** or **admin** to use this command.

Command Syntax

Use the following command syntax when creating B channel IP address assignments:

```
set pri b=<ch-range> remote_address <ip-addr> [<increment>]
```

where

<*ch-range*> is a single B channel number or the entire set of B channels specified by a range.

<*ip-addr*> is the IP address you want to assign to a single B channel or the first channel of the entire set.

<*increment*> is the value (number) by which you want to increment automatic IP address assignment to B channels in a range or list after the first assignment.

Usage Rules

You can associate IP addresses with PRI B channels the following ways:

- For a single B channel (e.g., 4).
- For the entire set of PRI B channels using one of the following choices:
  - Specifying channels as a range of two numbers separated by a dash (1-23 for T1; 1-30 for E-1).
  - Using the keyword **all.**

When assigning IP addresses to the entire set, you can specify an increment by which IP addresses are assigned in sequence based on the increment value. For example, if you specify the entire set of B channels available with a T1-based RA 6300 PRI module (23), an IP address of 132.245.66.230, and increment of 2 the following assignments results:

```
set pri b=1-23 remote_address 132.245.66.230 2
```

B channel #1 is assigned the IP address of 132.245.66.230, B channel #2 is assigned 132.245.66.232, B channel #3 is assigned 132.245.66.234,......, and B channel #23 is assigned 132.245.66.274.

When you do not specify any B channels, the command makes 23 or 30 IP address/B channel assignments based on the increment value.

When an increment is not specified, the command assumes a default increment of zero (0).

Configuration Samples

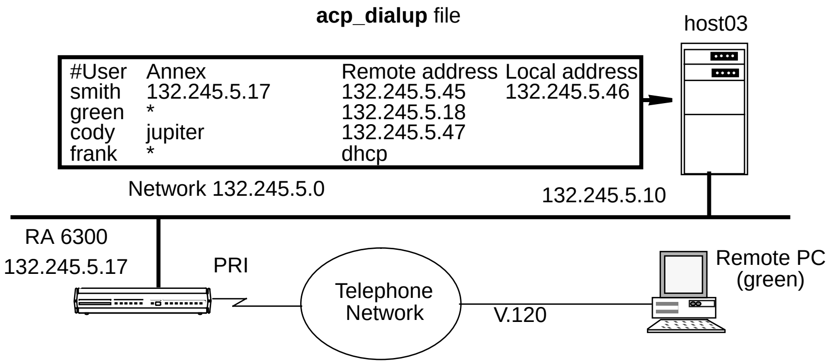
The following samples illustrate how to set global port parameters to enable PPP configurations.

Sample Configuration for Addressing Using the acp_dialup File

Figure A-5 shows a configuration in which a remote PC (i.e., user *green*) is connected to an RA 6300 through a PPP link. The PC appears to the network as directly attached device. This configuration uses addressing enabled through the **acp_dialup** file. The PC is connected through a BRI line with a V.120 terminal adapter and the PRI line to the Annex.

Based on the entries in the **acp_dialup** file, user *green* has access from all RA 6300s and other Remote Annexes since the **acp_dialup** file entry is a wildcard (*). User *green*’s **acp_dialup** file remote address is 132.245.5.18.

Following Figure A-5 are the steps to complete that implement this configuration.



```
#User   Annex          Remote address  Local address
smith   132.245.5.17   132.245.5.45    132.245.5.46
green   *              132.245.5.18
cody    jupiter        132.245.5.47
frank   *              dhcp
```

Figure A-5. Connecting a Single Host Using PPP

To enable this configuration:

1. **Edit the** acp_dialup **file. Provide user *green* with access from all RA 6300s and other Remote Annexes by specifying a wildcard (*) and a remote address of 132.245.5.18.**

2. **Use the** set pri b **command with the** remote_address **global port parameter to associate a set of IP addresses with the PRI B channels.**

   Step 2 is optional since the RA 6300 ignores the IP address/B channel assignments created using the **set pri b** command when it uses dial-up addressing. However, if the host where the ACP server resides is unreachable by the RA 6300, or there is no entry in the **acp_dialup** file for a particular user, the RA 6300 relies on the IP addresses assigned to the B channel to provide a remote address for the link.

3. **Edit the Annex configuration file to define an SPB.**

   You can use the default SPBs provided as part of the Annex configuration file or create them specifically for your requirements. For more details, see *Step 2: Edit the Annex Configuration File* on page 6-91.

4. **Reset the default global port parameters as required to the following settings:**

   - Enable CLI and/or connection security using the security parameters: **cli_security** and **connect_security**.
   - The **slip_ppp_security** parameter controls dial-in PPP access and use of ACP for PPP and protocol security such as **pap**, **chap** and **chap-pap**. If **enable_security** and **slip_ppp_security** are enabled, access to the PPP command is restricted via ACP and call access is logged in the ACP log file.
   - Set the **ppp_security_protocol** parameter to **pap**, **chap**, **chap-pap**, or, for CLI users (in some instances), **none**.
   - Set the **ppp_username_remote** and **ppp_password_remote** parameters to the values expected by the remote node (the PC in Figure A-5).

- Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
- Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

> PPP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the Remote Annex forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 *syslogs* an error message.

- Set the **address_origin** parameter to **acp** so that the RA 6300 requests the endpoint addresses, based on the user's login, from ACP.
- You can leave **ppp_mru** parameter set to its default.

Sample Configuration for Addressing Using Asynchronous Port Parameters

Figure A-6 illustrates a configuration in which a single remote PC is connected to an Annex through an asynchronous PPP session. The session is running via a BRI line with a V.120 terminal adapter and the PRI line to the Annex.

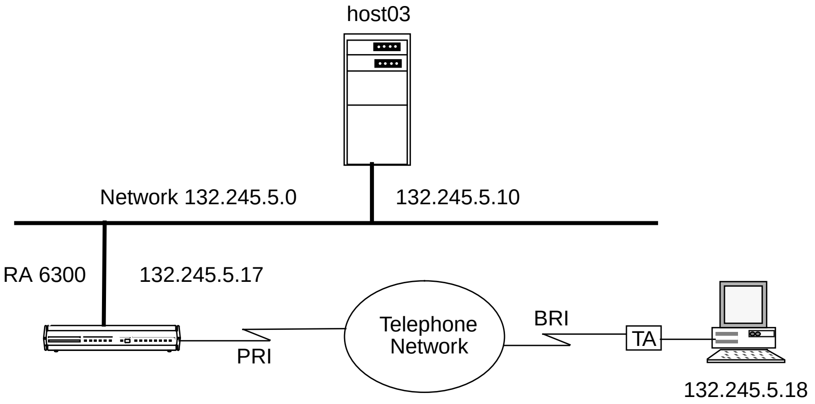


Figure A-6. Connecting a Single Host Using PPP with Fixed Addresses

To enable this configuration:

1. **Use the** set pri b **command with the** remote_address **global port parameter to associate a set of IP addresses with the PRI B channels.**

   See *How to Assign IP Addresses to PRI B Channels* on page 6-96 for instructions to perform this step.

2. **Edit the Annex configuration file to define an SPB.**

   You can use the default SPBs provided as part of the Annex configuration file or create them specifically for your requirements (for more details, see *Step 2: Edit the Annex Configuration File* on page 6-91).

3. **Reset default global port parameters as required to the following settings:**

   - Enable CLI and/or connection security using the security parameters **cli_security** and **connect_security**.

- The **slip_ppp_security** parameter controls dial-in PPP access and use of ACP for PPP and protocol security such as **pap**, **chap**, and **chap-pap**. If **enable_security** and **slip_ppp_security** are enabled, access to the **ppp** command is restricted via ACP and RA 6300 access is logged in the ACP log file.
- Set the **ppp_security_protocol** parameter to **pap**, **chap-pap**, or, for CLI users, **none**.
- Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
- Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

PPP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 *syslogs* an error message for the port.

- Set the **local_address** parameter to the RA 6300’s **en0** address.
- Set the **address_origin** parameter to **local**.
- Leave the **ppp_acm** and **ppp_mru** parameters set to their defaults.

Sample Configuration for Connecting Two Subnets

Figure A-7 illustrates two Ethernet subnets interconnected via synchronous PPP over ISDN using an ISDN router (e.g., Nautica Series Marlin) and the Remote Annex 6300. The RA 6300 has security enabled for this PPP session. Following the figure are the global port parameter settings required for this configuration.

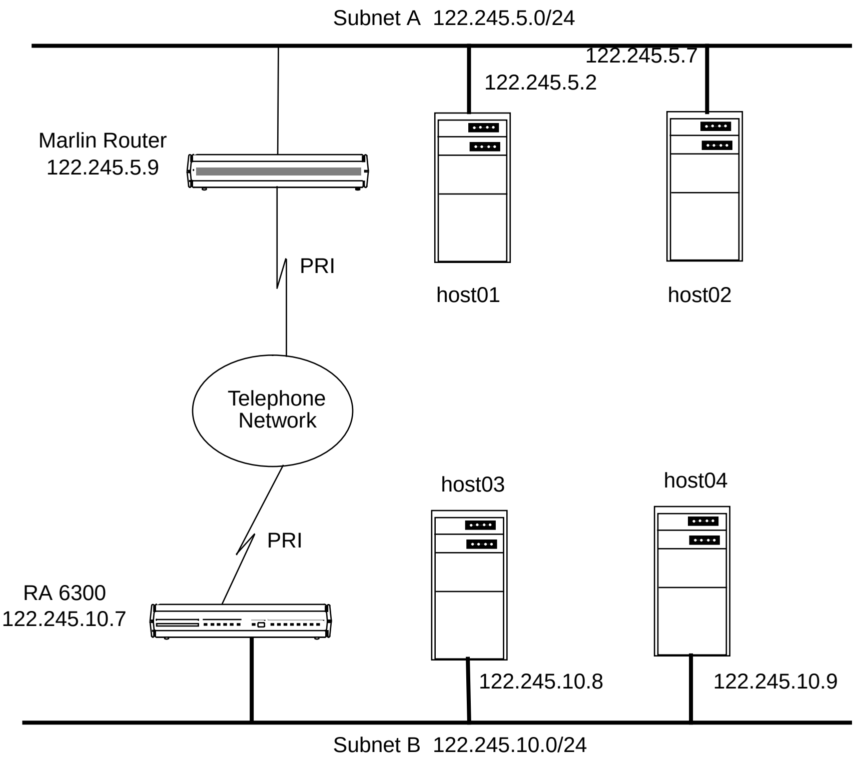


Figure A-7. PPP Link Connecting Two Ethernet Subnets

To enable this configuration:

1. **Use the** set pri b **command with the** remote_address **global port parameter to associate a set of IP addresses with the PRI B channels.**

   Specify an IP address of zero (0).

   When you specify an IP address of zero (0), the peer (the Marlin router in this sample configuration) must provide its IP address.

2. **Edit the Annex configuration file to define an SPB.**

   You can use the default SPBs provided as part of the Annex configuration file or create them specifically for your requirements. For more details, see *Step 2: Edit the Annex Configuration File* on page 6-91.

3. **Reset default global port parameters to the following settings as required:**

   - Use the defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

     PPP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 *syslogs* an error message for the port.

   - Set the **address_origin** parameter to **local**.
   - Set the **local_address** parameter to 122.245.10.7.
   - Set the **subnet_mask** parameter to 255.255.255.0.
   - Set the **metric** parameter to one.
   - Set the **ppp_username_remote** parameter to the string ““.
   - Set the **ppp_password_remote** parameter to the string “ ”.
   - Set the **slip_ppp_security** parameter to **Y.**
   - Set the **ppp_security_protocol** parameter to **pap** (password authentication protocol).

     Setting **ppp_security_protocol** parameter to **pap** assumes that the Marlin router is configured to use PAP.

   - Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.

4. **Edit the** acp_passwd **file to add the Marlin router's username and password to the file.**

   For more details, see *Creating User Password Files* on page A-247.

5. **Enter the routing information into the** gateway **section of the Annex configuration file. For example:**

```
%gateway
# PPP link to the 122.245.5.0 net
annex 122.245.10.7

     #122.245.5.9 is a gateway to the entire
     #122.245.5.0 net with a metric of 1
     route add 122.245.5.0 255.255.255.0 122.245.5.9 1

else

     #other Annexes will route to 122.245.5.0 via
     #122.245.10.7 with a metric of 2
     route add 122.245.5.0 255.255.255.0\
     122.245.10.7 2

end
```

Routing across a PPP Link (Basic Passive RIP)

Both active and passive routing are available via the Routing Information Protocol (RIP) on the RA 6300. The following sections deal with using only the most basic features of passive RIP and are intended for administrators who need minimal routing features.

Both active and passive RIP are enabled by default. To turn off active RIP, set the interface parameter **rip_advertise** to **none** for all Annex interfaces. See *rip_advertise* on page C-94.

The RA 6300 bases its routing table on the information you specify in the **gateway** section of the configuration file. As a passive gateway, the RA 6300 updates the table according to RIP information it receives from other routers, but does not broadcast routing information itself, as an active gateway would. This means that the RA 6300 with the PPP interface forwards packets addressed to the host at the remote end of the connection, but does not inform other hosts or Annexes that it has this capability. Other hosts, routers, and Annexes on the same network must be told about the route before they can use it.

Route Cache

The route cache is a list of routing entries stored by the RA 6300. When the RA 6300 boots, the route cache is created from the **annex...end** and **subnet...end** blocks in the **gateway** section of the configuration file. When **routed** starts, entries in the route cache are added to the routing table if their next hop addresses (i.e., destinations) are on a network or link directly connected to the RA 6300. The RA 6300 examines once and discards routes outside the **annex...end** blocks if the destination is not a directly connected network or is an inactive link (such as a PPP link at boot time).

Protocol Stack

Bringing up a PPP link includes three stages: link control protocol (LCP) negotiation, security, and NCP negotiation. The LCP establishes and negotiates the data link with the peer system. Next, an optional security phase authenticates the peer. Finally, NCP establishes and negotiates the network details and informs the RA 6300 that the interface (i.e., PPP link) is available.

Negotiating the LCP Options

The following subsections describe how the RA 6300 negotiates the LCP options.

Maximum Receive Unit (MRU)

The **ppp_mru** parameter sets the maximum receive unit (MRU). Acceptable values range from **64** to **1500**. The RA 6300 default is **1500**. The RA 6300 informs the peer that the **ppp_mru** parameter is its local Maximum Receive Unit (the largest packet it can receive).

Asynchronous Control Character Mask (ACCM)

The port parameter **ppp_acm** (asynchronous control mask) specifies which of the first 32 bytes (0x0 to 0x1F) can be sent as clear text and which should be protocol-escaped before being sent to the serial port.

The RA 6300 requests the **ppp_acm** parameter as its local mask. If the peer NAKs **ppp_acm**, the RA 6300 accepts the hint if it is a superset of the RA 6300's mask; otherwise, it uses the PPP default of 0xFFFFFFFF. The RA 6300 accepts any mask from the peer. Values range from **0x00000000** to **0xffffffff**. The RA 6300 default is **0x00000000**.

Setting the **ppp_acm** mask avoids sending characters that may bother the modems or devices through which the peers are connected. For example, if the modem uses CTRL-A (^A/0x01) as its attention character, it must be escaped before the RA 6300 sends it.

- **ppp_acm** for ASCII NUL (decimal 0) is 2 to the power of 0 = 0x00000001
- **ppp_acm** for ASCII SOH (decimal 1) is 2 to the power of 1 = 0x00000002
- **ppp_acm** for ASCII DC1 (decimal 17) is 2 to the power of 17 = 0x00020000
- **ppp_acm** for ASCII DC3 (decimal 19) is 2 to the power of 19 = 0x00080000

Thus, the mask for XON/XOFF (DC1 and DC3) equals the OR function of 0x00020000 and 0x00080000, or 0x000a0000.

When the RA 6300 sends an ACCM to the host, it follows this calculation to determine the initial value requested:

- The value set for **ppp_acm** (a 32-bit integer) is read in as the ACCM.
- If **input_flow_control** is set to **start/stop**, the following two additions are made:

  If **input_start_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

  If **input_stop_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.
- If **output_flow_control** is set to **start/stop**, the following two additions are made:

  If **output_start_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

  If **output_stop_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

For example, the initial ACCM sent to the peer is 0x000A0001 if **ppp_acm** is set to 0x00000001 (i.e., the ASCII NUL character will not be sent) and the following parameters are set as indicated:

input_flow_control	**start/stop**
input_start_char	**^S**
input_stop_char	**^Q**
output_flow_control	**start/stop**
output_start_char	**f**
output_stop_char	**h**

Since the output flow control parameters are outside the range 0–31 decimal, they do not affect the ACCM.

The **na/admin** command **show port ppp_acm** still displays the **ppp_acm** setting. The CLI command **netstat –ip***nn*, where *nn* is the port number, displays the true mask (ACCM) value, i.e., the value negotiated between the two PPP processes.

Magic Numbers

The Magic Number option detects data-link anomalies, namely loopback. The RA 6300 always requests this option by sending a random 4-byte word out as its Magic Number in an LCP *Configure Request*.

Link Quality Monitoring (LQM)

The RA 6300 will not request LQM. It rejects any attempts by the remote peer for LQM and hints for the PPP default of none.

Protocol Field Compression (PFC)

PFC compresses the two-byte Asynchronous HDLC protocol field to one byte. The RA 6300 always requests and accepts PFC from the peer. If NAKed, it accepts the PPP default of off. If the peer does not request PFC, the RA 6300 hints for PFC on. If the peer NAKs this hint, the RA 6300 accepts PFC off.

Address and Control Field Compression (AACFC)

ACFC deletes non-ambiguous constant address and control fields in the Asynchronous HDLC headers. The RA 6300 always requests, and accepts, ACFC. If NAKed, it accepts the PPP default of off. If the peer requests ACFC off, the RA 6300 hints for ACFC on. If the peer NAKs this hint, the RA 6300 accepts ACFC off.

Negotiating the Network Control Protocol

The Annex supports the following NCPs: AppleTalk Control Protocol (ATCP), Internet Packet Exchange Protocol Control Protocol (IPXCP), Internet Protocol Control Program (IPCP), CCP (Compression Control Protocol for PPP links), and Multilink PPP (MP). NCP options are negotiated in the same way as LCP options. An NCP peer opens the link and the interface is available to the Annex.

For information on Multilink PPP, see the *Multilink PPP Addendum to the Remote Annex Administrator's Guide for UNIX*.

To specify one or more NCPs, set the **ppp_ncp** port parameter to any combination of **ipxcp**, **ipcp**, **atcp**, **mp**, and **ccp**. Separate multiple values with a commas. You can also specify **all** to indicate all of the protocols, which is the default.

Negotiating Data Compression

If you specify **ccp** as an NCP, the Annex automatically requests data compression for a PPP link. Three types of compression are negotiated:

- Predictor-1, a public-domain algorithm
- BSD-Compress, a freely-available portion of the BSD UNIX sources
- STAC (with Check Modes 1, 3, and 4), a licensed, standard compression scheme. Check Mode 4 is one of the compression types used by the Windows ‘95 Dial-up Networking feature.

These three compression types have higher compression ratios than that provided by V.42 bis in standard modems.

Authentication Type

The authentication type specifies the style of authentication. The RA 6300 supports two authentication protocols for PPP:

- Password Authentication Protocol (PAP).
- Challenge-Handshake Authentication Protocol (CHAP).

Both of these protocols are run over the PPP link after the LCP negotiations are complete.

The RA 6300 can require the peer to pass a security check before starting NCP. The RA 6300 negotiates for the security specified by the **ppp_security_protocol** parameter. Valid arguments for this parameter are:

- **pap** (password authentication protocol [PAP]).
- **chap** (challenge-handshake authentication protocol [CHAP]).
- **chap-pap** (first negotiate for CHAP; if peer NAKs, negotiate for PAP).
- **none** (do not negotiate; the default).

The RA 6300 responds to an authentication request from a peer only if **ppp_password_remote** and **ppp_username_remote** are set for this port.

If the peer refuses a negotiation request from the RA 6300, the RA 6300 closes the link.

For a complete description of the RA 6300’s implementation of these protocols, see *Using PPP Security* on page A-293.

Negotiating the IP Address

The RA 6300 and the peer negotiate the IP address to be used on both sides of the link. Any address sent as zero requests that the peer set the address. Four parameters control the RA 6300 IP address negotiation: **address_origin**, **local_address**, **remote_address**, and **enable_security**.

If **address_origin** is set to **acp**, the RA 6300 makes an ACP **dialup_address()** call for the addresses to be used from the **acp_dialup file**.

If **address_origin** is set to **dhcp**, or if it is set to **acp** and the remote address field of the **acp_dialup** file is set to **dhcp**, the RA 6300 receives a dynamically assigned IP address.

If **address_origin** is set to **local** (its default value), or DHCP and ACP are not available, the RA 6300 defaults to using the **local_address** and **remote_address** as the addresses. The RA 6300 allows the other side of the link to select addresses only if these addresses are zero.

The RA 6300 uses two methods to negotiate the IP addresses. The preferred technique is to use the NCP type 3 *IP-Address* option. If the peer rejects this style of address negotiation, the RA 6300 falls back to using the deprecated NCP type 1 *IP-Addresses* option.

In either case, the RA 6300 requires the peer to use both the local and remote address of the RA 6300. To allow the peer to select addresses, the RA 6300 addresses must be set to zero.

If each end has a zero address and the peer cannot provide both, or the RA 6300 has a non-negotiable address, the RA 6300 and the peer will never agree upon an address, and the link will fail to come up.

Negotiating the Header Compression Type

The RA 6300 and the peer negotiate a specific protocol compression TCP/IP Header. The options are **VJ TCP/IP** and **none**. If the **allow_compression** parameter is set to **Y**, the RA 6300 always negotiates for compression on its side of the link and allows the peer to determine whether to compress data. If **allow_compression** is set to **N**, the RA 6300 never requests, and always rejects, TCP/IP header compression; the default is **N**.

BOOTP Requests

BOOTP is a bootstrap protocol that allows a diskless client to determine its Internet address, the Internet address of the server, and the name of the file to be loaded into memory.

- The RA 6300 ROMs use BOOTP to obtain boot information without requiring any manual set up on the RA 6300.
- If a diskless client sends a BOOTP request to the RA 6300 over a PPP line, the RA 6300 responds with its current local address, remote address, and boot host.

The *Remote Annex 6300 Hardware Installation Guide* discusses BOOTP in greater detail.

Using the CLI netstat –ip Command

The **netstat –ip** *device_id* command displays configuration and statistical data for serial interfaces. The *device_id* argument specifies a serial port.

Displaying Data for Ports

Ports are specified by port number alone, or the string *asy*, *ta*, or *syn* followed by the port number (with no intervening white space). Each of the following sample commands specify PPP port 1:

```
netstat -ip1
```

or

```
netstat -ipsyn1
```

Chapter 7
Serial Line Internet Protocol (SLIP)

This chapter describes how to configure the Remote Annex 6300 for Serial Line Internet Protocol (SLIP) applications. Applications include:

- Connecting a single remote node (i.e., PC) to an TCP/IP network.
- Connecting two TCP/IP networks together.

SLIP lets the RA 6300 establish sessions over its ISDN B channels. Once established, these sessions carry TCP/IP data from remote nodes and other types of RA 6300s that originated as serial traffic. SLIP sessions operate over RA 6300 ISDN B channels in combination with the following:

- Analog modems (e.g., V.34).
- ISDN BRI lines with terminal adapters using V.120 rate adaptation protocol.

The RA 6300 implementation of SLIP is compatible with the 4.3BSD implementation.

SLIP and Compressed SLIP

A SLIP link is a point-to-point connection between two hosts or devices that enables the transmission of TCP/IP packets over a serial line. Data transmission and reception is possible only between the nodes at the ends of the link.

For example, using a SLIP link, you can connect a remote PC to a network without requiring special interface hardware.

The Compressed Serial Line Internet Protocol (CSLIP) improves bandwidth by compressing the TCP/IP headers from 40 bytes to as few as three bytes when running over a SLIP link. Compression creates smaller packets, and therefore faster throughput.

You can choose either a configuration that uses compressed SLIP always, or one that uses compressed SLIP only when the remote end sends compressed SLIP packets. The RA 6300's implementation of CSLIP offers four options:

- Do compressed SLIP.
- Allow compressed SLIP.
- Discard ICMP requests over the SLIP link.
- Give interactive traffic priority over other traffic.

SLIP Configuration Overview

To configure the RA 6300 for SLIP sessions, follow these steps.

1. **Decide how IP addressing will be handled. Addresses can be handled by one of the following methods:**
   - Dial-up addressing (i.e., dynamic addressing).
   - Fixed addressing (i.e., static addressing).
2. **Edit the configuration file to define SPBs.**
3. **Review the default global port parameters, then reset the parameters you require to implement your SLIP configuration.**

Step 1: Decide How to Handle IP Addressing

The RA 6300 handles IP addresses using one of the following methods:

- Dial-up addressing
- Fixed addressing

You can choose to configure the RA 6300 for both methods, but dial-up addressing has priority over fixed addressing. For information about how the RA 6300 operates when both dial-up and fixed addressing are enabled, see *Determining Dial-up Addresses using the acp_dialup File* on page A-278.

About Dial-up Addressing

Dial-up addressing is controlled through the global port parameter **address_origin**. When this is set to **acp**, the RA 6300 uses the host-resident **acp_dialup** file to handle IP addressing. The file resides in the install directory. For more details on using the **acp_dialup** file, see *Dynamic Allocation of Network Addresses* on page A-274.

Any ACP dial-up address request that comes from the RA 6300 includes the RA 6300 address and an associated user name, which are used as keys in this file. Once the keys are matched, the corresponding dial-up addresses are returned to the caller on the RA 6300. Dial-up addressing offers the ability to assign IP addresses to individual users.

About Fixed Addressing

Fixed IP addressing for the RA 6300 is controlled through the **remote_address** parameter when used with the **set pri b** command. This parameter and command associates IP addresses with PRI B channels.

When needed (e.g., when the **address_origin** parameter is set to **local**), the IP addresses set using this command/parameter combination are automatically used for calls over the B channels. Unlike dial-up addressing, fixed IP addressing associates IP addresses with B-channels, not with specific users.

Step 2: Edit the Annex Configuration File

SPBs are structures within the configuration file. SPBs enable an RA 6300 to handle calls properly.

Before editing the file, determine the following:

- The type of calls that will be made to the RA 6300 (e.g., modem, V.120, and sync PPP).
- If more than one type of call will be made to the RA 6300, then the telephone number associated with the type of call.
- Whether you intend to use one or all of the default SPBs provided in the PRI section of the configuration file, disable one or all of the default SPBs, or write your own set of SPBs.

For more details on SPBs, see *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51.

To edit the configuration file:

The following instructions describe how to enable and disable the default SPBs that exist within the configuration file. These default SPBs will handle modem, V.120, and synchronous PPP calls made to the RA 6300.

1. **Open the configuration file.**

   The configuration file is located in the install directory on the UNIX load host (default filename is /usr/annex/config.annex).
   Use any system editor (e.g., **vi**, **textedit**) to edit the file.

2. **Go to the** pri **section in the file. The section begins with the percent symbol** (%) **and the name** pri**. Do the following:**

   - Read the information that precedes each default SPB provided in the configuration file. You should be familiar with this information before you enable or disable a default SPB.
   - Remove the comment character (#) from the beginning of each line of each SPB you want to enable.
   - Enter a comment character (#) at the beginning of each line of each SPB you want to disable.
   - Following the called_number field in an SPB that has one, replace the string with the telephone number callers will use from remote nodes that will use this SPB.

     For example, to modify a default SPB that handles SLIP calls over V.120, do the following:

```
#begin_session v120
#called_no Replace_this_string_with_your_V.120/SLIP_telephone_number
#bearer data
#call_action V.120
#set mode auto_detect
#end_session
```

Remove the comment character (#) from the beginning of each line.

Change the mode from auto_detect to slip.

Replace this string with the telephone number callers using SLIP over V.120 will use.

3. **Save the file.**

4. **Issue a** reset annex session **command from** na **or** admin**.**

Step 3: Review and Reset Global Port Parameters

The RA 6300 ships with a set of default global port parameters already stored in non-volatile RAM. Review the defaults to determine which ones you need to change to satisfy your configuration requirements for SLIP, security, etc.

The remainder of this section provides the following information:

- A list of the default settings for the Serial Networking and SLIP global port parameter groups.
- Instructions for changing a global port parameter setting.
- Instructions for using the **set pri b** command to associate IP addresses with RA 6300 PRI B channels.

To view the entire set of default global port parameters use **na** or **admin** to issue the **show port all** command.

Default SLIP-Related Global Port Parameters

Table A-9 lists the default parameters related to the SLIP protocol stored in the RA 6300 nonvolatile memory when shipped. You can view these SLIP-specific parameters through the **show port slip** command issued from **na** or **admin**.

Table A-9. Default Serial Networking Parameter Settings

Parameter	Default Setting
local_address	0.0.0.0
metric	1
net_inactivity	off
allow_compression	N
address_origin	local
slip_ppp_security	N
do_compression	N
net_inactivity_units	minutes
subnet_mask	0.0.0.0
slip_no_icmp	N
slip_mtu_size	small
slip_tos	N

Changing a Global Port Parameter Setting

To change a global port parameter setting using **na**:

1. **At a terminal connected to a UNIX host, enter:**

   ```
   % na
   ```

   The following prompt displays on the screen:

   ```
   Annex network administrator Rx.x January 1, 1997
   COMMAND:
   ```

2. **Specify the RA 6300 on which you intend to change global port parameter settings at the** COMMAND: **prompt. Specify the administrative password for host at the** password: **prompt.**

   You can specify the RA 6300 by its IP addresses or name. If you intend to change global port parameter settings on more than one RA 6300, separate their IP addresses or names using a comma (,). The password is the administrative password for this host.

   The following is an example:

   ```
   COMMAND:annex 132.245.6.40 or
           annex 132.245.6.40,132.245.6.45
           password:
   ```

3. **Specify a new setting for the global port parameter at the** COMMAND: **prompt.**

   For example, to change the default setting of the **address_origin** parameter (**local**) to enable dial-up IP addressing through the **acp_dialup** file, enter the following:

   ```
   COMMAND:set port address_origin acp
   ```

   The new parameter setting is automatically stored in nonvolatile RAM.

4. **To review your changes, issue the** show port all **command at the** COMMAND: **prompt.**

   This command displays all of the global port parameter settings. To locate the parameters you changed, press the return key which allows you to scroll down through the file.

   ```
   COMMAND: show port all
   ```

5. **Enter** quit **at the** COMMAND: **prompt to exit** na**.**

   ```
   COMMAND: quit
   ```

Assigning IP Addresses to PRI B Channels

Use the set **pri b** command (issued with the remote_address global port parameter) associates IP addresses with PRI B channels. This lets you configure fixed IP addressing for RA 6300 sessions. You can use **na** or **admin** to use this command.

Command Syntax

Use the following command syntax when creating B channel IP address assignments:

```
set pri b=<ch-range> remote_address <ip-addr> [<increment>]
```

where

<*ch-range*> is a single B channel number or the entire set of B channels specified by a range.

<*ip-addr*> is the IP address you want to assign to a single B channel or the first channel of the entire set.

<*increment*> is the value (number) by which you want to increment automatic IP address assignment to B channels in a range or list after the first assignment.

Usage Rules

You can associate IP addresses with PRI B channels:

- For a single B channel (e.g., 4).
- For the entire set of PRI B channels by:
  - Specifying channels as a range of two numbers separated by a dash (1-23 for T1; 1-30 for E-1).
  - Using the keyword **all**.

When assigning IP addresses to the entire set, you can specify an increment by which IP addresses are assigned in sequence based on the increment value. For example, if you specify the entire set of B channels available with a T1-based RA 6300 PRI module (23), an IP address of 132.245.66.230, and increment of 2, the following assignments results:

```
set pri b=1-23 remote_address 132.245.66.230 2
```

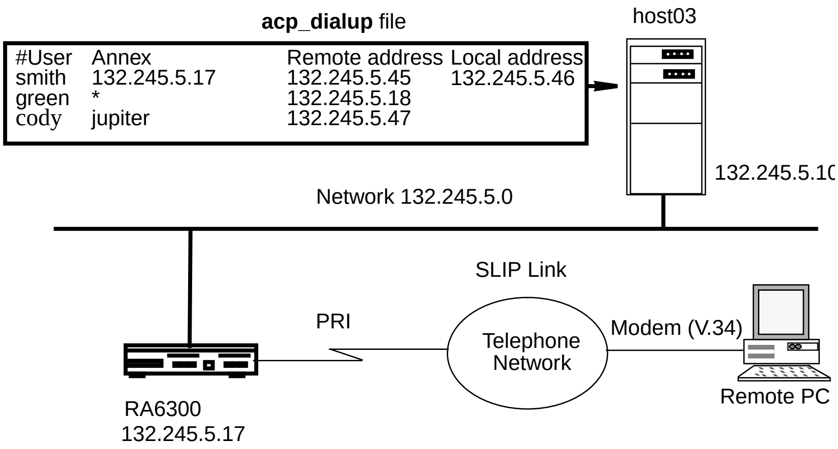
B channel #1 is assigned the IP address of 132.245.66.230, B channel #2 is assigned 132.245.66.232, B channel #3 is assigned 132.245.66.234,...., and B channel #23 is assigned 132.245.66.274.

When you do not specify any B channels, the command makes 23 or 30 IP address/B channel assignments based on the increment value.

When an increment is not specified, the command assumes a default increment of zero (0).

Sample Configuration for a Single Remote Node

In Figure A-8, a single remote PC (user *green*'s) is connected to an RA 6300 through a SLIP session. The session is running via a modem and the PRI line to the RA 6300. Once the SLIP session is established, the remote PC appears to the LAN as a directly attached device.



acp_dialup file

```
#User   Annex             Remote address Local address
smith   132.245.5.17      132.245.5.45   132.245.5.46
green   *                 132.245.5.18
cody    jupiter           132.245.5.47
```

Figure A-8. Connecting a Single Host Using SLIP

To enable this configuration:

1. **Choose whether you want to use dial-up IP addressing or fixed IP addressing.**

   If you choose dialup addressing, edit the **acp_dialup** file. Provide user *green* with access from all RA 6300s and other Remote Annexes by specifying a wildcard (*) and a specific remote address for user *green* (e.g., 132.245.5.18). Also, set the **address_origin** parameter to **acp**.

   For more details, see *Dynamic Allocation of Network Addresses* on page A-274.

If you choose fixed IP addressing, use the **set pri b** command with the **remote_addresses** global port parameter to associate a set of IP addresses with the PRI B channels. Also, set the **address_origin** parameter to **local**.

You can configure the RA 6300 for both methods, but dialup addressing has priority over fixed addressing. However, if the host where the ACP server resides becomes unreachable, or there is no entry in the **acp_dialup** file for a particular user, the RA 6300 relies on fixed addresses to provide a remote address for the link.

2. **Edit the configuration file to define an SPB.**

   You can use the default SPBs provided as part of the config.annex file or create them specifically for your requirements. For more details, see *Step 2: Edit the Annex Configuration File* on page 7-118.

3. **Review the global port parameter settings, then, if required, reset these parameter settings.**

   - Set the **subnet_mask** parameter to 255.255.255.0
   - Set the **cli_security** parameter to Y to enable this function.
   - Enabling **cli_security** implements user authentication by the host-based ACP server for all CLI connections.
   - Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

   SLIP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 generates an error message for the port.

- Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
- Set the **local_address** parameter to the RA 6300 **en0** address.

Sample Configuration for Connecting Two Subnets

Figure A-9 illustrates two Class C subnets connected through a SLIP link. The IP addresses assigned to the end-points of the SLIP link are the hosts' primary network IP addresses.

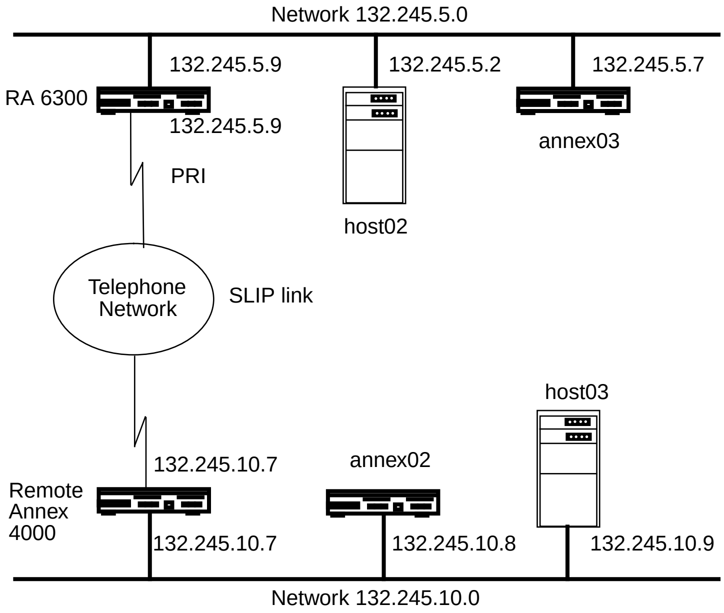


Figure A-9. SLIP Link with Two IP Addresses

To enable this configuration:

1. **Edit the configuration file to add the IP address of the Remote Annex.**

   For more details, see *Dynamic Allocation of Network Addresses* on page A-274.

2. **Edit the configuration file to define an SPB.**

   You can use the default SPBs provided as part of the config.annex file or create them specifically for your requirements. For more details, see *Step 2: Edit the Annex Configuration File* on page 7-118.

3. **Review the global port parameter settings, then, if required, reset these parameter settings.**

   - Set the **subnet_mask** parameter to 255.255.255.0.
   - Set the **cli_security** parameter to **Y** to enable this function.
   - Enabling **cli_security** implements user authentication by the host-based ACP server for all CLI connections.
   - Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

   > SLIP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 generates an error message for the port.

   - Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
   - Set the **local_address** parameter to the RA 6300 **en0** address.
   - Set the **address_origin** parameter to **acp**.

Routing Across a SLIP Link (Basic Passive RIP)

Both active and passive routing are available via the Routing Information Protocol (RIP) on the RA 6300. The following sections deal with using only the most basic features of passive RIP and are intended for administrators who need minimal routing features. For complete information on both passive and active RIP, see *Routing Information Protocol (RIP)* on page A-139.

Both active and passive RIP are enabled by default. To turn off active RIP, set the interface parameter **rip_advertise** to **none** for all Annex interfaces.

The RA 6300 bases its routing table on the information you specify in the **gateway** section of the configuration file. As a passive gateway, the RA 6300 then updates the table according to information it receives from other routers but does not broadcast routing information itself. This means that an RA 6300 with a SLIP interface forwards packets addressed to the host at the remote end of the connection, but does not inform other hosts, routers, or RA 6300s that it has this capability. Other hosts and routers on the same network must be told about the route before they can use it.

To guarantee that a route in the **gateway** section of the **config.annex** file (or in the gateway entry in **/etc/gateways**) uses a particular SLIP interface, the next hop in the route must match the remote address of the SLIP link.

Routing Between Two Networks

To make other hosts aware of a route over a SLIP link, use active routing in which a host running **routed** advertises a route for the RA 6300. Create an entry in a host's **/etc/gateways** file. Using the example in Figure A-9 on page A-127, *host03*, whose Internet address is 132.245.10.9, has the following **/etc/gateways** file entry:

```
host 132.245.5.9 gateway 132.245.10.7 metric 1 passive
```

This entry advertises a route for the host with the Internet address 132.245.5.9 through the RA 6300 at 132.245.10.7. A host running **gated** can accomplish the same thing.

Having a host advertise a route results in an *extra-hop* situation. Hosts must direct their traffic destined for host 132.245.5.9 to host 132.245.10.9, which then routes the traffic to the RA 6300 at 132.245.10.7. To avoid this extra hop, the host at 132.245.10.9 needs to send out an ICMP redirect message.

To make RA 6300s aware of a route using a SLIP link, create a **gateway** entry in the configuration file. Using Figure A-9 on page A-127, the entries for the RA 6300s on network 132.245.10.0 are:

```
annex 132.245.10.7

     route add 132.245.5.0 255.255.255.0 132.245.99.2 1

else

     route add 132.245.5.0 255.255.255.0 132.245.10.7 2
     route add 132.245.99.2 255.255.255.0 132.245.10.7 1

end
```

These entries inform *annex01* that *host01* is a gateway to network 132.245.5.0 (with a metric of 1) and inform other RA 6300s on network 132.245.10.0 that *annex01* is a gateway to either *host01*(with a metric of 1) or network 132.245.5.0 (with a metric of 2).

Route Cache

The route cache is a list of routing entries stored by the RA 6300. When the RA 6300 boots, the route cache is created from the **annex...end** and **subnet...end** blocks in the gateway section of the configuration file. When **routed** starts, entries in the route cache are added to the routing table if their next hops are on a network directly connected to the RA 6300. The RA 6300 examines once and discards routes outside the **annex...end** blocks if the destination is not a directly connected network or is an inactive link (such as a SLIP link at boot time).

Extending a Single Host onto the Network

The RA 6300 can use Proxy-ARP to attach a single host and remote RA 6300s onto the network transparently. Using Proxy-ARP, the RA 6300 answers ARP requests for the destination address of a SLIP link with its own hardware address. The following is an example of the type of ARP entry that would appear on the RA 6300 for the SLIP interface in Figure A-8:

```
bunky (132.245.5.18) at 00-80-2d-00-26-cd permanent published
```

Typically, a Proxy-ARP is used when the RA 6300's SLIP link is to a single device, i.e., both the device and the RA 6300 use the same Internet network address. No other routing information is required with this configuration.

The destination address of the SLIP link must be on the same network as the RA 6300.

BOOTP Requests

BOOTP is a bootstrap protocol that allows a diskless client to determine its Internet address, the Internet address of the server, and the name of the file to be loaded into memory.

- The RA 6300 ROMs use BOOTP to obtain boot information without requiring any manual set up on the RA 6300.
- If a diskless client sends a BOOTP request to the RA 6300 over a SLIP line, the RA 6300 responds with its current local address, remote address, and boot host (the *Remote Annex 6300 Hardware Installation Guide* describes BOOTP in detail).

Chapter 8
Dial-up Networking

This chapter in the *Remote Annex Administrator's Guide for UNIX* describes configuring dynamic dialing (also known as dial-out routing) for Remote Annexes other than the RA 6300. When using that chapter for an RA 6300, note the following changes:

- On the RA 6300, the configuration of the internal modems is hard-coded; you do not configure the modems yourself – for dial-out routing or for any other purpose.
- When defining dial-out routes in the Annex configuration file, you do not specify a port number (or a range of port numbers) for the *ports* field in a dial-out entry. Instead, you specify a rotary name or a port type. Table A-10 describes the valid port types.

Table A-10. Valid Port Types for ports Field in a Dial-out Entry

Port Type	Description
asy	For asynchronous modem calls.
ta	For V120 calls operating at 64 kilobytes.
ta_56	For V120 calls operating at 56 kilobytes.
ta_64	For V120 calls operating at 64 kilobytes (the same as **ta**).
ta_voice	For V120 data-over-voice calls. This is referred to as Dial Out Speech Bearer Service.
syn	For synchronous PPP calls operating at 64 kilobytes.
syn_56	For synchronous PPP calls operating at 56 kilobytes.
syn_64	For synchronous PPP calls operating at 64 kilobytes (same as **syn**).
syn_voice	For synchronous PPP data-over-voice calls. This is referred to as Dial Out Speech Bearer Service.

- If you specify a rotary instead of a port type, the RA 6300 uses the phone number specified in the rotary definition instead of the phone number specified in the dial-out entry. (The rotary also specifies a port type – one of the first five types shown in Table A-10). For more information on rotaries, see Chapter 4.
- All parameter settings for the global port are in effect for dial-out routing configurations, unless the parameters are overridden by *set* commands in the dial-out entry. To see the asynchronous and TA global port settings, issue the superuser **admin** or **na** command **show port all**. To see the synchronous PPP settings, issue the superuser **admin** or **na** command **show port syn**.
- Parameter settings in SPBs apply to incoming calls only and have no effect on outgoing calls. However, you must define an SPB to handle the incoming call on the remote side of a dial-out route if the destination is another RA 6300.

Sample Configurations for Dial-out Routing

This section illustrates two RA 6300s, Router A and Router B, configured for dial-out routing. Figure A-10 depicts the two Annexes. The text that follows the figure shows:

- The dial-out definitions that would appear in the RA 6300 configuration files.
- The SPBs required in order for the RA6300s to handle dial-out calls from each other.

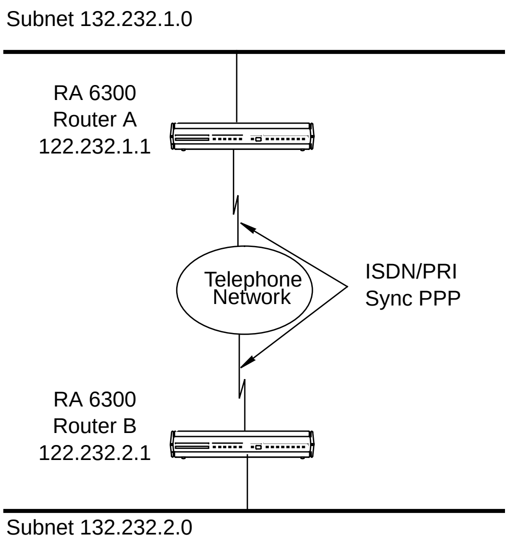


Figure A-10. RA 6300s to be Used for Dial-out

Router A's **dialout** configuration:

```
%dialout
annex 132.245.1.1

begin_route   1
mode          ppp
local         122.232.1.1
remote        122.232.2.1
set           net_inactivity 20
phone         16175551234
```

(continued on next page)

```
set          do_compression Y
set          allow_compression Y
set          net_inactivity_units minutes
set          subnet_mask 255.255.255.0
set          rip_sub_advertise Y
set          rip_sub_accept Y
set          rip_advertise all
set          rip_accept all
advertise    Y
ports        syn
set          ppp_ncp ipcp
filter       in excl proto udp src_port router netact
filter       out excl proto udp src_port router netact
end_route
end
```

In the previous example:

- The subnet mask of 255.255.255.0 applies to the remote end of the PPP link.
- The **ppp_ncp** parameter *must* be set to **ipcp** in order for the filters (defined next) to operate; filters apply to IP packets only.
- The two filters at the end of the dial-out entry prevent RIP updates (generated every 30 seconds) from being considered activity. Without these filters, RIP updates could activate a dial-out connection that might (and should) otherwise time out.

Router B's **dialout** configuration:

```
%dialout
annex 132.245.2.1
begin_route   1
mode          ppp
local         122.232.2.1
remote        122.232.1.1
set           net_inactivity 20
phone         16175554321
set           do_compression Y
set           allow_compression Y
set           net_inactivity_units minutes
set           subnet_mask 255.255.255.0
set           rip_sub_advertise Y
set           rip_sub_accept Y
set           rip_advertise all
set           rip_accept all
advertise     Y
ports         syn
set           ppp_ncp ipcp
filter        in excl proto udp src_port router netact
filter        out excl proto udp src_port router netact
end_route
end
```

The following SPB is defined in Router A’s Annex configuration file. This SPB handles Router B’s dial-out calls when they arrive at Router A.

```
%pri

begin_session sync
called_no 16175554321
call_action sync
set mode ppp
end_session
```

The framework for this SPB is included in the default RA 6300 configuration file. To activate the SPB, remove the comment character (#) from the beginning of each line that is not a comment, and specify the called number, so that the SPB resembles the example above. Then save the file and issue a **reset annex session** command from **na** or **admin**.

The following SPB is defined in Router B's Annex configuration file to handle Router A's dial-out calls when they arrive at Router B.

```
%pri

begin_session sync
called_no 16175551234
call_action sync
set mode ppp
end_session
```

For more information on SPBs, see Chapter 3.

Chapter 9
Routing Information Protocol (RIP)

This chapter in the *Remote Annex Administrator's Guide for UNIX* describes the Annex implementation of IP routing and the Routing Information Protocol (RIP) for SLIP, PPP, and Ethernet ports. When using that chapter for configuring routing on an RA 6300, note the following:

- Active RIP is enabled by default; you do not have to set the **option_key** parameter.
- Typically, you do not configure interfaces by port name and number e.g., asy5. There are only two RA 6300 interfaces, **en0** (the Ethernet interface) and **port**. Entering **port** with an **na** or **admin set** command sets the parameter value globally, that is, for all calls arriving on the PRI interface. For example, the following sample **admin** session disables all RIP advertising over the PRI interface:

  ```
  annex: su
  password:
  admin: interface port
  admin: set interface rip_advertise none
  admin: reset interface
  admin: reset default interface [Y]?
  ```

  For more details, see *interface* on page C-15.
- Specifying **port** in a **set** command in a Session Parameter Block (SPB) sets the parameter for only those calls matching the SPB's criteria. The following three SPBs show one way to handle the configuration illustrated in Figure A-11 on page A-140.

  ```
  %pri

  begin_session routing_1
  calling_no 6173335555
  called_no 6175559999
  call_action modem
  set mode ppp
  set subnet mask 255.255.255.0
  end_session
  ```

(continued on next page)

```
begin_session routing_2
calling_no 6173330291
called_no 6175559999
call_action modem
set mode ppp
set subnet mask 255.255.255.0
end_session

begin_session proxy_ARP
called_no 6175559999
call_action modem
set mode ppp
end session
```

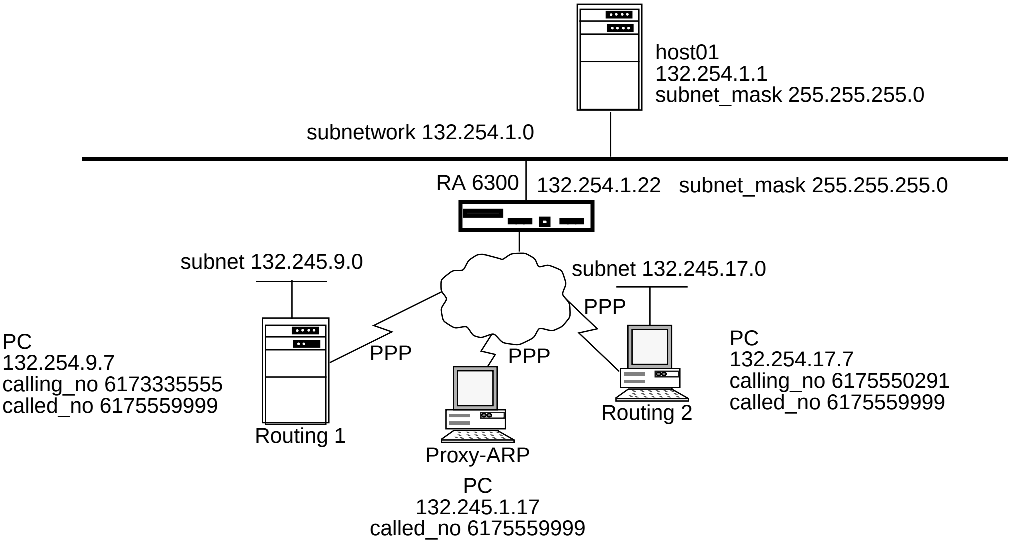


Figure A-11. Sample PPP Routing Configuration

Note the following:

- All of the SPBs define the same number for users to dial into the RA 6300 (the called number).
- The routing SPBs define different numbers from which the user can dial in (calling number). Obviously, two users cannot dial in from the same number at the same time.

  Sometimes, the telephone company does not supply the calling number. In this case, use different called numbers in the SPBs.
- The Proxy-ARP SBP should not have a subnet defined nor should there be any static routes defined for the Proxy ARP interface. Attempting to route through a Proxy-ARP interface causes packets to be routed improperly or not at all.

Other important issues to note when configuring routing for the RA 6300 are:

- **remote_address** is not a port parameter; it is a **pri b** parameter. To assign remote addresses, you issue the **set pri b** command or you define the remote address in the **acp_dialup** file. For detailed information, see *remote_address* on page C-92.
- You do not need to reboot the RA 6300 to activate a parameter change for **en0**. Instead, issue the **admin** or **na** command **reset interface en0**.

Chapter 10 Filtering

This chapter covers filtering for the Remote Annex 6300 (RA 6300) and discusses the following topics:

- Include and exclude
- Enabling filtering
- Filter numbers
- Filter lists
- The **add** subcommand
- The **list** subcommand

For the RA 6300, please note the following changes:

- Use filtering to log traffic for security or network management purposes and to determine what constitutes activity on a dial-out serial port for inactivity timeout purposes.
- Filters are applied to a particular physical interface on an RA 6300 or to all RA 6300 interfaces and can affect incoming or outgoing packets. An interface is the Ethernet port (specified as *en0* in commands) or any other determined from **acp_dialup** per user (specified by an asterisk [*] in commands).
- The **add** subcommand adds new filter(s) and enables them in both the currently running system and non-volatile memory; the RA 6300 need not be rebooted for the added filters to take effect. Please note the description of the interface, scope, netact, and protocol arguments in Table A-11. Also note the addition of **ercp** in Table A-13.
- The **list** subcommand displays the defined filters, along with their status (enabled or disabled) and assigned number (used by **enable**, **disable**, and **delete**). Please note the description of the **-e** argument in Table A-14 as well as the comments in the note below the list.

Table A-11. Arguments for the add Subcommand

Argument	Description
interface	Specifies the physical interface to which this filter applies. Valid values are **en0** (for Ethernet) or "*****".
direction	Specifying **input** applies the filter to incoming packets. Specifying **output** applies the filter to outgoing packets. Two filter definitions are required to apply a filter to both incoming and outgoing packets.
scope	Specifying **include** means the filter matches only those packets that meet all of the specified *criteria*. Specifying **exclude** means the filter matches only those packets that do not meet at least one of the specified criteria.
family	(Optional) Specifies the network level address family (protocol) to which the filter applies. Currently, the RA 6300 only supports **ip**.
criteria	Specify the conditions on which the filter is based. All criteria must be met for the filter to match the packet. Specify criteria in the form: **keyword** *value*).
actions	Specifies what a filter does when all of its *criteria* match a packet. You can specify any combination of *actions*. Possible *actions* are discard, icmp, netact, and syslog.
discard	Discards the packet. Discarding is done after any **syslog**, **icmp**, or **netact** actions are taken.
icmp	Discards the packet and sends an ICMP *destination unreachable* message.
netact	Customizes the definition of activity for a SLIP or PPP dynamic dial-out line. If one or more filters containing this action are enabled on one of these lines, only the traffic matching the filters constitutes activity. If the link is quiescent, **netact** discards the packet.
syslog	Logs the event in the system log file.

Table A-12 lists valid keywords and values for the **add** subcommand's *criteria* argument. The syntax is:

keyword value

Table A-12. Keywords for the add criteria Argument

Keyword	Value	Explanation
dst_address	{*ip_addr*[/*n*] | ***** | **-1**}	Matches the packet's destination IP address. To test only the non-host portion of the address, enter /*n* after the address, where *n* is the number of bits in the non-host portion of the subnet mask for this address. For example, **132.245.33.0/24** denotes a mask of 255.255.255.0, which matches destination addresses on network 132.245.33.0. (If you list the filter, 132.245.33.0/24 appears as the destination address.) To match all addresses, enter **–1** or "*****" instead of an address.
dst_port	{*pnum* | *sname* |***** | **-1**}	Matches the TCP or UDP destination port. Specify the port as a decimal number (*pnum*) from **1** – **65535** or as a standard service name (*sname*), such as **finger**, **ftp**, **nfs**, **rlogin**, **smtp**, **telnet**, or **tftp**. Specifying **–1** or "*****" matches all port numbers. For a list of service names and their corresponding port numbers, see Table A-13.

(continued on next page)

Table A-12. Keywords for the add criteria Argument (continued)

Keyword	Value	Explanation
src_port	{*pnum* | *sname* | ***** | **-1**}	Matches the TCP or UDP source port number. Specify the port as a decimal number (*pnum*) from **1** – **65535** or as a standard service name (*sname*), such as **finger**, **ftp**, **nfs**, **rlogin**, **smtp**, **telnet**, or **tftp**. Specifying **–1** or “*****” matches all port numbers. For a list of service names and their corresponding port numbers, see Table A-13.
src_address	{*ip_addr*[/*n*] | ***** | **-1**}	Matches the packet’s source IP address. To match only the non-host portion of the address, enter /*n* after the address, where *n* is the number of bits in the non-host portion of the subnet mask for this address. For example, **132.245.33.0/24** denotes a mask of 255.255.255.0, which matches destination addresses on network 132.245.33.0. (If you list the filter, 132.245.33.0/24 appears as the destination address.) To match all addresses, enter **–1** or “*****” instead of an address.

(continued on next page)

Table A-12. Keywords for the add criteria Argument (continued)

Keyword	Value	Explanation

address_pair	{*ip_addr1*[/*n*] | ***** | **-1**}
	{*ip_addr2*[/*n*] | ***** | **-1**}
	(Enter both addresses on the same line; separate them with a space)
		Matches packets passing in either direction between two specified IP addresses. To match only the non-host portion of an address, enter /*n* after the address, where *n* is the number of bits in the non-host portion of the subnet mask for this address. For example, **132.245.33.0/24** denotes a mask of 255.255.255.0, which matches destination addresses on network 132.245.33.0. (If you list the filter, 132.245.33.0/24 appears as the destination address.)
		To match all packets to or from a given address, enter one *ip_addr* and then specify ***** or **-1** for the other. For example, *** 132.254.33.2** and **132.254.33.2 -1** match all packets to or from 132.254.33.2.
		Restriction: if you use the address_pair keyword, you cannot use the keyword dst_address or src_address.

(continued on next page)

Table A-12. Keywords for the add criteria Argument (continued)

Keyword	Value	Explanation
port_pair	{*p1 p2*| *s1 s2* |*****|**-1**}	Matches packets passing in either direction between the two specified TCP or UDP port numbers (*p1* and *p2*) or standard service names (*s1* and *s2*), such as **finger**, **ftp**, **nfs**, **rlogin**, **smtp**, **telnet**, or **tftp**. Use a space to separate the port numbers or names. To match all packets to or from a given port number, enter one port number or service name and specify **–1** or ***** for the other. For a list of service names and their corresponding port numbers, see Table A-13. Restriction: if you use the port_pair keyword, you cannot use the dst_port or src_port keyword.
protocol	{*protonum*|*protoname*}	Matches the transport protocol in the packet. Valid protocol numbers range from **1** to **255**. Or, specify a protocol name, such as **tcp**, **udp**, or **icmp**. If no protocol is given but a port is specified (dst_port, src_port, or port_pair), the port specification applies to both TCP and UDP packets. **Warning**: A command such as the following can cause infinite loops: **filter add asy1 output include\** > **protocol icmp icmp**

Table A-13 shows the standard service names and port numbers you can supply for service name and port number values in Table A-12.

Table A-13. Standard Service Names and Port Numbers

Service Name	Port Number
domain	53
ercp	121
finger	79
ftp	21
name	42
nfs	2049
nntp	119
rlogin	221
route, routed, router	520
rtelnet	107
sftp	115
smtp, mail	25
snmp	161
telnet	23
tftp	69
time	37
who, login	513

Multiple service names shown on the same line in Table A-13 are synonyms. Using any one of them in a filter implies using the other. However, when you list the filter using the **list** subcommand, you will see only the first service name.

Using the add Subcommand

Since the NFS and TFTP protocols do not support password protection, you may want to use filtering to prevent hosts on an external network from using those protocols to access files on your internal network. To do this, pick an RA 6300 to act as a firewall between the local and external network and create filters on it to block NFS and TFTP traffic. For example, you could create the following two filters, which prevent TFTP or NFS packets from crossing RA 6300 interface *en0*:

```
annex# filter
filter: add en0 input include protocol udp port_pair\
> nfs * icmp
filter: add en0 input include protocol udp port_pair\
> tftp * icmp
```

Note the following about the preceding sample filters:

- Both filters apply only to packets arriving on RA 6300 interface *en0*. To apply a filter to another interface, specify a second filter for that interface, or specify ***** instead of *en0*, thereby blocking the protocol on all interfaces.
- Both filters match packets whose network protocol family is IP. Since the **family** argument is optional (IP is assumed), the examples omit it.
- Both filters specify **protocol** as UDP because UDP is the transport-level protocol on which NFS and TFTP operate.
- The **port_pair** argument in each filter specifies that the filter applies to any UDP packet that contains NFS or TFTP in its source or destination protocol field.

- When one of these filters matches a packet, the RA 6300 discards the packet and sends the ICMP message *destination unreachable, communication administratively prohibited* to the originator of packet. To discard the packet without sending a message, specify **discard** instead of **icmp**.

The following example creates a filter that logs the arrival of every IP packet on the Ethernet interface (*en0*). The example omits the network protocol family because it is optional; IP is assumed.

```
filter: add en0 input include src_addr * syslog
```

Logging events requires configuration (see *Logging User and RA 6300 Events* on page B-33 for more information).

The following example allows packets to and from 132.254.100.2 and 132.254.100.3 to be forwarded over interface *en0*; all other packets are discarded.

```
filter: add en0 input exclude address_pair 132.254.100.2\
> * discard
filter: add en0 input exclude address_pair 132.254.100.3\
> * discard
```

The following example allows UDP and ICMP packets to and from 132.254.100.2 and 132.2534.100.3 to be forwarded over interface *en0*; all other packets are discarded.

```
filter: add en0 input exclude address_pair 132.254.100.2 *\
> protocol icmp discard
filter: add en0 input exclude address_pair 132.254.100.3 *\
> protocol icmp discard
filter: add en0 input exclude address_pair 132.254.100.2 *\
> protocol udp discard
filter: add en0 input exclude address_pair 132.254.100.3 *\
> protocol udp discard
```

Using the list Subcommand

Table A-14 describes the arguments for the **list** subcommand.

The syntax for the **list** subcommand is:

list [**–eia**]

Table A-14. Arguments for the list Command

Argument	Description
–e	Lists the filters stored in non-volatile memory instead of the filters in the currently running system. Using **list –e** eliminates the status column from the display because the enabled/disabled status is not saved in non-volatile memory.
–i	Sorts the output by interface name, instead of sorting by filter number.
–a	Sorts the output by action, instead of sorting by filter number.

If you do not specify **–e**, **list** displays only filters that are associated with active interfaces or those that were acquired from **acp_dialup**.

Book A

Chapter 11
Internetwork Packet Exchange (IPX) Protocol

This chapter describes how to configure the RA 6300 for access by remote nodes using the Internet Packet Exchange (IPX) protocol. IPX is the network-layer communications protocol that Novell networks use to deliver data packets. The RA 6300 provides Novell dial-in connectivity and routing.

This chapter discusses the following topics:

- Novell Networks
- Standards-based IPX (IPX over PPP) Features
- Enabling IPX
- Default IPX Parameter Settings
- IPX Configuration Overview
- Configuring Standards-based IPX (IPXCP)
- Obtaining IPX Information

Novell Networks

Nodes on Novell network are *servers* or *clients*. Servers provide shared access to files, printers, and specialized peripheral devices on the network. The RA 6300 functions as a communications server, providing shared access to the network by non-Novell as well as Novell nodes. *Clients*, also referred to as *workstations*, connect to the server(s) via a network interface (Ethernet, Token Ring, or Arcnet) to access files and services. The most common client and server hardware platforms are PCs.

The Novell environment is unlike that of UNIX, in which users connected by terminals execute programs on a UNIX host. In the Novell environment, users generally execute programs on the client, not on the file server. The programs are stored on the server and retrieved for execution on the client. Moreover, a client PC, unlike a terminal, can operate as a stand-alone computer since it has its own processor, storage, operating system, and application software.

IPXCP Features

The RA 6300 implements standards-based IPX (IPX over PPP) via the IPX Control Protocol (IPXCP) described in RFC 1552. IPXCP allows a PC to dial into an RA 6300 as an endpoint node on an IPX network. The same PC can also simultaneously run IP over the connection, allowing the user to use either IP or IPX services as the need arises. (The same link can also be used for AppleTalk over PPP.)

To dial into an RA 6300 via IPXCP, a PC client can be running any operating system that supports IPXCP networking. This includes Windows ‘95, Windows NT, and DOS or Windows running FastLink II version 2.*x* or higher.

In addition to dial-in access, IPX over PPP provides asynchronous routing, allowing an IPX network to be run across a PPP LAN-to-LAN link. The RA 6300 provides this routing by default: an RA 6300 configured for IPXCP automatically sends and accepts RIP and SAP packets, provided that you set a network number for the link. However, a client can choose whether or not to receive RIPs and SAPs (if the client software allows this choice).

RA 6300 IPXCP does not support NLSP routing.

Enabling IPX

Initially, all IPX functions are disabled on a multiprotocol RA 6300. To enable standards-based IPX, set the **option_key** parameter. The **option_key** parameter activates a variety of RA 6300 features, including tn3270, AppleTalk, dialout and filtering, and IPX. Each RA 6300 requires a unique option key. You can set the **ipx_frame_type** parameter at the same time to avoid having to reboot twice. The procedure that follows explains setting both of these configuration parameters.

1. **Obtain a valid IPX value for the RA 6300** option_key **parameter.**

   Some option key values are attached to the bottom of the RA 6300. If you find the value there, enter it as described in Step 2, below.

   If the value is not there, contact your supplier to obtain a key. You will need to specify the Ethernet address of your RA 6300; it is taped to the back of the RA 6300.

   When requesting an IPX **option_key**, mention any other **option_key** features currently enabled for your RA 6300.

   To determine which options are activated (*Keyed On*), issue the CLI **stats** command with the **–o** option, as follows:

```
annex: stats –o

KEYED OPTIONS:

                   LAT: keyed off
                 Atalk: keyed off
                tn3270: keyed on
     dialout/filtering: keyed off
                   IPX: keyed off

MODULES DISABLED
     vcif
```

The *DISABLED MODULES* field displays any software modules that have been disabled via the RA 6300 **disabled_modules** parameter. (See *disabled_modules* on page C-51.) If this field displays *ipx*, then IPX is unavailable even if the IPX **option_key** parameter is set correctly.

2. **Set the** option_key **parameter to the value you obtained in Step 1.**

   In the following example, the option key is set to *RaqbDwv8e* using **admin**:

```
annex: su
Password:
annex# admin
Annex administration Remote Annex Rx.x, 72 ports
admin: set annex option_key RaqbDwv8e
```

   The **option_key** value is case-sensitive.

   The default superuser password for the RA 6300 is its IPX node.

3. **Specify the type of IPX frame that the Novell network uses to encapsulate IPX packets on the Ethernet.**

   To specify the frame type, set the RA 6300 **ipx_frame_type** parameter. Valid values are **ethernetII**, **raw802_3** (the default), **802_2**, or **802_2snap**. The following command sets the type to **802_2**:

```
admin: set annex ipx_frame_type 802_2
```

   To determine the frame type the network is using, check the AUTOEXEC.NCF file on your Novell server or use the Novell file server console command called PROTOCOL.

4. **Reboot the RA 6300 to put the parameter settings into effect:**

```
admin: q
annex# boot
```

5. **Reconnect to the RA 6300 and issue the CLI** stats –o **command to make sure that IPX is *Keyed On* and that *ipx* is not listed as a disabled module.**

6. **Issue the** stats **command with no arguments and check to make sure the network number is not displayed as *0*.**

   A network number of 0 indicates an incorrectly configured **ipx_frame_type** parameter or the absence of an IPX server on the network.

To disable IPX, set the RA 6300 **disabled_modules** parameter to **ipx** and reboot the unit. The **stats –o** command should now display *ipx* in the *DISABLED MODULES* field. (See Step 1.)

Default IPX Parameter Settings

The following table lists the default Serial Networking Protocol and IPX parameter settings.

Table A-15. Default Serial Networking Parameter Settings

Parameter	Default Setting
local_address	0.0.0.0
metric	1
net_inactivity	off
allow_compression	N
address_origin	local
slip_ppp_security	N
do_compression	N
net_inactivity_units	minutes

IPX Configuration Overview

To configure the RA 6300 for IPX sessions:

1. **Decide how IPX nodes will be handled. Nodes can be handled through either of the following methods:**
    - Dialup addressing.
    - Fixed addressing.
2. **Edit the Annex configuration file to define SPBs.**
3. **Review the default global port parameters, then reset the parameters you need for the IPX configuration.**

Step 1: Decide How to Handle Addressing

The RA 6300 handles IPX nodes using one of the following methods:

- Dial-up addressing
- Fixed addressing

You can choose to configure the RA 6300 for both methods, but dial-up addressing has priority over fixed addressing. For information about how the RA 6300 operates when both dial-up and fixed addressing are enabled, see *Determining Dial-up Addresses using the acp_dialup File* on page A-278.

Dial-up Addressing

Dial-up addressing is controlled through the global port parameter **address_origin**. When this parameter is set to **acp**, the RA 6300 uses the host-resident **acp_dialup** file to handle addressing. The file resides in the RA 6300 install directory. For information on making entries into the **acp_dialup** file, see *Creating the acp_dialup File* on page A-276.

Any ACP dial-up address request that comes from the RA 6300 includes the RA 6300 address and an associated user name, which are used as keys in this file. Once the keys are matched, the corresponding dial-up addresses are returned to the caller on the RA 6300. Dial-up addressing offers the ability to assign IPX nodes to individual users.

Fixed Addressing

Fixed addressing for the RA 6300 is controlled through the **ipx_network** parameter used with the **set pri b** command. This parameter and command associates IPX nodes with PRI B channels.

When needed (e.g., when the **address_origin** parameter is set to **local**), the IPX nodes set using this command/parameter combination are automatically used for calls over the B channels. Unlike dial-up addressing, fixed addressing associates IPX nodes with B-channels, not with specific users.

For more information, see *ipx_network* on page C-62.

Step 2: Edit the Annex Configuration File

SPBs are structures within the Annex configuration file. SPBs enable an RA 6300 to handle calls properly.

Before editing the file, determine the following:

- The type of calls that will be made to the RA 6300 (e.g., modem, V.120, and sync PPP).
- If more than one type of call will be made to the RA 6300, then the telephone number associated with the type of call.
- Whether you intend to use one or all of the default SPBs provided in the PRI section of the configuration file, disable one or all of the default SPBs, or write your own set of SPBs.

See *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51, for detailed information on SPBs.

The following instructions describe how to enable and disable the default SPBs that exist within the configuration file. These default SPBs will handle modem, V.120, and PPP calls made to the RA 6300.

To edit the configuration file:

1. **Open the Annex** configuration **file.**

   The configuration file is located in the install directory on the UNIX load host (the default file is **/usr/spool/erpcd/bfs/config.annex**). Use any system editor (e.g., **vi**, **textedit**) to edit the file.

2. **Go to the** pri **section in the file. The section begins with the percent symbol** (%) **and the name** pri**. Do the following:**

   Read the information that precedes each default SPB provided in the configuration file. You should be familiar with this information before you enable or disable a default SPB.

   Remove the comment character (#) from the beginning of each line of each SPB you want to enable.

   Enter a comment character (#) at the beginning of each line of each SPB you want to disable.

   Following the called_number field in an SPB that has one, replace the string with the telephone number callers will use from remote nodes that will use this SPB.

   For example, to enable the default SPB that handles synchronous IPX over PPP calls, do the following:

```
#begin_session sync
#called_no Replace_this_string_with_your_SYNC_PPP_telephone_number
#call_action sync
#set mode ppp
#end_session
```

Remove the comment character (#) from the beginning of each line.

Replace this string with the telephone number callers using sync IPX over PPP will use.

   To handle IPX/PPP asynchronous modem calls, create the following SPB in the **%pri** section of the configuration file:

```
begin_session ipxcp_modem
called_no Replace_this_string_with_your_IPXCP_telephone_number
call_action modem
set mode ppp
end_session
```

Replace this string with the telephone number callers using async IPX over PPP will use.

3. **Save the file.**

4. **Issue a** reset annex session **command from** na **or** admin**.**

Step 3: Review and Reset Global Port Parameters

The RA 6300 ships with a set of default global port parameters already stored in non-volatile RAM. Review the defaults to determine which ones you need to change to satisfy your configuration requirements for PPP, security, etc.

The remainder of this section provides the following information:

- A list of the default settings for the Serial Networking and PPP global port parameter groups.
- Instructions for changing a global port parameter setting.
- Instructions for using the **set pri b** command to associate IPX nodes with RA 6300 PRI B channels.

To view the entire set of default global port parameters use **na** or **admin** to issue the **show port all** command.

Default PPP-related Global Port Parameters

Table A-1 lists the default parameters related to the PPP protocol stored in the RA 6300 nonvolatile memory when shipped. You can view these PPP-specific parameters through the **show port ppp** command issued from the **na** or **admin** utility.

Table A-16. Default PPP-related Global Port Parameter Settings

Parameter	Default Setting
allow_compression	N
address_origin	local
do_compression	N
local_address	0.0.0.0
metric	1
net_inactivity	off
net_inactivity_units	minutes
ppp_acm	0x0
ppp_mru	1500
ppp_ncp	all
ppp_password_remote	“<unset>”
ppp_sec_auto	N
ppp_security_protocol	none
ppp_username_remote	““
slip_ppp_security	N

How to Change a Global Port Parameter Setting

To change a global session parameter setting using **na**:

1. **At a terminal connected to a UNIX host, enter:**

   ```
   % na
   ```

   The following prompt displays on the screen:

   ```
   Annex network administrator Rx.x January 1, 1997
   COMMAND:
   ```

2. **Specify the RA 6300 on which you intend to change global port parameter settings at the** COMMAND: **prompt. Specify the administrative password for host at the** password: **prompt.**

   You can specify the RA 6300 by its IPX nodes or name. If you intend to change global port parameter settings on more than one RA 6300, separate their IPX nodes or names using a comma (,). The password is the administrative password for this host.

   For example:

   ```
   COMMAND:annex 132.245.6.40 or
           annex 132.245.6.40,132.245.6.45
           password:
   ```

3. **Specify a new setting for the global port parameter at the** COMMAND: **prompt.**

   For example, to change the default setting of the **address_origin** parameter (local) to enable dial-up addressing through the **acp_dialup** file, enter the following:

   ```
   COMMAND:set port address_origin acp
   ```

   The new parameter setting is automatically stored in nonvolatile RAM.

4. **To review your changes, issue the** show port all **command at the** COMMAND: **prompt.**

   This command displays all of the global port parameter settings. To locate the parameters you changed, press the return key, which allows you to scroll through the file.

   ```
   COMMAND: show port all
   ```

5. **Enter** quit **at the** COMMAND: **prompt to exit** na**.**

   ```
   COMMAND: quit
   ```

Assigning IPX Networks or Nodes to B Channels

The set **pri b** command (issued with the **ipx_network** or **ipx_node** parameter) associates IPX networks or nodes with PRI B channels.

Command Syntax

Use the following command syntax when creating B channel IPX network or node assignments:

```
set pri b=ch_range {ipx_network net_no|ipx_node node_no} [increment]
```

ch-range is a single B channel number, a list of B channels separated by commas, a range of B channels separated by a hyphen, or the keyword **all**.

net_no is the IPX network (in hexadecimal) to which you want to assign a single B channel or the first channel of a set of B channels.

node_no is the IPX node (in hexadecimal, with dashes separating the octets) that you want to assign to a single B channel or to the first channel of a set of B channels.

increment is the value by which you want to increment automatic IPX node or network assignment to the B channels specified by *ch_range*. For networks, you can specify increment as an integer. For nodes, you must specify increment in the same format you would specify a node – in hexadecimal, with dashes separating the octets.

Usage Rules

You can associate IPX networks or nodes with PRI B channels in the following ways:

- For a single B channel (e.g., 4).
- For the entire set of PRI B channels by using one of the following choices:
  - Specifying channels as a range of two numbers separated by a dash (1–23 for T1; 1–30 for E–1).
  - Using the keyword **all**.

When assigning IPX nodes to the entire set, you can specify an increment by which IPX nodes are assigned in sequence based on the increment value.

The following example specifies the entire set of B channels available with a T1-based RA 6300 PRI module (23), an IPX node of 00-00-00-00-00-a0, and an increment of 2:

```
set pri b=1-23 ipx_node 00-00-00-00-00-a0 2
```

In this case, B channel #1 has an IPX node of `00-00-00-00-a0`. B channel #2 is assigned `00-00-00-00-a2`. B channel #3 is assigned `00-00-00-00-a4`, and B channel #23 is assigned `00-00-00-00-cc`.

When you do not specify any B channels, the command makes 23 or 30 IPX node or network B channel assignments based on the increment value.

When an increment is not specified, the command assumes a default increment of zero (0).

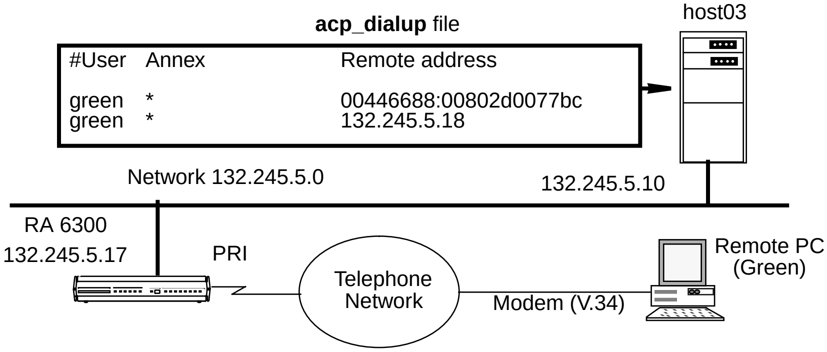
Configuration Samples

The following samples illustrate how to set global port parameters to enable PPP configurations.

Sample Configuration Using Dial-up Addresses

Figure A-12 shows a configuration in which a remote PC (i.e., user *green*) is connected to an RA 6300 through a PPP link. The PC appears to the network as directly attached device. This configuration uses dial-up addressing through the **acp_dialup** file. The PC is connected through a BRI line with a V.120 terminal adapter and the PRI line to the RA 6300.

Based on the entries in the **acp_dialup** file, user *green* has access from all RA 6300s and other Remote Annexes since the **acp_dialup** file entry is a wildcard (*). User *green*'s **acp_dialup** file remote address is 00446688:00802d0077bc. The IPX address is 00802d0077bc and the IP address is 132.245.5.18. Following Figure A-12 are the steps to complete that implement this configuration.



acp_dialup file

```
#User   Annex               Remote address

green   *                   00446688:00802d0077bc
green   *                   132.245.5.18
```

Figure A-12. Connecting a Single Host Using PPP

To enable this configuration:

1. **Edit the** acp_dialup **file. Provide user *green* with access from all RA 6300s and other Remote Annexes by specifying a wildcard** (*) **and a remote address of 132.245.5.18.**

   See *Dynamic Allocation of Network Addresses* on page A-274 for instructions.

2. **Use the** set pri b **command with the** ipx_node **parameter to associate a set of IPX nodes with the PRI B channels.**

   > Step 2 is optional since the RA 6300 ignores the IPX node/B channel assignments created using the **set pri b** command when using dial-up addressing. However, if the host where the ACP server resides is unreachable by the RA 6300, or there is no entry in the **acp_dialup** file for a particular user, the RA 6300 relies on the IPX nodes assigned to the B channel to provide a remote address for the link.

3. **Edit the configuration file to define an SPB.**

   You can use the default SPBs provided as part of the config.annex file or create them specifically for your requirements. See *Step 2: Edit the Annex Configuration File* on page 11-160, for more information.

4. **Reset default global port parameters as required to the following settings:**

   - Enable CLI and/or connection security using the security parameters: **cli_security** and **connect_security**.
   - The **slip_ppp_security** parameter controls dial-in PPP access and use of ACP for PPP and protocol security such as **pap**, **chap**, and **chap-pap**. If **enable_security** and **slip_ppp_security** are enabled, access to the PPP command is restricted via ACP and call access is logged in the ACP log file.

- Set the **ppp_security_protocol** parameter to **pap**, **chap**, **chap-pap**, or, for cli users, **none** in some instances.
- Set the **ppp_username_remote** and **ppp_password_remote** parameters to the values expected by the remote node (the PC in Figure A-12).
- Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
- Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

PPP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the Annex forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 *syslogs* an error message.

- Set the **address_origin** parameter to **acp** so that the RA 6300 requests the endpoint addresses, based on the user's login, from the **acp_dialup** file. If no remote network and node address is specified in **acp_dialup**, the RA 6300 uses the values set for the **pri b ipx_node** and **ipx_network** parameters.
- You can leave **ppp_mru** parameter set to its default.

Sample Configuration Using Fixed Addressing

Figure A-13 illustrates a configuration in which a single remote PC is connected to a RA 6300 through an asynchronous PPP session. The session is running via a BRI line with a V.120 terminal adapter and the PRI line to the Annex.

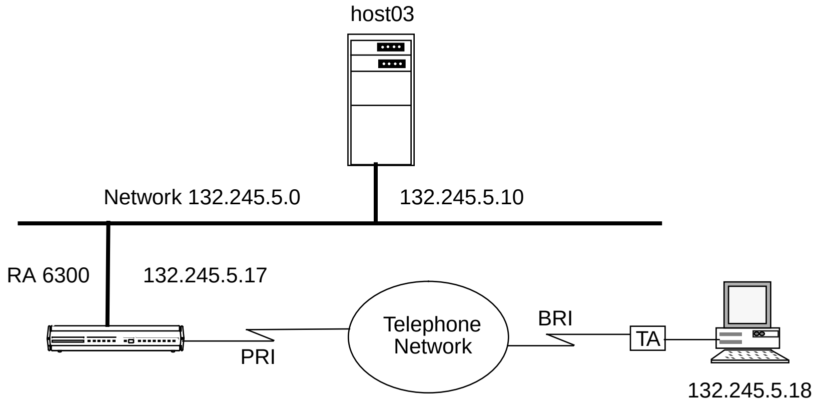


Figure A-13. Connecting a Single Host Using PPP with Fixed Addresses

To enable this configuration, follow these steps:

1. **Use the** set pri b **command with the** ipx_node **parameter to associate a set of IPX nodes with the PRI B channels.**

   See *Assigning IPX Networks or Nodes to B Channels* on page 11-165, for instructions to perform this step.

2. **Edit the configuration file to define an (SPB).**

   You can use the default SPBs provided as part of the config.annex file or create specific SPBs for your requirements. See *Step 2: Edit the Annex Configuration File* on page 11-160, for more information.

3. **Reset default global port parameters as required to the following settings:**

   - Enable CLI and/or connection security using the security parameters: **cli_security** and **connect_security**.

- The **slip_ppp_security** parameter controls dial-in PPP access and use of ACP for PPP and protocol security such as **pap**, **chap**, and **chap-pap**. If **enable_security** and **slip_ppp_security** are enabled, access to the PPP command is restricted via ACP and RA 6300 access is logged in the ACP log file.
- Set the **ppp_security_protocol** parameter to **pap**, **chap-pap**, or, for cli users, **none**.
- Set the **allow_compression** parameter to **Y** if you want the RA 6300 to accept compressed packets.
- Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

PPP is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 *syslogs* an error message for the port.

- Set the **local_address** parameter to the RA 6300's **en0** address.
- Set the **address_origin** parameter to **local**.
- Leave the **ppp_acm** and **ppp_mru** parameters set to their defaults.

Obtaining IPX Information

IPX information is available from several sources, including log messages the RA 6300 creates automatically and output that various commands display.

System Logs

The RA 6300 automatically logs **ppp** and **auto_detect** events to a 4.3BSD system log daemon (**syslogd**) on the RA 6300. For more information on syslogging, see *Using Event Logging* on page A-40 and *Logging Security Events* on page A-324.

IPXCP Interface Statistics

The **netstat –ip** command displays the IPXCP state and IPXCP options. The following is an example of IPXCP statistics.

annex: **netstat –ip 6**

```
                        LCP Status
State              Current: Open    Prior: Ack sent
Options            Local:           Remote:
MRU                1500             1500
Auth type          None             None
LQM                None             None
ACFC               On               On
ACCM               0x00000000       0x000a0000
Magic              0xbb1ee499       0x0047501b
PFC                On               On

                     NCP (IPXCP)Status
State              Current: Open    Prior: Ack sent
Options            Local:           Remote:
Network No         12345678         12345678
Node No            00802d009c30     00802d009c30
Compression        None             None
Routing Port       RIP/SAP          RIP/SAP
Router Name        LM009c30         None
```

The fields displayed for *LCP Status* are explained in *Negotiating the LCP Options* on page A-107. The fields displayed for *IPXCP Status* are explained in Table A-17.

Table A-17. Fields in (NCP) IPXCP Status Display

Field	Explanation
State	Shows the current and prior state of the IPXCP link. The states are:
Closed	The link has shut down via an administrative or peer request.
Request sent	The RA 6300 has sent a configure request and is waiting for an answer.
ACK received	The RA 6300 has received a configure ACK and is waiting for a configure request.
ACK sent	The RA 6300 received and answered a configure request.
Open	IPXCP negotiation has completed successfully.
Closing	The link is in the process of closing. The RA 6300 has sent a terminate request and is waiting for a terminate ACK.
Options	Shows the current values of the negotiated options. The *Local:* column displays the value suggested by the RA 6300. The *Remote:* column displays the value suggested by the remote client. The options are:
Network No	The 8-digit hexadecimal IPX network number of the remote client.
Node No	The 12-digit hexadecimal IPX node number of the remote client.
Compression	The kind of IPXCP header compression used. This can be *Telebit* or *None.*
Routing Proto	The routing protocol used by the RA 6300 and (optionally) the client. This can be *RIP/SAP* or *None*.
Router Name	The name by which the RA 6300 is known as an IPX router.

IPX Interfaces, Memory Buffers, Routes, and Servers

The CLI command **netstat –x** provides options for displaying information about:

- IPX in general.
- Using the **netstat –x** command itself.
- IPX network interfaces.
- The amount of memory available in the IPX buffer pools.
- IPX routes (RIPs).
- IPX servers.

The **netstat -x** syntax is:

netstat -x [**i** | **r** [*network_number*] | **s** [*server_name*] | **? | m**]
or
netstat -x [**i** | **r** [*network_number*] | **S** [*server_name*] | **? | m**]

IPX in General

Issuing the **netstat -x** command displays the number of NICs, RIPs, and Service Advertising Protocol (SAP) services on the RA 6300. *NICs* indicates the number of active IPX interfaces (including **en0**) on the RA 6300, and *RIPs* indicates the number of Novell networks the RA 6300 can reach.

The **netstat -x** command display looks like this:

```
annex: netstat –x
There are 2 NICs, 3 RIPs, and 4 SAPs
```

Using the netstat -x Command

Issuing the **netstat -x ?** command displays information about the use of **netstat x**, as follows:

```
annex: netstat -x?
Usage: netstat -x
                -xi
                -xm
                -xr [network]
                -xs [server_name]
                -xS [server_name]
```

IPX Network Interfaces

Issuing the **netstat -xi** command displays information about the RA 6300s currently in use for dial-in. The following is a sample display:
The field headings in the above display indicate the following:

```
Name  Network   Tics C0    NB   S0   Ipkts  Ierrs 0pkts  0errs Collis
en0   00001234  2    n     y    n    21592  0    201380 0    0
asy18 00003456  4    n     y    n    72     0    98     0    0
```

- *Name* is the interface name of the corresponding IPX port over which IPX dial-in or routing is currently occurring.
- *Network* is the number of the network to which interface *Name* connects.
- *Tics* indicate the amount of time associated with the cost of using interface *Name*. A tic is approximately 55 milliseconds.

- The *CO* field is not used.
- *NB* indicates whether or not this interface propagates NetBIOS information.
- *SO* indicates whether or not this interface propagates Server information only.
- *Ipkts* is the number of IPX packets received on this interface.
- *Ierrs* is the number of incoming IPX packets that contained errors.
- *Opkts* is the number of IPX packets transmitted over this interface.
- *Oerrs* is the number of outbound IPX packets that contained errors.
- *Collis* is the number of times a packet transmission was terminated due to a collision.

IPX Buffer Pools

Issuing the **netstat -xm** command displays the amount of memory available in the large and small IPX buffer pools. The RA 6300 creates these buffer pools when it boots, allotting the appropriate amount of memory for the configuration. If you change the configuration, then reboot the RA 6300 so it can allot the proper amount of buffer memory.

```
annex: netstat -xm

Large IPX Buffer Pool: Free = 0125 Total = 0125 Min =  0109

Small IPX Buffer Pool: Free = 0125 Total = 0125 Min =  0117
```

IPX Routes

Issuing the **netstat -xr** command displays the routes defined in the RA 6300’s IPX routing table. In the following example, **netstat -xr** displays five routes.

```
annex: netstat -xr

Network     Gateway          Tics      Hops      Interface

2d90ab99    0000a2816349     3         2         en0

00000042    0000a2816349     24        5         en0

00000043    0000a2816349     3         2         en0

00000044    0020af07dec4     3         2         en0

00001234    ffffffffffff     0         0         en0
```

The field headings in the above display indicate the following:

- *Network* is the number of a destination Netware network.
- *Gateway* is the number of the next hop on the path to *Network*. A gateway of ffffffffffff indicates a directly-attached network.
- *Tics* indicate the amount of time required to reach *Network* when *Gateway* is the next hop. A tic is approximately 55 milliseconds.
- *Hops* are the number of routers that must be crossed to reach *Network.*
- *Interface* is the network interface using the route.

Issuing the **netstat -xr** command followed by a network number displays the RA 6300 route for that network. The following example shows how to display the route for network 42 (you can omit the leading zeros when specifying the network number):

```
annex: netstat -xr 42

Network     Gateway           Tics      Hops      Interface

00000042    0000a2816349      24        5         en0
```

IPX Servers

Issuing the **netstat -xs** command displays server names, types, and addresses.

```
annex: netstat -xs

OSCAR      File Server   [2e80703c] 000000000001  [0451]
CTEST      Annex NAS     [00000055] 00802d01d252  [e480]
VENUS      File Server   [00006501] 000000000001  [0451]
SMTPQ      Advert Print  [00000043] 000000000001  [8060]
SNOWY      Annex NAS     [00000063] 00802d01ea57  [e480]
```

From left to right, the fields in the previous displays are as follows:

- The first field is the server name, for example, *`SUPT_TJB_INT.`* If the name is longer than 34 characters, **netstat -xs** displays only the first 34 characters.
- The second field is the server type, which can be:
  - *File Server*
  - *Job*
  - *Print*
  - *Archive*
  - *Job Queue*
  - *NAS SNA Gate(way)*

- *TimeSync VAP*
- *Dynamic SAP*
- Annex NCS
- *Annex NAS*
- *Advert(ised) Print*
- *Btrieve (5.0) VAP*
- *SQL VAP*
- *TES-NetW(are) VMS*
- *NetW(are)* Access
- Named Pipes
- *NetW(are)* UNIX
- Netware 386
- *NETW(are) manage(ment) (type 0x6601)*
- *NETW(are) manage(ment) (type 0x6a02)*
- *Unknown <type>*

In the list above, text in parentheses is provided for clarity; **netstat -xs** does not display it.

- The third field is the server’s hexadecimal address, displayed in the format [*network*] *address* [*socket*].

Issuing the **netstat -xS** command displays an additional line of information for each server. The additional line contains the RA 6300 route for the server.

```
annex: netstat -xS
HOBBESAnnex NAS      [00000012] 00802d009930 [e480]
  Gateway = [00000009] 0000a2816349  Hops = 2   IF = en0
ARAMIS NetWare 386   [0beef123] 000000000001 [8104]
  Gateway = [00000009] 0000a2816349  Hops = 3   IF = en0
ARAMIS File          [0beef123] 000000000001 [0451]
  Gateway = [00000009] 0000a2816349  Hops = 2   IF = en0
ROSA NetWare 386     [1a2a3b4c] 000000000001 [8104]
  Gateway = [00000009] 0000a2816349  Hops = 3   IF = en0
ROSA File            [1a2a3b4c] 000000000001 [0451]
  Gateway = [00000009] 0000a2816349  Hops = 2   IF = en0
```

When issued with a *server_name* argument after the **–s** or **–S** option, **netstat -sx** or **netstat -Sx** displays information for that specified server only.

Server names are typically in upper case.

IPX Frame Type and Network Number

Issued with no arguments, the CLI **stats** command displays various RA 6300 statistics, including the RA 6300 Netware network number.

The following is part of a **stats** display; IPX information is on the last line shown:

```
annex: stats

S/W: Remote Access Rx.x         Fri Jan 10 00:33:16 EST 1997
H/W: 5393/Turbo, MLB Rev 128.0  ROM Rev: 1013, PRI VERSION A
Ports: eth 1pri 32mod 32syn/ta  8MB RAM 64KB EE 67.7KB SLC1 2MB FLSH
Boot from: 132.245.33.71        Date: Mon Jan 13 16:23:24 1997 EST
Image: oper/oper.63.enet/B13.3.6Uptime: 1 day 21 hours 14 min.
Inet addr: 132.255.12.14        Subnet mask: 255.255.255.0
Ethernet addr: 00-60-2d-04-ec-bbBroadcast addr: 132.245.11.255
Primary NS: 132.255.32.7        Domain: xylogics.com
QUICC Ver: 130
IPX Frame Type: EthernetII      IPX Network Number: 11
```

(continued on next page)

If a 0 is displayed for the IPX network number, either the **ipx_frame_type** parameter was not configured properly or there is no IPX file server on the network.

IPX State

Issued with the –**o** option, the CLI **stats** command shows whether or not IPX is enabled (see *IPX Configuration Overview* on page 11-158).

IPX Connections

For all IPX ports, the CLI **who** command displays specific information about an IPX connection, including what protocol the connection is using, the user name associated with the connection, where the connection is located, when the connection was created, how long the connection has been idle, and the address from which the connection was made.

The following is an example of a **who** command display:

```
annex: who

Port  What  User  Location     When     Idle Address
v1    CLI   ---   ---          10:00am       132.245.9.4
2     PPP   ---   ---          11:00am :20  [local]
```

Statistics for All Interfaces and for 802.2

Use the CLI command **netstat –i** to display statistics for current RA 6300 interfaces and for the 802.2 data-link layer. An example follows:

```
annex: netstat –i

Name   Mtu   Network      Address    Ipkts  Ierrs Opkts  Oerrs Collis
en0    1500  132.245.66.0 worm       26563  0     15085  744   0
en0    1500  10000–20000  18062.79   1626   0     823    0     0
lo0    1536  127          127.0.0.1 0      0     0      0     0
asy2   604   18358        18062.79   0      0     0      0     0
asy16 1006   132.245.6    annex01    14770  0     7468   0     0
asy3   1500  192.9.200    zipwad     3453   0     3002   0     0

                    *** Hardware Interface Statistics ***

Ethernet Address:      00-80-2d-00-00-9b
Frames Received:       39861             Frames Transmitted:   45239
Bytes Received:        33965470          Bytes Transmitted:    29453
CRC Errors:            2                 Alignment Errors:     10
Bad Type/Length Fields:6                 Buffer Drops:         0
FIFO Drops:            1                 Interface Resets:     1
TX DMA Underruns:      241               RX DMA Overruns:      0
Carrier Sense Losses:  451               Clear to Send Losses: 0
Collisions Detected:   17526             Max Collision Retries:125
```

Chapter 12 AppleTalk

The RA 6300 implementation of AppleTalk provides dial-in connectivity in a multi-protocol network. Using the RA 6300 as a dial-in AppleTalk Remote Access (ARA) server, a remote ARA user can dial into the RA 6300 and become a directly connected ARA network user. The RA 6300 is transparent to the ARA user; it behaves like an AppleTalk end node.

AppleTalk Remote Access Protocol (ARAP)

ARAP allows Apple PowerBook and Macintosh computers to communicate with one another or with an AppleTalk network over standard telephone lines. A remote ARA user can dial into an AppleTalk network and take advantage of all the services available on the network, including:

- File transfer.
- Electronic mail.
- Database access.
- Printing.
- Mounting remote disks.

AppleTalk on the RA 6300 supports ARAP V1 and V2.

Configuring the RA 6300 for AppleTalk

Initially, all AppleTalk functions in the RA 6300 are disabled. To enable the AppleTalk functions, the network administrator must obtain and enter the correct **option_key** parameter value and then reboot the RA 6300. The way to obtain a key depends on the configuration and type of RA 6300 you purchased. Some option key values are physically attached to the bottom of the RA 6300. If the number is there, use it. If not, contact your supplier for an **option_key** value.

The **option_key** parameter enables a variety of RA 6300 features, including AppleTalk and IPX. When requesting an AppleTalk **option_key** value from your supplier, be sure to mention any of the other **option_key** features currently enabled for your RA 6300.

After the reboot, the RA 6300 automatically determines the appropriate network information, e.g., its AppleTalk node ID, etc. The AppleTalk-specific RA 6300 parameters **a_router**, **zone**, and **node_id** are hints for the RA 6300 to use at start up (*AppleTalk-specific RA 6300 Parameters* on page 12-185 describes these parameters).

The RA 6300 behaves like an AppleTalk phase II end node. At start-up, it listens for an AppleTalk router in the start-up network range and begins the process of finding its address. The RA 6300 selects as its A_Router the first router it detects broadcasting an RTMP Route Data Request. It acquires an address within the A_Router's net-range and then uses a ZIP GetNetInfo and ZIP GetZoneList to find the network's zones; otherwise, the RA 6300 obtains an available address within the start-up range.

The RA 6300 also installs a net-range route and an AppleTalk default route from the A_Router. If another router broadcasts an RTMP message, and its Ethernet address matches the address defined in the RA 6300 parameter **a_router**, the RA 6300 discards the current router information and tracks to this new router. If the RA 6300 does not hear from the current A_Router for 50 seconds it selects a new A_Router. This 50 second hold-down prevents the RA 6300 from bouncing between routers.

When ARA clients connect to an RA 6300 port, the **node_id** parameter acts as a hint to acquire an address for the client. The RA 6300 then installs a proxy *aarp* entry and the client's zone multicast address.

AppleTalk-specific Configuration Parameters

You can use either **na** or **admin** to configure RA 6300 parameters.

The AppleTalk-specific configuration parameters are divided into two groups:

- AppleTalk-specific RA 6300 parameters.
- AppleTalk-specific global port parameters.

AppleTalk-specific RA 6300 Parameters

The AppleTalk-specific RA 6300 parameters are visible only when the **option_key** parameter contains the correct key value for the RA 6300. These parameters provide some AppleTalk protocol control, limits, and identification. Table A-18 lists these parameters; the following subsections describe them in detail.

If the **option_key** parameter is invalid, the RA 6300 automatically disables AppleTalk.

Table A-18. AppleTalk-specific RA 6300 Parameters

Parameter	Default	Description
a_router	00-00-00-00-00-00	The Ethernet address of the network's A_Router.
default_zone_list	""	This zone list is sent to ARAP clients as the local back-up to ACP.
node_id	0.0	The address the RA 6300 tries to acquire at start-up.
option_key	""	Enables/disables AppleTalk.
zone	""	The AppleTalk zone for use at start-up.

Since AppleTalk uses dynamic addressing, AppleTalk addresses are acquired at boot time. The **a_router**, **zone**, and **node_id** parameters are hints for the RA 6300 to use at start-up. If another AppleTalk node is using an address defined as a hint, the RA 6300 chooses a different address. The **stats** command displays the run-time values for these parameters.

a_router

The Ethernet address of the network's A_Router. The RA 6300 uses this value as a hint at start-up. When a Routing Table Maintenance Protocol (RTMP) message arrives from this Ethernet address, the RA 6300 gleans the AppleTalk DDP address from the packet and tries to talk to the AppleTalk router. The address is a hexadecimal Ethernet address, e.g., 00-7F-12-33-44-55. The default is **00-00-00-00-00-00**.

default_zone_list

This zone list is sent to ARA clients as the local back-up to ACP. The parameter is a 100-character string with spaces separating the zones, e.g., **marketing engineering sales**. When this parameter is not set, the RA 6300 provides the network zone list. The default is a **null string** ("").

You must use the backslash (\) character to escape embedded spaces.

node_id

This is the address the RA 6300 tries to acquire at start-up. If this address is in use, the RA 6300 must acquire a new node ID. The **node_id** is an AppleTalk address in the form *net.node*. Valid *net* values are **0** to **65534**; valid *node* values are **0** to **254**. The default is **0.0**.

option_key

The **option_key** parameter enables the AppleTalk-specific RA 6300 parameters as well as the ARA protocol. AppleTalk commands, parameters, and port functions are enabled only after the correct key is set; after setting the key, the administrator *must* reboot the RA 6300. Each RA 6300 requires a unique key value. The way to obtain a key depends on the configuration and type of RA 6300 you purchased. Some option key values are physically attached to the bottom of the RA 6300. If the number is there, use it. If not, contact your supplier to obtain an **option_key** value.

The **option_key** parameter enables a variety of RA 6300 features, including tn3270, AppleTalk, and IPX. When requesting an AppleTalk **option_key** value from your supplier, be sure to mention any of the other **option_key** features currently enabled for your RA 6300.

zone

The **zone** parameter provides the AppleTalk zone for use at start-up. It is a 32-byte string variable.This is the zone in which ARA clients are located unless overridden by security. The default is a **null string** ("").

AppleTalk-specific Global Port Parameters

The **set <parameter>** command modifies the AppleTalk-specific port parameters (where <parameter> is one of the parameters described in this section). The **show port appletalk** command displays them. Table A-19 lists these parameters; the following subsections describe them in detail.

Table A-19. AppleTalk-specific Global Port Parameters

Parameter	Default	Purpose
arap_v42bis	Y	Enables/disables V.42bis compression during an ARA session.
at_guest	N	Allows ARA guest login service.
at_nodeid	0.0	The node ID given to an ARA client during connection establishment.
at_security	N	Enables/disables ACP service for this port.

arap_v42bis

The **arap_v42bis** parameter enables/disables V.42bis compression during an ARA session. A **Y** enables the parameter; an **N** disables it. The default is **Y**.

If you disable this parameter, you may want to change the Communications Control Language (CCL) script for the remote modem to improve performance. Sample CCL scripts are included in the software distribution.

at_guest

The **at_guest** parameter allows guests to log into an ARA service. When this parameter is enabled and a client requests guest access, the RA 6300 asks ACP for user name (guest) privileges. A **Y** enables guest privileges; an **N** disables guest privileges. The default is **N**.

at_nodeid

The **at_nodeid** parameter defines the node ID hint used for an ARA client during connection establishment. This parameter value is an AppleTalk address in the form *net.node*. The valid *net* values are **0** to **65534**. The valid *node* values are **0** to **254**. The default is **0.0**.

at_security

The **at_security** parameter enables/disables ACP service for this port. When both **at_security** and **enable_security** are set, the RA 6300 uses ACP to get per-user security information about the client (authentication, logging, and zone access) from the **acp_userinfo** file (see *Creating the acp_userinfo File* on page A-249). If **at_security** is not set, the RA 6300 uses only local security (**port_password** and **username** for authentication, and the **default_zone_list**). A **Y** enables this parameter; an **N** disables it. The default is **N**.

CLI AppleTalk Commands

The Command Line Interface (CLI) is the command interface for the RA 6300. At the CLI, you enter commands that connect to hosts, manage jobs (or sessions), display and modify port parameters, and display information for the RA 6300 and the network.

The CLI provides two groups of commands: user and superuser. You administer the RA 6300 using the superuser commands. Table A-20 lists the CLI commands for use with AppleTalk; the following subsections describe them (*Using the CLI Commands* on page C-111 describes all CLI commands).

Table A-20. CLI AppleTalk Commands

Command	Description
arap	Converts a CLI line into an ARA connection. When the port is reset, it reverts to its original mode. After entering the command, the RA 6300 prompts: *Annex switching line to ARAP*. The **arap** command does not apply to VCLI connections.
arp	This superuser command displays ARP cache on the RA 6300.
netstat	Displays information about AppleTalk interfaces.
ping	This superuser command generates AppleTalk Echo Packets (AEP).
stats	Displays AppleTalk information.
who	Displays a line’s type as ARA.

Command Syntax

You can shorten any CLI command or host name to the minimum number of letters that make the name unique. This is referred to as *minimum uniqueness*. If you do not want the RA 6300 to interpret a host name using minimum uniqueness, enclose the name in double quotes (""). For example, entering hosts “new” prevents ambiguities between hosts newark and new. You can enter commands and host names in all lower case, all upper case, or a combination of both. The RA 6300 performs any necessary case conversion.

arap

The **arap** command converts a CLI line into an ARA connection. Resetting the port returns the CLI to its original mode. The syntax is:

arap

The command display looks like this:

```
annex: arap
Annex switching line to ARAP.
```

arp

The **arp** command displays and, optionally, modifies the IP-to-hardware address translation table that the Address Resolution Protocol (ARP) uses. Since the RA 6300 builds the ARP table dynamically, you rarely need to modify it. Table A-21 lists the arguments for this command.

Although the **arp** command shows AppleTalk information, you cannot manipulate it. Since **arp** interprets all address as IP addresses, if you try to delete an AppleTalk address such as 1.123 using **arp –d**, the ARP table entry 1.0.0.123 is deleted.

The syntax is:

arp [**–ads**] [*host*] [*addr*] [**temp** | **pub**]

Using either the *host* or the **–a** argument, **arp** displays a host name, if known, or a **?** in place of the host name, the Internet and Ethernet addresses, and the *time to live* (TTL) field for each entry. For example:

```
annex01# arp –a

xenna (192.9.200.95) at 08-00-4C-00-2a-c0 tt1=20
2356.189 at 08-00-4e-34-22-39 tt1=16
```

Table A-21. Arguments for the arp Command

Argument	Description
host	Displays the current ARP table entry for that host.
addr	Displays the current ARP table entry for that address.
–a	Displays all entries in the table.
–d	Deletes the entry specified with *host*.
–s	Creates an entry for the host, specified using either *host* or an Internet address, at the hardware address specified using *addr*. If you do not include **temp** or **pub**, the entry is permanent and not published.
temp	The created entry is temporary and is to be deleted after 20 minutes. Temporary entries are not published.
pub	The created entry is to be published. The RA 6300 responds to requests for the host’s hardware address.

AppleTalk over ARA

AppleTalk over ARA allows Apple PowerBook and Macintosh computers to communicate with one another or with an AppleTalk network over standard telephone lines. An ARA user can dial into a remote AppleTalk network and use all the available services as if that user is physically connected to the network through EtherTalk.

AppleTalk Configuration Overview

To configure the RA 6300 for PPP sessions:

1. **Specify the option key through the** option_key **parameter using** na **or** admin**.**
2. **Edit the** config.annex **file to define SPBs for AppleTalk.**
3. **Review the default global port parameters, then reset the parameters you need for the PPP configuration.**

Step 1: Setting the option_key Parameter

To enable the AppleTalk functions, the network administrator must obtain and enter the correct **option_key** parameter value and then reboot the RA 6300. The way to obtain a key depends on the configuration and type of RA 6300 you purchased. Some option key values are physically attached to the bottom of the RA 6300. If the number is there, use it. If not, contact your supplier for an **option_key** value.

For example, if the option key is **RaqbDwv8e**, specify it with the **option_key** parameter from **na** or **admin**. From **admin**, enter:

```
set annex option_key RaqbDwv8e
```

Step 2: Edit the Annex Configuration File

Session Parameter Blocks (SPBs) are structures within the Annex configuration file. SPBs enable an RA 6300 to handle calls properly.

Before editing the file, determine:

- The type of calls that will be made to the RA 6300.
- The telephone number associated with the type of call (if there is more than one type).
- Whether you intend to use one or all of the default SPBs provided in the PRI section of the configuration file, disable one or all of the default SPBs, or write your own set of SPBs.

See *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51, for detailed information on SPBs.

The following instructions describe how to enable and disable the default SPBs that exist within the configuration file.

To edit the configuration file:

1. **Open the configuration file.**

   The configuration file is located in the install directory on the UNIX load host (default filename is **/usr/spool/erpcd/bfs/config.annex**). Use any system editor (e.g., **vi**, **textedit**) to edit the file.

2. **Go to the** pri **section in the file. The section begins with the percent symbol (%) and the name** pri**. Do the following:**

    - Read the information that precedes each default SPB provided in the configuration file. You should be familiar with this information before you enable or disable a default SPB.
    - Remove the comment character (#) from the beginning of each line of each SPB you want to enable.
    - Enter a comment character (#) at the beginning of each line of each SPB you want to disable.
    - Following the called_number field in an SPB that has one, replace the string with the telephone number callers will use from remote nodes that will use this SPB.

      For example, to modify a default SPB to handle AppleTalk calls:

```
#begin_session appletalk
#called_no Replace_this_string_with_your_appletalk_telephone_number
#call_action modem
#set mode arap
#end_session
```

Remove the comment character (#) from the beginning of each line.

Replace this string with the telephone number callers using AppleTalk will use.

3. **Save the file.**

4. **Issue a** reset annex sessions **command from** na **or** admin**.**

Step 3: Review and Reset Global Port Parameters

The RA 6300 ships with a set of default global port parameters already stored in non-volatile RAM. Review the defaults to determine which ones you need to change to satisfy your configuration requirements for PPP, security, etc.

The remainder of this section provides instructions for changing a global port parameter setting.

To view the entire set of default global port parameters use **na** or **admin** to issue the **show port all** command.

How to Change a Global Port Parameter Setting

To change a global port parameter setting using **na**:

1. **At a terminal connected to a UNIX host, enter:**

   ```
   % na
   ```

   The following prompt displays on the screen:

   ```
   Annex network administrator Rx.x January 1, 1997
   COMMAND:
   ```

2. **Specify the RA 6300 on which you intend to change global port parameter settings at the** COMMAND: **prompt. Specify the administrative password for host at the** password: **prompt.**

   You can specify the RA 6300 by its IP addresses or name. If you intend to change global port parameter settings on more than one RA 6300, separate their IP addresses or names using a comma (,). If prompted for a password, the password is the administrative password for the host on which na is running.

For example:

```
COMMAND:annex 132.245.6.40 or
        annex 132.245.6.40,132.245.6.45
        password:
```

3. **Specify a new setting for the global port parameter at the** COMMAND: **prompt.**

   For example, to change the default setting of the at_security parameter (N) to enable security enter the following:

   ```
   COMMAND:set port at_security Y
   ```

   The new parameter setting is automatically stored in nonvolatile RAM.

4. **To review your changes, issue the** show port all **command at the** COMMAND: **prompt.**

   This command displays all of the global port parameter settings. To locate the parameters you changed, press the return key which allows you to scroll down through the file.

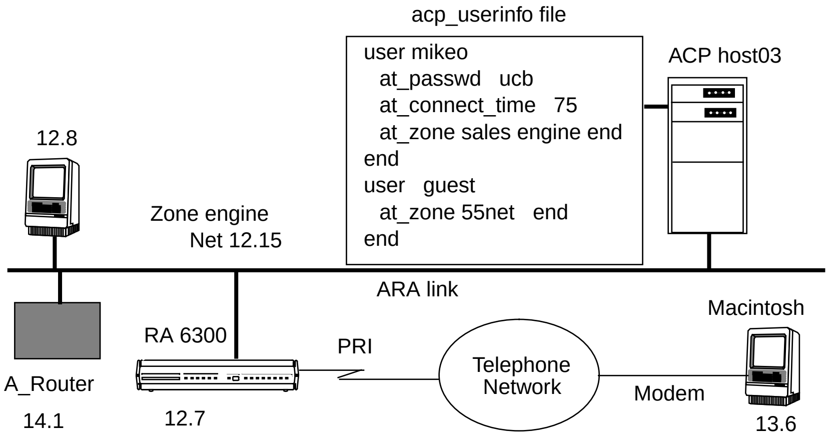
   ```
   COMMAND: show port all
   ```

5. **Enter** quit **at the** COMMAND: **prompt to exit** na**.**

   ```
   COMMAND: quit
   ```

Sample AppleTalk Configuration

Figure A-14 illustrates the following sample settings. In this configuration, a Macintosh connected to an RA 6300 through an ARA link appears to the network as an attached node.



```
user mikeo
  at_passwd   ucb
  at_connect_time   75
  at_zone sales engine end
end
user   guest
  at_zone 55net   end
end
```

Figure A-14. Connecting a Macintosh Using ARA

To enable this configuration:

1. **Make sure that you have obtained the** option key **and specified it through the** option_key **parameter.**
2. **Edit the configuration file to define an SPB.**

   You can use the default SPBs provided as part of the config.annex file or create them specifically for your requirements. For more details, see *Step 2: Edit the Annex Configuration File* on page 12-194.
3. **Reset default global port parameters as required to the following settings:**
   - Enable ARA security by setting **at_security** to **Y**. ACP and port access is logged in the ACP log file.

- Use the supplied defaults for the **data_bits** (8), **stop_bits** (1), and **parity** (none) parameters.

ARA is an 8-bit protocol. If **data_bits** is set to **7**, and **parity** is not set to **none**, the RA 6300 forces the **data_bits** setting to **8** and the **parity** setting to **none**. Otherwise, the RA 6300 generates an error message for the port.

- Set the **arap_v42bis** parameter to **Y**.
- Setting the **at_guest** parameter to **Y** enables guest access.
- The **at_nodeid** parameter defines the AppleTalk node address for the remote Apple PowerBook or Macintosh.

If you intend to use AppleTalk over PPP, see *Point-to-Point Protocol (PPP)* on page A-87.

ARA Security

The RA 6300 provides comprehensive security features that assist you in securing your RA 6300s and the network from unauthorized access. Using these features, you can select between host-based security (where at least one host on the network is functioning as a security server) and local password protection (where the passwords are stored on the RA 6300). Optionally, you can use local password protection as a back-up to host-based security.

If you are using host-based security, you must define the user name and password in the **acp_userinfo** file (see *Creating the acp_userinfo File* on page A-249).

The RA 6300 provides protection through the use of an administrative password that controls access to the superuser CLI commands. This password can also protect access to an RA 6300 through **na**. The security system provides audit trails that monitor users and their activities. The RA 6300 also provides the source code for the Access Control Protocol (ACP) security system, and the flexibility to integrate RA 6300 security with existing security for a network-wide system.

The following subsections briefly describe RA 6300 security as it relates to ARA. For a detailed description of ACP, host-based security, and the **acp_userinfo** file, see *Using RA 6300 Security* on page A-211.

Security Features

The RA 6300 implementation of ARA provides three areas of security:

- ARA security.
- Zone security.
- Logging.

ARA Security

The basic ARA security features are:

- **username and password authentication**

  The RA 6300 authenticates the client using Apple’s DES encryption algorithm. To define a user name and password for a registered (as opposed to guest) user, see *Creating the acp_userinfo File* on page A-249.

- **guest access**

  The RA 6300 allows anonymous access to the network. Restrictions can be applied to *guests* by setting up an ACP *guest* profile with limitations. For more details, see *at_zone* on page A-263.

- **connection timer**

  The connection timer is stored in the **acp_userinfo** file. For more details, see *Creating the acp_userinfo File* on page A-249.

Zone Security

Every user can have a zone list assigned via remote ACP. If a list is not available via ACP, the RA 6300 provides all the zones it has learned from the network. If local security is used, use the per RA 6300 parameter **default_zone_list.** For more details, see *at_zone* on page A-263.

Logging

The RA 6300 logs activity and errors from the ARA session. The log is accessed via remote ACP and **syslog** (see *Logging User and RA 6300 Events* on page B-33 for more details).

Network-Visible Entity (NVE) Filtering

NVE filtering controls a remote access Apple user's view of network resources: when using *Chooser* to select resources, only the resource set defined for the user by the administrator will be visible. The administrator can specify the NVE filter on a per-user basis. This feature complements the existing zone list, described above, by offering a higher level of control.

The administrator uses the **nve_filter** entry in the **acp_userinfo** file to specify a list of filters on a per-user basis. Please see *at_nve_filter* on page A-265 for detailed information on creating **nve_filter** entries.

This method of limiting NBP traffic is not secure, and can be circumvented by a person willing to write some code to probe the network without using NBP. Also, this feature has no local RA 6300 security equivalent.

AppleTalk over PPP

AppleTalk over PPP allows Apple PowerBook and Macintosh computers to connect as an endpoint node to an AppleTalk network. The same Macintosh can also simultaneously run IP over the connection, allowing the user to use either IP or AppleTalk services as the need arises.

When the RA 6300 opens a PPP connection, it negotiates for link-level options, and then runs an optional security phase to authenticate the user. Finally, the two ends negotiate for network control protocol (NCP) options. The link is then opened and becomes a generic interface for the RA 6300. An AppleTalk point-to-point link is configured, enabled, and disabled using AppleTalk Control Protocol (ATCP).

The RA 6300 implementation of ATCP currently supports dial-in only.

See *Point-to-Point Protocol (PPP)* on page A-87, for information about setting PPP parameters.

Using the CCL Converter

The Macintosh CCL Converter application converts the CCL modem configuration file to allow access to the RA 6300 via ARAP (Versions 1 and 2). Typically, the CCL file sets up the modem and issues the dial command. When you dial into the RA 6300 from a Macintosh, the Macintosh CCL modem configuration file also controls the connection until protocol negotiation is complete.

The CCL Converter supports SecureID, Enigma, and CLI Dialback.

Configuring the CCL Converter

The RA 6300 administrator can configure the CCL Converter Application to connect to an RA 6300.

As administrator, configure the CCL Converter on the Macintosh as follows:

1. **Using a Macintosh-based** ftp **program, such as Fetch, that is set to MacBinary, copy the CCL Converter from the following directory on your UNIX load host:**

   ```
   usr/annex/src/examples/ccl_scripts
   ```

   The file name is CCL Converter.

   > You must ‘escape out’ the space between the word *CCL* and the word *Converter* in the CCL Converter file name. Do this by entering the file name inside a pair of double quotes, e.g., “CCL Converter”.

2. **From the Macintosh Settings menu, select ARAP V1/Autodetect Delay, CLI Security, or both, depending on the type of RA 6300 security you desire (see Table A-22).**

Table A-22. Selecting Security Type

CCL Setting	Port Parameter Settings	Security Result
ARAP V1/Autodetect Delay	at_security=Y	ARAP ACP security.
CLI Security	at_security=N	Normal (non-ARAP) ACP security, including port password and SecurID, if configured.
ARAP V1/Autodetect Delay *and* CLI Security	at_security=N	Normal (non-ARAP) ACP security, including port password and SecurID, if configured.
CLI Security	at_security=Y	Both ARAP ACP security and normal (non-ARAP) security.
ARAP V1/Autodetect Delay *and* CLI Security	at_security=Y	Both ARAP ACP security and normal (non-ARAP) security.
CLI Security	mode= arap (set in SPB)	Connection fails.
ARAP V1/Autodetect Delay *and* CLI Security	mode=arap (set in SPB)	Connection fails.

3. **If, in Step 2, you selected V1/Autodetect Delay, configuration is complete. Skip to *Running the Application,* below.**

   If you selected either CLI security or both, and you have modified the CLI prompt and/or the ACP Policy file on the RA 6300, you must use the CCL Converter's Customize menu to configure for the Macintosh any prompts you changed on the RA 6300.

   For example, RA 6300 administrators typically change the CLI prompt. Assume your customized RA 6300 prompt is the RA 6300 name followed by the port number. Under Customize on the Macintosh, select the *CLI Prompt...* menu item. A dialog box now appears. In the dialog box, enter the RA 6300 name portion of the RA 6300 prompt, but not the port number (since the user will be connecting to different ports) and either click **OK** or press **Return**. Make sure the RA 6300 name you enter is enclosed in quotes.

   Once all modifications are made, the configuration is complete. Go to *Running the Application*, below.

Running the Application

1. **Double-click on the CCL Converter application.**
2. **From the File menu, select Open (to convert a single file) or Open Folder (to convert a whole directory of files).**

   When you select a file, the application creates a new file containing the conversion, and appends ANNEX to the filename. For example, if the selected file is named AE Datalink PB, the application creates a new file named AE Datalink PB ANNEX. Names longer than 31 characters are truncated.

   Files with exactly 31 character names cannot be converted and cause the application to abort.

3. Quit the CCL Converter.

4. Select the converted CCL file from the remote access client (see your Apple Remote Access Client® documentation).

Chapter 13
Printers

The Remote Annex 6300 does not support printers.

Chapter 14
Configuring Hosts and Servers

Chapter 14 in the *Remote Annex Administrator's Guide for UNIX* describes the following configuration requirements for hosts providing the following Annex services:

- Accessing 4.2BSD hosts.
- Installing the ACE/Server software.
- Setting up the file server.
- Parsing the configuration file.
- Setting up the configuration file.
- Setting up the **motd** file.
- Setting up an Annex as a boot server.
- Self-booting without a local Ethernet interface.
- Installing a time server.
- Dump host services.
- Setting up name servers.
- Setting up a host for 4.3BSD syslogging.
- Configuring LAT services.

For the RA 6300, please note the following changes:

- The subsection *Installing Software Using bfs* in the section *Accessing 4.2BSD Hosts* is not up to date. For current information, see *Software Installation Notes for the Remote Annex 6300.*
- In all sections, configuring port parameters by number is not applicable. Instead, you leave a *port* or *port_set* blank to specify all ports on a particular Annex. When creating macros, you can leave the *port* or *port_set* blank, or you can specify **v** to designate virtual ports.

- In the section *Parsing the Configuration file*, please note that a new configuration section has been added for the RA 6300. The new section begins with the keyword **pri** and contains Session Parameter Blocks that define how the Annex handles PRI calls. Detailed information on entering SPBs is contained in *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51.
- The section *Creating modem Entries in the Configuration File* is not applicable. The RA 6300 supports only one type of modem (Microcom) and that type is hard-coded.
- The section *Host Initiated Connections* is not applicable because the RA 6300 does not support any physical ports for printing.
- The section *Creating rotary Entries in the Configuration File* is not completely accurate for the RA 6300. To understand what is different, see *The Port Server and Rotaries* on page A-77.
- The section *Creating dialout Entries in the Configuration File* is not completely accurate for the RA 6300. To understand what is different, see *Dial-up Networking* on page A-133.

Book A

Chapter 15
Using RA 6300 Security

The entire Annex product family, including the Remote Annex 6300 (RA 6300), provides comprehensive security features that assist you in securing your network from unauthorized access. Using these features, you can select between the following types of security:

- Local password protection (where the passwords are stored on the RA 6300).
- Host-based security (where at least one host on the network is functioning as a security server).

If unauthorized users can access your RA 6300, we strongly suggest that you enable security after loading the host code and booting the unit.

About RA 6300 Ports

Rather than a set of physical ports, the RA 6300 implements a set of internal ports which use the B channels that make up an ISDN PRI line. Each internal port is configured during call set up for the following operational characteristics:

- The type of call, which can be one of the following:
  - Modem (asynchronous).
  - Terminal Adapter (V.120).
  - Synchronous PPP.
- The ISDN PRI B channel assigned to carry the session (call).
- The internal RA 6300 port assignment (asy, ta, or syn).

Once the call is set up, a “session” between the remote device and resources accessible through the RA 6300 port commences. The session is maintained until terminated.

However, since the telephone network switch from which the PRI line to the RA 6300 originates determines which ISDN B channel is assigned to a internal port, static (i.e., port-specific) configuration of RA 6300 internal ports is not possible.

For example, you cannot guarantee that a user connecting via a remote PC will always use ISDN PRI B channel 9 and RA 6300 internal asynchronous port asy12 every time the user connects through the RA 6300.

As a result, the RA 6300 relies on a set of parameters to configure each port when performing the negotiations necessary to set up a call and establish a session (i.e., dynamic configuration). See *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51, for more information.

For RA 6300 security, this means that security should be applied to RA 6300 internal ports in global way (i.e., to all ports of specific type (asy, ta, and syn)) applying security based on types of connections.

Internal Port Types

The internal port types the RA 6300 uses to handle calls and other RA 6300 activities are the following:

This RA 6300 Port Type:	Applies to:
pts	virtual connections on the RA 6300 and all other Remote Annex servers
asy	modem connections on RA 6300 and all other Remote Annex servers
ta	V.120 TA connections on the RA 6300
syn	synchronous PPP connections on the RA 6300
ctl	internal RA 6300 control ports **Note:** Internal control port #1 (ctl1) is the RA 6300’s console port and is the only one available. Other control ports are reserved for future use.

Port Passwords and the RA 6300

Since there is not any way to predict which RA 6300 internal port a user will establish a session through, you cannot associate passwords with specific RA 6300 internal ports.

Guidelines for Creating ACP-related Files

To create ACP-related files for use with the RA 6300 that will not impede its operation, observe the following guidelines:

- Do not specify ports by number, range of numbers or groups of numbers (e.g., ports=1-10) in security profiles.
- Do specify ports, when appropriate, by port type (e.g., **ports=ta**) in security profiles.
- Do not specify port passwords, within the **acp_passwd** file.
- Use the correct RA 6300 name or IP address when specifying an RA 6300 within the **acp_dialup** file but do not specify ports by number, range, or group.

Information for Users of Other Remote Annexes

Read this section thoroughly if RA 6300s are installed in an environment with other Remote Annexes (e.g., a Remote Annex 4000).

If you are using other Remote Annex products, your existing ACP-related files should not be used with the Remote Annex 6300 without modification.

If you use your existing ACP-related files with the RA 6300, modifying them only to specify the RA 6300's IP address (when required), you will affect the operation of the RA 6300 in the following ways:

- Specified ports (e.g., ports=1), port ranges (e.g., ports=1-10), and port groups (e.g., ports=1,3,6,7,12,17) within profile criteria of your existing ACP-related files will be recognized and applied by the RA 6300 to internal RA 6300 asynchronous ports (asy ports).

  However, the RA 6300 will ignore ports specified by number, range of numbers, or group of numbers for internal V.120 terminal adapter (ta) ports and synchronous PPP (syn) ports and will apply profile criteria to all RA 6300 ta and syn ports.
- Port passwords specified in the **acp_passwd** file will affect RA 6300 operation negatively since the passwords will be matched to the equivalent RA 6300 internal asynchronous port. As a result, asynchronous modem connections made to an RA 6300 are subject to random password associations (i.e., each prompt asking for a different password).

Editing Existing ACP Files

When editing existing ACP files for use with the RA 6300, keep in mind that ACP profile criteria cannot be port specific and that you need to modify your existing files accordingly. For example, an **acp_userinfo** file user-end block that uses **ports=** for an RA 4000 on a per-port basis, as shown below, should be modified for the RA 6300:

```
user group=eng;ports=1-5
climask telnet end
end
```

Modifying the file as shown will apply the climask to all modem connections on any server, not just ports 1-5:

```
user group=eng;ports=asy
climask telnet end
end
```

If the former were used in an environment with the RA 6300, the climask would inconsistently apply to modem connections because the dynamic allocation of ports will not be constrained to ports 1–5.

About Setting Security-related Parameters

The network administrator determines which global security parameters to change and which to set for specific sessions (by creating or editing Session Parameter Blocks (SPBs) within the Annex configuration file). For more details, see *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51.

About Local Password Protection

Local password protection can be defined for access through virtual ports (i.e., sessions). Local password protection does not provide logging of security events to the security server. If event logging is enabled, user activities can be logged to **syslog** with local password protection (see *Event Logging Using syslog* on page B-37).

Local password protection can be used as a stand-alone security mechanism or as a back-up to host-based security. It validates access from either the device or the network. Local password protection supports **cli**. The local password protection policy cannot be altered.

About Host-based Security

The Access Control Protocol (ACP) provides host-based security in which a UNIX host on the network is defined as a security server. You can modify the host-based software to implement a security policy that fits the needs of your environment.

enable_security Parameter

To use any security feature, you *must* enable security for the RA 6300 by setting the **enable_security** parameter to **Y**. This parameter is mandatory if you intend to use any security mechanisms (except the administrative password for access to administrative tools).

If the **enable_security** parameter is set to **N**, no security is used, and no logging is performed regardless of any other parameter setting.

The **enable_security** parameter does not take effect until the RA 6300 is either rebooted or reset.

Overview of Local Password Protection

Local password protection allows you to assign a password that a user must enter before accessing an RA 6300. Because this password is stored locally on the RA 6300, it does not require a remote security server. Local password protection can be used as a back-up security mechanism in case the host-based security servers are unavailable.

The passwords set in the following parameters are stored on the RA 6300 and do not involve the use of a security server:

- **password**.
- **port_password**.
- **ppp_password_remote**.
- **vcli_password**.

The following subsections describe local security on the RA 6300 provided the **enable_security** parameter is set to **Y** and a host-based security server is not available.

Implementing Local Virtual CLI Password Protection

Local password protection can be implemented for the RA 6300 in one of two ways:

- Upon virtual CLI (VCLI) connection.
- Upon access through administrative utilities.

The **vcli_password** parameter allows you to define a local password for VCLI connections. The user enters only a password as opposed to a user name and password.

To configure the RA 6300 for a local VCLI password protection:

1. **Enable local security by setting the enable_security parameter to** Y**.**
2. **Disable VCLI remote security by setting the** vcli_security **parameter to** N**.**
3. **Define a password using the** vcli_password **parameter.**

The RA 6300 acts as follows:

- If the **vcli_password** parameter is not set (**“<unset>”**) and the **vcli_security** parameter is set to **N**, the RA 6300 prompts for the password specified by the **password** parameter.
- If the **password** parameter is not set (**“<unset>”**), the RA 6300 fails the VCLI attempt.
- If the **vcli_security** parameter is set to **N** and the **vcli_password** parameter is set (**“<set>”**), the RA 6300 prompts for the password specified in **vcli_password**.
- If the **vcli_security** parameter is not set (**N**) and the **vcli_password** parameter is not set (**“<unset>”**), the RA 6300 does not perform a security check for VCLIs.
- If the **vcli_security** parameter is set to **Y**, the **vcli_password** parameter is not set (**“<unset>”**), and the **password** parameter is not set (**“<unset>”**), the RA 6300 denies access to the VCLI if the security server is unreachable.

You can also use the **vcli_password** as a back-up to host-based security. When local VCLI password protection is used as a back-up, the RA 6300 first accesses the security server to validate a CLI connection request. If no response is received from a security server, the RA 6300 requests the local VCLI password. The user can enter either the VCLI password or the RA 6300 administrative password.

To set up the local VCLI password for back-up security:

- Enable security by setting the **enable_security** parameter to **Y**.
- Enable VCLI security by setting the **vcli_security** parameter to **Y**.
- Define a password using the **vcli_password** parameter.
- Define a security server host using the **pref_secure1_host**, **pref_secure2_host**, or **security_broadcast** parameter (e.g., 0.0.0.0).
- Create a password file on the security server (see *Creating User Password Files* on page 15-247).

If the remote server(s) fail:

- Access is permitted only through the VCLI password.
- No access is permitted if the **vcli_password** parameter is not set.

The **show annex** command does not display the value of the **vcli_password** parameter. Instead, it displays "<set>" or "<unset>".

Administrative Password

The RA 6300 administrative password protects the administrative tools; the default administrative password is the RA 6300's IP address. When the **show annex** command displays the password as **"<unset>"**, use the default administrative password for:

- Access to superuser CLI commands.
- Access to sessions locked with the CLI **lock** command.
- Access to a virtual CLI connection through local password protection (if the **vcli_security** parameter is set to **N**).

Modifying the assigned administrative password enables password protection on access to RA 6300s through **na** and **admin**.

The administrative password validates access to an RA 6300 through **na** only when security is enabled and the password is defined. Also, it can be used as the VCLI password and to override the password assigned with the CLI **lock** command.

The administrative password never displays. If you forget the modified password, you can reset it only by erasing the RA 6300’s non-volatile memory using the ROM Monitor **erase** command, and re-entering all parameters.

As a safeguard against losing the unit’s current configuration, use the **na** command **write** to save the RA 6300 and global port parameter settings; if necessary, you can restore these settings using the **read** command (for more details on using these commands, see *na Commands* on page C-1).

Protecting the Superuser CLI

An RA 6300 administrative password is required for access to the superuser CLI. The default password is the RA 6300’s IP address. There are two ways to change the password:

- Using the superuser CLI **passwd** command.
- Changing the **password** parameter using **na** or **admin**.

Using either method, the new password takes effect immediately for access to the superuser CLI. Reset the password to the RA 6300’s IP address by either:

- Using **na** or **admin** to set the **password** parameter to the null string:

```
command: set annex password ""
```

- Using the superuser CLI **passwd** command and pressing **Return** in response to the prompt for a new password.
- Erasing all parameters using the ROM monitor **erase** command.

Protecting Resources from Unauthorized Access

When terminals are connected to a network, they provide users with the potential for unauthorized access to network resources. In addition to the available security schemes, the RA 6300 provides timers that can terminate a session. The default **cli_inactivity** parameter (default setting: off) sets the CLI inactivity timer. When enabled, the RA 6300 terminates the session after the amount of time specified in this parameter has elapsed or the last session is completed.

Users can protect their login sessions using the CLI **lock** command if they do not want to log out when leaving the terminal unattended.

Protecting the na Utility from Unauthorized Access

When using the **na** utility, users can access RA 6300 parameters and obtain useful information, or reconfigure and reboot RA 6300s. Protecting **na** involves UNIX superuser protection and the RA 6300 administrative password.

Upon installation, **na** is owned by root and executable by all. Only a superuser can execute the **set**, **reset**, **broadcast**, **dumpboot**, **boot**, **read**, and **copy** commands.

Overview of Host-based Security

ACP security has three requirements: 1) at least one UNIX host on the network must act as a security server running RA 6300 security software; 2) security must be enabled on the RA 6300 (the **enable_security** parameter is set to **Y**); and 3) a security regime, such as **acp** or **securid**, must be defined for authenticating RA 6300 users.

The security server maintains a database of files that reside by default in the directory **/usr/annex**. These files include:

- **acp_keys** (encryption key information).
- **acp_dialup** (user names and addresses for dial-up connections).
- **acp_group** (user-group associations for security).
- **acp_regime** (security authentication system and associated password file name).
- **acp_passwd** (security passwords).

  Do not specify port passwords for the RA 6300.

- **acp_userinfo** (initial login environment and start-up CLI commands).
- **acp_restrict** (restricted hosts and host ports).
- **acp_logfile** and **acp_logfile.*Annex_IPaddress*** (security audit trails).

  The contents of these files should match on all security servers (except for **acp_logfile**).

The following sections describe these aspects of ACP security:

- *Basic ACP Configuration* on page 15-225.
- *Encrypting Security Messages* on page 15-232.
- *Defining Security Profiles* on page 15-235.
- *Using include Files in the acp_userinfo File* on page 15-274.
- *Dynamic Allocation of Network Addresses* on page 15-274.
- *Enhancing Password Security* on page 15-279.
- *Using AppleTalk Security* on page 15-291.
- *Using IPXCP Security* on page 15-293.
- *Using PPP Security* on page 15-293.
- *Using Kerberos Authentication* on page 15-301.
- *Using the ACE/Server* on page 15-303.
- *Using SafeWord AS Security* on page 15-314.
- *Configuring Security for the RA 6300 FTP Daemon* on page 15-321.
- *Configuring the IP Basic Security Option (IPSO)* on page 15-323.
- *Logging Security Events* on page 15-324.
- *Modifying the Supplied Security Application* on page 15-325.

Basic ACP Configuration

This section outlines procedures for configuring the basic ACP features and describes what happens in each case if ACP goes down.

CLI Security

You can set up host-based security for CLI connections in which users must provide a valid user name and password before they are granted access to a CLI:

1. **Set the** cli_security **parameter to** Y**, so that the RA 6300 will use ACP.**
2. **Define a security server using the** pref_secure1_host**,** pref_secure2_host**, or** security_broadcast parameter **(see *Configuring the Security Server* on page 15-230).**
3. **Create entries in the** acp_regime **file defining the authentication systems to be used and the conditions under which to use them.**

   The install program creates the **acp_regime** file, then prompts you for a default regime and (in some cases) a password file name, which it enters into **acp_regime**. Subsequently, you can add to and/or change the contents of this file. (See *Configuring the acp_regime File* on page 15-245.)
4. **Create entries in the appropriate password files (see *Creating User Password Files* on page 15-247).**
5. **(Optional) Configure encryption for security messages (see *Encrypting Security Messages* on page 15-232).**

If ACP is down, the RA 6300 acts as follows:

- First, the RA 6300 prompts for the password specified in the **port_password** parameter. If the **port_password** parameter is not set (**“<unset>”**), the RA 6300 does not connect the user to the CLI.

- If the **cli_security** parameter is set to **N** and the **port_password** parameter is set (**“<set>”**), the RA 6300 prompts for the password specified in **port_password**.
- If **cli_security** is set to **N** and the **port_password** parameter is not set (**“<unset>”**), the RA 6300 does not perform a security check for CLI connections and allows unrestricted access to the CLI.

Virtual CLI Security

You can set up host-based security for virtual CLI (VCLI) connections in which users must provide a valid user name and password before they are granted access to a virtual CLI:

1. **Set the** vcli_security **parameter to** Y**, so that the RA 6300 will use ACP.**
2. **Define a security server using** pref_secure1_host**,** pref_secure2_host**, or** security_broadcast **parameter (see *Configuring the Security Server* on page 15-230).**
3. **Create entries in the** acp_regime **file defining the authentication systems to be used and the conditions under which to use them.**

   The install program creates the **acp_regime** file, prompts you for a default regime and (in some cases) password file name, and then enters them into **acp_regime**. Subsequently, you can add to and/or change the contents of this file (see *Configuring the acp_regime File* on page 15-245).
4. **Create entries in the appropriate password files (see *Creating User Password Files* on page 15-247).**
5. **(Optional) Configure encryption for security messages (see *Encrypting Security Messages* on page 15-232).**

If ACP is down, the RA 6300 acts as follows:

- First, the RA 6300 prompts for the password specified in the **vcli_password** parameter (see *Implementing Local Virtual CLI Password Protection* on page 15-218).
- If the **vcli_password** parameter is not set (**“<unset>”**) and the **vcli_security** parameter is set to **N**, the RA 6300 prompts for the password specified by the **password** parameter.
- If the **password** parameter is not set (**“<unset>”**), the RA 6300 fails the VCLI attempt.
- If the **vcli_security** parameter is set to **N** and the **vcli_password** parameter is set (**“<set>”**), the RA 6300 prompts for the password specified in **vcli_password**.

Connection Security

You can authorize or deny IP or CLI access to specific hosts, host ports, or networks for a particular user, group, time of day, or protocol in use.

1. **Define a security server using the** pref_secure1_host**,** pref_secure2_host**, or** security_broadcast parameter **(see *Configuring the Security Server* on page 15-230).**
2. **Set the** connect_security **parameter to** Y**, so that the RA 6300 uses ACP on a CLI connection (via** telnet **and/or** rlogin**).**
3. **Configure the** acp_restrict **file on the security server (see *Limiting Access to Hosts via acp_restrict* on page 15-269).**

   For CLI **telnet** or **rlogin** connections, ACP checks **acp_restrict** to see whether or not access should be granted to the user. For SLIP and IP over PPP connections, the **acp_restrict** file controls access by creating filters based on your input.
4. **(Optional) Configure encryption for security messages (see *Encrypting Security Messages* on page 15-232).**

SLIP and PPP Security

You configure access to a SLIP or PPP link from the RA 6300 as follows:

1. **Set the** mode **parameter to** cli **and have the user issue the** slip **or** ppp **command from the CLI.**

   If the **mode** parameter is set to **slip**, the RA 6300 does not perform a security check.

2. **If you want authentication performed on the CLI connection (rather than authenticating when the user issues the** slip **or** ppp **command), set the** slip_ppp_security **and** cli_security **parameters to** Y**. Then proceed to step 4.**

3. **To have authentication performed when the user issues the** slip **or** ppp **command (rather than authenticating when the CLI connection is made) set the** cli_security **parameter to** N **and the** slip_ppp_security **parameter to** Y**.**

4. **Define a security server using the** pref_secure1_host**,** pref_secure2_host**, or** security_broadcast parameter **(see *Configuring the Security Server* on page 15-230).**

5. **Create entries in the** acp_regime **file defining the authentication systems to be used and the conditions under which to use them.**

   The install program creates the **acp_regime** file, then prompts you for a default regime and password file name, which it enters into **acp_regime**. Subsequently, you can add to and/or change the contents of this file. (See *Configuring the acp_regime File* on page 15-245.)

6. **Create entries in the appropriate password files (see *Creating User Password Files* on page 15-247).**

7. **(Optional) Configure encryption for security messages (see *Encrypting Security Messages* on page 15-232).**

If ACP is down, the **slip** or **ppp** command fails.

The RA 6300 never uses local security with the **slip** or **ppp** command.

Port Server Security

You can set up security for port servers in which users must provide a valid user name and password before they are granted access to an outgoing port:

1. **Set the** port_server_security **parameter to** Y**, so that the RA 6300 will use ACP.**
2. **Define a security server using the** pref_secure1_host**,** pref_secure2_host**, or** security_broadcast parameter **(see *Configuring the Security Server* on page 15-230).**
3. **Create entries in the** acp_regime **file defining the authentication systems to be used and the conditions under which to use them.**

   The install program creates the **acp_regime** file, then prompts you for a default regime and password file name, which it enters into **acp_regime**. Subsequently, you can add to and/or change the contents of this file. (See *Configuring the acp_regime File* on page 15-245.)
4. **Create entries in the appropriate password files (see *Creating User Password Files* on page 15-247).**
5. **(Optional) Configure encryption for security messages (see *Encrypting Security Messages* on page 15-232).**

If ACP is down, the RA 6300 acts as follows:

- If the **port_server_security** parameter is set to **Y**, the RA 6300 prompts for the password specified in the **port_password** parameter.

- If the **port_password** parameter is not set (**“<unset>”**), the RA 6300 fails the port connection attempt.
- If the **port_server_security** parameter is set to **N** and the **port_password** parameter is set (**“<set>”**), the RA 6300 prompts for the password specified in **port_password**.
- If the **port_server_security** parameter is set to **N** and the **port_password** parameter is not set (**“<unset>”**), the RA 6300 does not perform a security check for port connections.

Configuring the Security Server

The ACP security server software is provided as part of the expedited remote procedure call daemon (**erpcd**) software. Included with the software is the **eservices** file that has two entries: one for the block file server (**bfs**) and one for ACP.

The **erpcd** process must be running; **erpcd** requires the **/etc/services** file to have an entry for *erpc 121/udp*.

Setting Up a Security Server

To set up a security server, you must install the file server software on a host and delete the # symbol in front of the ACP entry in the **eservices** file. For example:

```
# erpc remote programs
#
# prog no. verlo       verhi     name
#
  1       0            99        bfs
  3       0            99        acp
```

Specifying the Security Hosts

The **pref_secure1_host** and **pref_secure2_host** parameters specify the preferred security hosts. The RA 6300 first queries the **pref_secure1_host** for user validation. If a response is not received within the time defined in the **network_turnaround** parameter, the RA 6300 repeats the query several times. If the RA 6300 still does not receive a response, it queries the host defined in the **pref_secure2_host** parameter. If a response is not received from the second security host within the allowable time limit, and the **security_broadcast** parameter is set to **Y**, the RA 6300 broadcasts to the network for another host with **erpcd** running to authorize the access request. If the **security_broadcast** parameter is set to **N**, the RA 6300 denies the authentication request.

The **network_turnaround** parameter specifies the amount of time in seconds in which the RA 6300 expects a response from the security servers. To reduce the possibility of a retry, the network turnaround time should be long enough to allow for a network transmission to the security server and transmission back to the RA 6300; unfortunately, if this period of time is too long, the RA 6300 will attempt multiple retries before sending a query to the second security server.

Disabling Broadcasting for Security Servers

The RA 6300 broadcasts to the network for a security server if:

- The **security_broadcast** parameter is set to **Y**.
- The **pref_secure1_host** and **pref_secure2_host** parameters do not respond.

Setting the **security_broadcast** parameter to **N** disables RA 6300 broadcasting. If the hosts defined in the **pref_secure1_host** and **pref_secure2_host** parameters do not respond, the RA 6300 refuses the connection request.

Encrypting Security Messages

Messages between the security server and the RA 6300 are encrypted if the RA 6300 parameters **enable_security** and **acp_key** are set. The parameters do not take effect until the RA 6300 is either rebooted or RA 6300 security is reset.

The **acp_key** parameter specifies the encryption key the RA 6300 uses to exchange messages with the security server. The security server maintains the encryption key for each RA 6300 in the **acp_keys** file (see *Creating the acp_keys File* on page 15-233 and *Configuring Hosts and Servers* on page A-209).

The encryption key also validates the security host: the host must know the RA 6300's ACP key for the RA 6300 to consider the host valid. Without the appropriate key, the RA 6300 denies the user's request even if the host is defined as a preferred security host.

The **show annex** command does not display the value of the **acp_key** parameter. Instead, it displays **"<set>"** or **"<unset>"**.

Creating the acp_keys File

The security server maintains the encryption key for each RA 6300 in the **acp_keys** file. Each entry in this file contains a list of RA 6300 names or IP addresses separated by commas and an encryption key for those RA 6300s. The RA 6300 or the list of RA 6300s and the key are separated by a colon. The order of placement in the file is important, as the file is read sequentially.

When the security server receives an encrypted message from the RA 6300, the server tries to match that key against the key assigned to the RA 6300 in the file. If no match exists, the RA 6300 and the server cannot communicate.

The syntax rules for the **acp_keys** file are:

- Any part of an IP address in the list can be specified with an asterisk (*).
- A backslash (\) is used to continue a line.
- Any ASCII character except spaces and tabs are valid encryption keys (keys are case sensitive).
- Each key can contain a maximum of fifteen characters.

RA 6300s with no entries are assumed to have no key set. Since wildcards are valid, some entries in the file may require an explicit “no key” declaration:

```
annex01, annex02:   seKret2
#131.21 net Annexes have the same key except for 3 Annexes
131.21.2.1, 131.21.2.2:
131.21.1.1: SpeciaL
131.21.*: Gub-Net
```

In the following example, the first three entries specify *insomniac-1* as the key for the RA 6300 whose IP address is 132.245.6.15, no encryption for the RA 6300 whose IP address is 132.245.6.75, and *Piano* as the key for all other RA 6300s on the 132.245.6 subnet. The last entry specifies *gl12ch* as the key for *annex01*, *annex02*, and *annex03*. Each **acp_key** parameter for the RA 6300s listed in the example must be identical to the key included in the **acp_keys** file.

```
132.245.6.15:insomniac-1
132.245.6.75:
132.245.6.*:Piano
annex01,annex02,annex03:gl12ch
```

Changing the value of the **acp_key** parameter on any RA 6300 requires the same change to the **acp_keys** file on the security server. The recommended order for changing the ACP encryption key on an RA 6300 is:

1. **Edit the** acp_keys **file on all security server hosts.**
2. **Change the value of the** acp_key **parameter for all affected RA 6300s.**
3. **Update the cache by sending the erpcd on all security server hosts a HUP signal with** kill**.**

   ```
   kill -HUP <pid_number>
   ```

4. **Reset the security subsystem for all affected RA 6300s using the** na **command** reset annex security**.**

Defining Security Profiles

The expedited remote procedure call daemon (**erpcd**) that implements ACP permits you to define different security profiles for different users, groups of users, or for other *connection conditions*, such as the time of day or the date. Specifically, you can use the **acp_regime**, **acp_userinfo**, and **acp_restrict** files to create diverse security profiles based on any combination of the *profile criteria* shown in Table A-23.

Table A-23. Profile Criteria

Criterion	Description
username	The user's userid.
group	The name of a group to which the user belongs, as defined in the **/etc/groups** or **acp_group** file; see *Creating User Groups* on page 15-244.
time	The day of the week and/or the time of day.
protocol	The connection protocol (e.g., PPP, CLI).
annex	The name or IP address of the RA 6300 on which the connection is made.
port type	The RA 6300 internal port type on which the connection is made.

Overview of Security Profile Criteria

Security profile criteria specify the connection conditions that must be met in order for the RA 6300 to:

- Use a particular security regime for authentication (in **acp_regime**).
- Define the user environment that will be in effect upon login (in **acp_userinfo**).
- Permit or restrict access to hosts or host ports (in **acp_restrict**).

Together, the security regime, user environment, and host access restrictions define the security profile.

A profile criterion begins with one of the keywords listed in Table A-23 on page A-235. The keyword is followed by an = sign, which is followed by a value. No space is permitted before or after the = sign. The syntax is:

keyword=*value*

To enter more than one criterion, separate the criteria with semicolons(;). Keep the criteria on one line. Use the backslash (\) continuation symbol to extend the line beyond the right margin, if necessary. No spaces are allowed on either side of the semicolon or within the *value* field, with the exception of data for the **time** criterion (see *Time* on page 15-240). A particular keyword may appear only once in a line of criteria. An entire line of profile criteria is called a *profile criteria specification*. The following examples could appear in any or all of the **acp_regime**, **acp_userinfo**, and **acp_restrict** files.

For the **acp_regime** and **acp_restric**t files, the entry looks like this:

```
username=chris;time=”9:00am – 10:30pm Monday-Friday”;annex=annex03
```

For the **acp_userinfo** file, the entry looks like this:

```
user username=chris;time=”9:00am–10:30pm Monday-
Friday”;annex=annex03
```

A profile criteria specification cannot exceed 80 characters.

After the profile criteria specification, you specify the security measure(s) to be applied if the criteria are met. The following is an example from the **acp_userinfo** file:

```
user username=chris;time=”9:00am-10:30pm Monday-Friday”;annex=annex03
climask ppp end
end
```

When user *chris* connects to *annex03*, **erpcd** records all the conditions related to the connection – the userid and any group associations (as defined in the **acp_group** or **/etc/group** file), the RA 6300 and port that *chris* connects to, the time of connection, and the connection protocol – CLI, PPP, or SLIP. **erpcd** saves these connection conditions for comparison with profile criteria specifications in the **acp_regime**, **acp_userinfo**, and **acp_restrict** files.

All of the profile criteria in a specification must be met in order for **erpcd** to consider that the specification *matches* the connection conditions recorded. The specification for Chris is considered a match if he gives a user name of *chris* and connects to *annex03* between 9:00 A.M. and 10:30 P.M Monday through Friday. If all of these criteria are met, user *chris* is prevented (via the *climask* entry) from issuing the **ppp** command while he is logged into *annex03*.

Note that, in the example cited, the profile criteria specification replaces the **username** field in **acp_userinfo**. The **user** field can be specified instead, for compatibility with earlier RA 6300 releases. For more information, see *Creating the acp_userinfo File* on page 15-249.

One Match per File

You can enter an unlimited number of profile criteria specifications in each of the **acp_regime** and **acp_userinfo** files. However, for any single set of connection conditions, **erpcd** uses only the first matching specification it finds in each file. Consequently, the placement of profile criteria specifications is important. For example, suppose that user *chris* belongs to a group named *engineering* and that the first line in **acp_regime** specifies that the *engineering* group should be authenticated via Kerberos, while the second line specifies that user *chris* should be authenticated by SecurID. The result is that *chris* is authenticated by Kerberos, since a match for the group entry is found first.

The first-match algorithm is also true for **acp_restrict** entries that apply to CLI (**telnet** and **rlogin**) connections. However, **acp_restrict** entries for PPP and SLIP are treated differently (see *Limiting Access to Hosts via acp_restrict* on page 15-269).

The Resulting Security Profile

Once **erpcd** has found all the matching profile criteria in **acp_regime**, **acp_userinfo**, and **acp_restrict** (using the one-match-per file rule where appropriate) for a given set of connection conditions, the result is a single security profile.

Profile Criteria Syntax

The following sections give the purpose and syntax for each of the different criteria you can include in a profile criteria specification. Additional information and examples are supplied in the sections on **acp_regime**, **acp_userinfo**, and **acp_restrict**.

Username and Group Criteria

The **username** criterion lets you control security based on the RA 6300 userid (the name the user specifies at login).

The **group** criterion lets you control security based on a user's membership in a group. You assign users to groups via either the **acp_group** file or the **/etc/group** file (see *Creating User Groups* on page 15-244). When a **group** profile criterion is specified, **erpcd** checks the **acp_group** file to find the users belonging to the group. If it cannot find an **acp_group** file, **erpcd** looks in the **/etc/group** file.

A wildcard (*) can be used to represent as many of the final characters in a **username** or **group** as can be removed and still leave the name unique. The following are examples of **username** and **group** criteria:

```
username=fritz
username=fri*
username=frank
username=fra*
group=finance
group=fi*
group=fun
```

The following designates all users:

```
username=*
```

Time

The **time** criterion lets you control security based on the day of the week, the date, and the time of day. The following are the four possible syntaxes:

```
time="day"
time="time1-time2 day1-day2"
time="time1 day1 – time2 day2"
time="time1 date1 – time2 date2"
```

Enclose the **time** criterion in quotation marks and specify the arguments as follows:

- For *day*, specify a weekday, e.g., Sunday or Monday. The time criterion will apply to that entire day. Weekday specifications observe minimum uniqueness and are not case sensitive.
- For *time1*, specify the beginning of a time range; for *time2*, specify the end of a time range. Use *hh*:*mm*[**am**|**pm**], where *hh* is the hour and *mm* is the minutes, as the format for each end of the range. If you do not include **am** or **pm**, the Remote Annex assumes you are using military (24-hour) notation. Both ends of a range must use the same type of notation – you cannot use military time for one part of a range and **am** or **pm** for the other.

  To indicate midnight, specify either 12:00am or 00:0. Specify noon as 12:00pm or 12:00. To indicate a 24-hour range, use either 00:00 – 23:59 or 12:00am – 11:59pm.

  Be sure to include the colon and minutes (**:***mm*) after the hour (*hh*). For example, 9:00am – 5:00pm is valid; 9am – 5pm is not.

You cannot specify time ranges without also specifying either a range of days or a range of dates. A time range with only a single day or date is not permitted. For example, *time= 9:00am – 5:00pm Sunday* is invalid. The correct usage would be *9:00am – 5:00pm Sunday-Sunday.*

- For *date1*, specify the beginning of a month and day range, e.g., January15, February10; for *date2* specify the end of a month and day range. Month specifications observe minimum uniqueness and are not case sensitive.

The following are examples:

```
time=”9:00am-5:00pm Monday-Friday”
time=”9:00-22:00 Sunday”
time=”Wed”
time=”8:00AM Friday - 6:35PM Friday”
time=”10:30 Nov 30 - 21:30 Nov 31”
```

The **time** criterion applies to initial access by the user. For instance, in the first example above, the criterion is met if the user logs in at any time between 9:00 A.M. and 5:00 P.M. on Monday through Friday of any week in any month.

RA 6300 and Port Type

The **annex** and **port type** criteria let you control security based on the RA 6300 and RA 6300 port type that the user tries to access. You can use an asterisk (*) symbol as a wild card in place of an RA 6300 name or the host port of an RA 6300 IP address. The following are valid **annex** and **port type** specifications:

```
annex=annex03;ports=asy
ports=syn
annex=192.17.5*
annex=*
```

The first example specifies all internal asy ports (asynchronous ports) on *annex03*.

In the second example, *annex*=* is implied and the user can access all internal syn ports (synchronous PPP ports) on all RA 6300s.

In the third and fourth examples, *ports*=* is implied. The fourth example specifies all ports on all RA 6300s, which is the default.

You cannot abbreviate the **ports** keyword.

Protocol

The **protocol** criterion lets you control security based on the protocol used to attempt access to a host or host port. Valid values are:

- **slip**
- **ppp**
- **cli** (for **telnet** and **rlogin**)

Specify a protocol criterion using the syntax:

protocol=*protocol_name*

To specify more than one protocol, you must specify multiple security profile definitions. For example, to specify both PPP and SLIP, enter:

```
protocol=ppp
protocol=slip
```

The default is any protocol.

Overview of Files Used to Define Security Profiles

Following are the files you use to define security profiles:

- **acp_group** or **/etc/group**. If you intend to assign different security profiles to different groups of users, you must first define the groups in the **acp_group** or **/etc/group** file.
- **acp_regime**. An initial **acp_regime** file is created by the Annex **install** program. It is based on answers you supply to prompts from **install,** and it contains a single authentication scheme, such as **acp**, to be used for authenticating all RA 6300 users. It also contains the name of a password file, if the regime is **acp** or **kerberos**.

  You can modify the initial **acp_regime** file so that different authentication schemes are used when particular criteria are met.

  > Do not confuse ACP, the RA 6300's Access Control Protocol that controls all host-based security, with **acp**, one of several authentication systems (regimes) that can be used with ACP.

- **acp_userinfo**. This file allows you to configure login environments based on a single userid or one or more profile criteria. Configurable aspects of login environments include CLI commands to be executed at start-up, CLI commands not permitted during a login session, filter and route definitions, a CHAP secret token, and various AppleTalk session characteristics. You can also use **acp_userinfo** to deny login access.
- **acp_restrict**. You can use this file to restrict access to hosts or host ports based either on the RA 6300 that attempts the connection or on specified access criteria.

The following sections describe these files in detail.

Creating User Groups

One of the most useful aspects of customizing security is the ability to apply different access rights and restrictions to different groups of users. To associate individual users with one or more groups, you create entries in the **/etc/group** or **acp_group** file.

The **/etc/group** file already exists on most UNIX systems, so you may prefer to add entries to this file rather than entering them in **acp_group**, which you must create. Either file must reside in the install directory (default is **/usr/annex**) on the UNIX security host.

erpcd looks for **acp_group** first, only using **/etc/group** if it cannot find **acp_group**. To designate that **erpcd** should use **/etc/group** rather than **acp_group**, see *Changing the Expected File Names Used by ACP* on page 15-328.

The **acp_group** file must have the same format as the /**etc/group** file. The following systems do not support the **acp_group** file: Ultrix, FreeBSD, and BSDI. On these systems, you must use the / **etc/group** file.

An **/etc/group** or **acp_group** file contains a one-line entry for each group. To retain compatibility with **/etc/group**, the **acp_group** file includes passwords and group ID fields, although ACP does not use them. Due to UNIX implementation, you must specify a value in each of these two fields, although what you choose to enter is arbitrary. The format for an ACP group entry in **acp_group** or /**etc/group** is:

groupname:*password*:*groupid*:*userlist*

The *groupname* field specifies the name of the group; the *userlist* field is a comma-separated list of user names belonging to the group. There is no arbitrary limit to the number of names in *userlist*. Fields are separated by the colon (:) character.

Following are two sample **acp_group** (or **/etc/group**) entries:

```
accounting:p:g:kim,herbert,sam,louise,bill
engineering:p:g:dilbert,jim,sharon,scott,john,liza,\
carrie,edna,dena,caroline,marsha,sue,don,phil,eric,\
dan,fritz,jeremiah,amy
```

The *p* and *g* in the previous examples are placeholders for the values that UNIX requires but ACP ignores.

Configuring the acp_regime File

The initial security regime that the RA 6300 uses to authenticate all users is defined in the **acp_regime** file. This file is created the first time the network administrator runs the RA 6300 install program. The program prompts for a security regime and, if the regime requires it, a password file name. The regimes from which you can choose are **acp**, **securid**, **safeword**, **kerberos**, **native**, and **none** (see Table A-24). Password files are required for **acp** and **kerberos**; the defaults are **acp_passwd** and **/temp/tkt_erpcd_**. Both the **acp_regime** file and the password files (if any) must be stored in the installation directory, which defaults to **/usr/annex**.

The **acp_regime** file created by **install** has the following format:

:*initial_regime*[**:***initial_password_file*]

Once the **acp_regime** file has been created by the **install** program, you can modify the file to specify more than one regime and to include profile criteria that determine the conditions under which different regimes are used. The syntax for an **acp_regime** entry is:

[*profile_criteria*]**:***regime*[**:***password_filename*]

Each field in the entry must be separated by the colon (:) character; a space may follow but not precede the colon. The syntax for *profile_criteria* is:

keyword=*value*[**;***keyword*=*value*;...]

If profile criteria are omitted, the specified regime applies to any user logging in under any circumstances. Profile criteria are explained in *Overview of Security Profile Criteria* on page 15-236. Valid regimes are explained in Table A-24.

The *password_filename* field is valid only for the **acp** and **kerberos** regimes. If you specify the **acp** or **kerberos** regime but supply no *password_filename*, the default is used (see Table A-24). If the file is not found, an error message is logged and access is denied.

Table A-24. Authentication Regimes

Regime	Description
acp	ACP authentication, using the password file you specify. Default is the **acp_passwd** file.
safeword	SafeWord authentication.
kerberos	Kerberos authentication, using the ticket-directory prefix you specify. Default is **/temp/tkt_erpcd_**.
native	Authentication via the native operating system of the security server. For UNIX, native means the **/etc/passwd** file is used for authentication.
none	No authentication is performed; the user is unconditionally authenticated.
securid	SecurID authentication.

The following is a sample **acp_regime** file:

```
username=jack;time=”9:00am-10:00pm Tuesday - Thursday”:securid
group=finance:acp:special_acppw
:acp
```

Given this sample, **erpcd** uses SecurID to authenticate user *jack* if he logs in between 9:00 A.M and 10:00 P.M. on the specified day.

Next, **erpcd** looks in the **acp_group** (or **/etc/group**, if **acp_group** does not exist) file to find the members of the group named *finance*. If one of these users tries to log in at any time on any day or date, **erpcd** attempts to authenticate that user via the **acp** regime, using the **special_acppw** password file (which must reside in the RA 6300 install directory). Even if user *jack* is defined in *finance*, if he logs in between 9:00 A.M. and 10: P.M., **erpcd** nevertheless tries to authenticate him via SecurID, since the profile criteria specification that begins with **username** is matched first.

Finally, any users whose login characteristics do not match the first two profile criteria specifications are authenticated via ACP, using the default password file, **acp_passwd**.

Creating User Password Files

Password Files for the acp Regime

If the security regime defined is **acp**, **erpcd** prompts the user for a user name and password. The RA 6300 validates this information by instructing the security server to compare these entries against entries in the password file specified in **acp_regime**. If no password file is specified, ACP uses **acp_passwd**, which must reside in the install directory (default is **/usr/annex**). In either case, if a match is found, the user is granted access; otherwise, the user is denied access.

A typical session looks like this:

```
Annex Command Line Interpreter * Copyright 1988, 1995 Xylogics, Inc.
Checking authorization, Please wait...
Annex username: kate
Annex password:
Permission granted
annex:
```

The **acp_passwd** file uses the same format as the **/etc/passwd** file. The easiest way to create this password file is to copy the **/etc/passwd** file to **acp_passwd**. One advantage to creating the **acp_passwd** file this way is that you can merge **/etc/passwd** files from different hosts into one file on the security server, thus allowing you to create a network-wide password file.

If you are using a System V.4 or V.5 host, use the **/etc/shadow** file rather than the **/etc/passwd** file.

Not all password files work with ACP. For example, you could not merge SCO UNIX password files into the **acp_passwd** file.

Non-superusers can change their passwords only if the *username* in the **acp_passwd** file matches the *username* in the **/etc/passwd** (or **/etc/shadow**) file on the host.

After creating this entry, use the **ch_passwd** command to enter the port password:

```
% ch_passwd 132.245.33.11.1
New password: <password>
```

This port password is independent of the port parameter **port_password**. The port parameter is used only for local security.

The ACP prompts appear as follows:

```
Annex username:
Annex password:
Port password:
```

Password File for the Kerberos Regime

If **kerberos** is defined in **acp_regime**, **erpcd** validates the user name and password by comparing them to entries in the password file specified in **acp_regime**. If no password file is specified, **erpcd** looks for /**temp/tkt_erpcd_** in the install directory (default is **/usr/annex**). If **erpcd** does not find a match in that file, the user is denied access to the RA 6300. For more information, see *Using Kerberos Authentication* on page 15-301.

Password Files for Other Regimes

For information on passwords used with third-party systems other than Kerberos, see the following sections:

- *Using the ACE/Server* on page 15-303.
- *Using SafeWord AS Security* on page 15-314.

Password Histories and Blacklisting

You can enhance security for passwords by configuring the RA 6300 to record password histories and to blacklist users who have a configurable number of failed login attempts (for more details, see *Enhancing Password Security* on page 15-279).

Creating the acp_userinfo File

The **acp_userinfo** file resides in the install directory and is maintained by the network administrator. The file primarily defines aspects of the user login environment. This environment can be defined on the basis of profile criteria and/or a userid (user name).

The information from **acp_userinfo** is loaded into the **erpcd** internal database. To update the database, send a USR1 signal to **erpcd** (**kill –USR1** *pid*). When updating **acp_userinfo**, it is a good idea to check syntax using the **erpcd –u** *filename* command (see *erpcd* on page C-115).

To create entries in the **acp_userinfo** file, use the following format, which is referred to as a **user...end** block:

user username={*name*| *profile_criteria*}
entry
:
:
end

The syntax for *profile_criteria* is:

keyword=*value* [**;***keyword*=*value***;**...]

Entering profile criteria is described in detail in *Profile Criteria Syntax* on page 15-239.

If you use the *name* argument instead of *profile_criteria*, specify a valid userid. This argument is supported for compatibility with Release 10.1 and earlier releases but is treated as if it were the profile criterion **username**=*name*. In searching **acp_userinfo**, **erpcd** looks only for a first match, whether that match is a single userid or all the criteria in a profile criteria specification.

The following is an example:

```
user username=jill
      climask slip ppp end
end
user group=finance;time=”8:00AM-6:00PM Monday-Wednesday”
      clicmd ppp end
user group=finance
      deny
end
```

In the above example, **user username=jill** can also be specified as **user jill.**

In this example, even if user *jill* is a member of the *finance* group and meets all of the criteria in that profile criteria specification, *jill* is not permitted to use **slip** or **ppp**, since the first match found is the userid *jill*. The remainder of the example specifies that the finance group is allowed to connect only if its members log in between 8:00 A.M. and 6:00 P.M. on the specified days. The CLI port they are connected to will be converted to **ppp** mode after the group members have been authenticated. At any other time, they are denied access.

You can specify the following *entry* options (the following subsections discuss these options in detail):

- **accesscode**
- **clicmd**
- **climask**
- **deny**
- **filter**
- **route**
- **at_zone**
- **at_connect_time**
- **at_nve_filter**
- **at_passwd**
- **chap_secret**

accesscode

accesscode is a string for which the user is prompted. You specify **accesscode** at the beginning of an accesscode entry. For each user, or for conditions that meet profile criteria, you can define one or more accesscode entries in the **acp_userinfo** file. Depending on the contents of the entry, one of several actions can occur, including dial-back. Each entry can include a phone number, inbound and outbound modem pools, and a job name (see Table A-25). The syntax is:

accesscode *code*

accesscode_entry

end

Table A-25. Entries for accesscode in the acp_userinfo File

Entry	Description
code	A character string defined by the administrator. The user is prompted for this string (after the **user name and password** prompts) when logging onto a port defined for dial-back security.
accesscode_entry	A list of one or more of the **accesscode** entries: **phone_no**, **in_pool_name**, **out_pool_name**, **job**.
phone_no	Specifies the dial-back phone number with the format: **phone_no** *phone_no* *phone_no* is the phone number to be called. If this optional parameter is not specified, the user is prompted for the dial-back phone number. System administrators are encouraged to specify this entry to avoid compromising system security. Any characters accepted by the modem can be used here. Notice that the escape character (\) must precede each special character (* , # @ ! ; =).

(continued on next page)

Table A-25. Entries for accesscode in the acp_userinfo File (continued)

Entry	Description
in_pool_name	Specifies the name of the inbound modem pool with the format: **in_pool_name** *pool_name* *pool_name* is the name of an inbound modem pool. For the dial-back request to be initiated, the designated port type must be defined for the inbound pool.
out_pool_name	Specifies the name of the outbound modem pool with the format: **out_pool_name** *pool_name* *pool_name* is the name of an outbound modem pool. For the dial-back request to be initiated, the designated port type must be defined for the outbound pool.
job	Defines a specific CLI command. The default is the CLI. Each **accesscode** can have up to one job record, using the format: **job** *command* [*argument...*]**end** *command* is a CLI command name, e.g., **rlogin**. *argument...* is an option list of command-specific arguments, e.g., to remotely log the user *Morse* into the host *amos*, the job entry is: `job rlogin amos -1 Morse end`

The following example illustrates accesscode entries in the **acp_userinfo** file. When logging in, the user *cobb* is prompted for a user name, password, and **accesscode**. If *cobb* enters the information at each prompt, the Annex determines whether or not *cobb's* access is via the inbound modem pool. If so, one of the following occurs:

- If cobb enters *access* at the **accesscode** prompt, the RA 6300 calls *cobb* back at the number *9–765–4321* and then logs *cobb* into the host *calvin*.
- If *cobb* enters *promptphone* at the **accesscode** prompt, the RA 6300 prompts for a phone number, drops the connection, and calls *cobb* back via the outbound modem pool. Then the Annex prompt is displayed.
- If *cobb* enters *direct* for the accesscode, the Annex prompt is displayed and no dial-back occurs.

```
user cobb
     at_passwd          nedry
     at_zone            bn-33net bn-55net end
     accesscode         access
           phone_no         9\,7654321
           in_pool_name     inbound
           out_pool_name    outbound
           job              rlogin calvin –1 cobb end
     end

     accesscode             direct
           in_pool_name     inbound
     end

     accesscode         promptphone
           in_pool_name     inbound
           out_pool_name    outbound
     end
end

pool inbound
     ports              asy@hobbes
     ports              asy@simon
end
pool outbound
     ports              asy@hobbes
     ports              asy@simon
end
```

clicmd

For a single user or for conditions that meet profile criteria, you can define one or more CLI commands and macros in the **acp_userinfo** file. These commands will be executed, in the order in which they are specified, if the profile criteria are met or the user name matches the userid supplied at login. If the RA 6300 detects an error in a command, **erpcd** stops sending commands, syslogs an error, and denies access to the user.

Table A-26 describes the **clicmd** entry. The syntax is:

clicmd *CLI_command* **end**
clicmd ... end

For **clicmd** to work, the **cli_security** parameter must be set to **Y**.

Table A-26. Arguments for the clicmd Entry in the acp_userinfo File

Argument	Description
CLI_command	Any user or superuser CLI command, or the name of a macro previously defined for the RA 6300. Only one command or macro is allowed per **clicmd** entry, although a **user...end** block can contain multiple **clicmd** entries. After the final command in a **user...end** block executes, the CLI session ends. To continue the session, use the **clicmd** with the ellipses (**...**) argument explained next. (For descriptions of the CLI commands, see *Using the CLI Commands* on page C-111.
...	Specifies that the CLI session should not end when the last **clicmd** entry for a given user has been executed. Subsequent commands in the same **user...end** block are ignored.

The **clicmd** entry is useful for configuring dedicated connections. In the following example, if user *kip* logs in at any time between 9:00 A.M. and 5:00 P.M. on the specified days, the RA 6300 executes the **ppp** command (after authenticating *kip* at the CLI level). The port to which *kip* is connected is thereby converted from CLI to PPP mode. When the PPP link goes down, *kip* is disconnected from the RA 6300.

```
user username=kip;time=”9:00am – 5:00pm Tuesday-Friday”
     clicmd ppp end
end
```

The next example does the same thing as the previous example, except that it does not disconnect *kip* when the PPP link terminates.

```
user username=kip;time=”9:00am – 5:00pm Wednesday-Friday”
     clicmd ppp end
     clicmd ... end
end
```

climask

For a single user or for conditions that meet profile criteria, you can define a CLI command mask in the **acp_userinfo** file that limits which CLI commands the user(s) can execute (see *Masking CLI Commands* on page 15-333). Table A-27 describes the entry for **climask** in the **acp_userinfo** file. The syntax for adding the CLI command mask to a user profile is:

climask *command_list* **end**

Table A-27. Entry for climask in the acp_userinfo File

Entry	Description
command_list	A list of user-level CLI commands, separated by spaces, that are ***not*** available to the user. Valid values are **bg**, **call**, **fg**, **hangup**, **help**, **hosts**, **jobs**, **kill**, **netstat**, **rlogin**, **stats**, **stty**, **telnet**, **who**, **lock**, **su**, **slip**, **connect**, **services**, **ppp**, **arap**, **ipx**, and **none** (the default). The list of restricted command names is sent to the RA 6300 and the user is prevented from executing those CLI commands. Do not specify the same command as both a **clicmd** and a **climask** in a given **acp_userinfo** entry (for more details on CLI commands, see *Using the CLI Commands* on page C-111).

The **climask** entry allows minimum uniqueness for command names. If you specify an ambiguous command name, **climask** generates a warning but cannot prevent the user from issuing the command.

The following is an example of **climask**:

```
user username=sam;time=”9:00am-10:30pm Friday-Monday”
     climask ppp arap end
end
```

If user *sam* logs into any RA 6300 between 9:00 A.M. and 10:30 P.M. on the specified days, he cannot issue the **ppp** or **arap** command. In all other situations, this particular **user...end** block is ignored. For example, if *sam* logs into an RA 6300 at 11:00 PM, the entry is ignored.

deny

For a single user or for conditions that meet profile criteria, you can deny access to the RA 6300 in the **acp_userinfo** file. If the profile criteria are met or the user name in the user entry matches the userid supplied at login, ACP refuses access to the RA 6300. Table A-28 describes the entry for **deny** in the **acp_userinfo** file. The syntax is:

deny

Table A-28. Entry for deny in the acp_userinfo File

Entry	Description
deny	A keyword indicating the user will be denied access to the RA 6300. If used, **deny** should be the only entry in the **user...end** block. A message is logged in the ACP log file indicating why access is being denied. For CLI users, a message is displayed.

The following is an example of using **deny** in the **acp_userinfo** file:

```
user username=liza
     deny
end
user group=eng;time=”9:00am -10:30pm Saturday-Sunday”
     clicmd ppp end
end
```

In this example, even if user *liza* is a member of the *eng* group, she is denied access, since **erpcd** finds the match with the userid first.

In the following example, no user is permitted to connect to any RA 6300 between 11:00 PM and 12:00 PM on any of the specified days:

```
user time=”11:00pm – 12:00pm Saturday-Wednesday”
     deny
end
```

filter

For a single user or for conditions that meet profile criteria, you can define one or more IP filters in the **acp_userinfo** file. These filters can apply to PPP and/or SLIP packets. Table A-29 describes the entry for **filter** in the **acp_userinfo** file. The syntax is:

filter *filter_definition* **end**

Filters are session-specific; they are dynamically applied to each internal port for the duration of a session based on the entries in the **acp_userinfo** file.

Table A-29. Entry for filter in the acp_userinfo File

Entry	Description
filter_definition	Defines a filter to apply to the port on which the user logs in. You can enter only one filter per line but multiple filters are allowed within one **user...end** block. Filters are applied in the order in which they are specified. Unlike the **filter** command, a filter specification in **acp_userinfo** does not start with the word **add** (since it is assumed that you are adding a filter) and does not contain the name of the login interface (since that is known).

You can also restrict the transmission and reception of SLIP and IP over PPP packets by using the **acp_restrict** file. Using **acp_restrict** for this purpose can be easier than using **acp_userinfo** because you do not have to enter actual filters in **acp_restrict**. Instead, you enter user-friendly statements from which filters are created for you.

Any filters you enter in **acp_userinfo** or arrange to have generated by **acp_restrict** will be combined with, and interpreted according to the algorithm used for, filters created by the superuser **filter** command. For more information, see *Filtering* on page A-143.

The following example creates a filter that discards any IP packets destined for address *132.245.4.33* – if transmission of such packets is attempted on the port from which user *sam* logs in.

```
user username=sam
     filter output include dst_address 132.245.4.33 discard end
end
```

Like all other **acp_userinfo** entries (except **deny**), the **filter** entry can be accompanied by other entries within the same **user...end** block. In the following example, not only is the above filter created, but a pre-defined macro named *special_setup* and the CLI command **ppp** are also executed for user *sam*.

```
user username=sam
     clicmd special_setup end
     filter output include dst_address 132.245.4.33 discard end
     clicmd ppp end
end
```

route

For a single user or for conditions that meet profile criteria, you can define one or more IP routes in the **acp_userinfo** file. You can enter only one route per line, but multiple routes are allowed within one **user...end** block.

Routes in **acp_userinfo** are entered into the routing table when their interfaces become active, but they are not entered into the route cache. You cannot use a **route** entry in **acp_userinfo** to define a default route. The syntax for the **route** entry is:

route [**–h**] *dest mask gateway* [*metric*] **end**

Table A-30. Argument for route Entry in the acp_userinfo File

Entry	Description
–h	Defines the route as hardwired.
dest	Specifies the destination address of the route.
mask	Specifies the subnet mask for the destination address. You can enter the mask in dotted decimal notation, e.g., 255.255.255.0, or you can specify the mask by appending /*n* to the destination address, where *n* is the number of 1 bits in the mask, from left to right. For example, appending /24 specifies 255.255.255.0 as the subnet mask.
gateway	Specifies the IP address of the gateway (router) that is the next hop for the route. If you specify an asterisk (*) for gateway, the RA 6300 uses the port's remote address as the gateway.
metric	Specifies the number of hops to the destination. Values range from **1** through **15**; the default is **1**.

Typically, a **route** entry in **acp_userinfo** is used when a router attached to a small network dials into the RA 6300 but does not want to incur the overhead of running a routing protocol itself. Consider the configuration in Figure A-15.

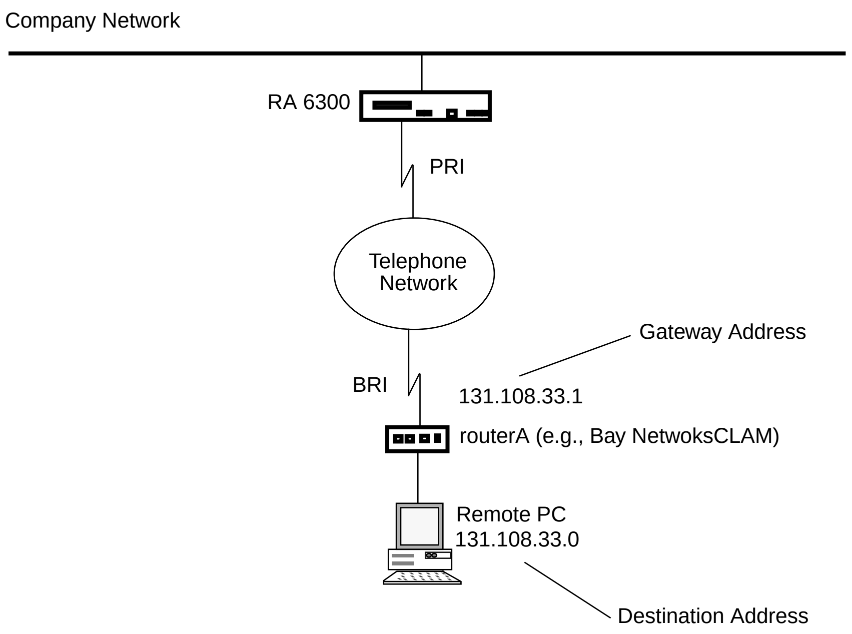


Figure A-15. Sample Configuration for a route Entry in acp_userinfo

Given the configuration in Figure A-15, the following example defines a route on the *RA 6300* that will be used for routerA (e.g., Bay Networks CLAM). This route allows packets to be sent back and forth between the company network and the remote PC with the IP address *131.108.33.0*. The destination address is *131.108.3.0*, using a subnet mask of *255.255.255.0*. The gateway address is *131.254.33.1*, and the metric for the route is 1 (the default).

```
user username=routerA;annex=RA6300
     route 131.108.33.0/24 131.254.33.1 1 end
end
```

at_zone

For a single user or for conditions that meet profile criteria, you can define AppleTalk zone list entries in the **acp_userinfo** file. This zone list consists of zone names (for more details, see *AppleTalk* on page A-183). Table A-31 lists the entry for **at_zone** in the **acp_userinfo** file. The syntax is:
at_zone *zone*... **end**

Table A-31. Entry for at_zone in the acp_userinfo File

Entry	Description
zone	A list of one or more ASCII character strings. You can have any number of zones specified in a zone list, subject to the following constraints: • A zone identifier cannot contain non-printable characters. • An individual zone identifier cannot exceed 32 characters in length. • The combined length of the entire zone list cannot exceed 524-*n* characters, where *n* is the number of zones in the list. • The reserved keyword **end** cannot appear as a zone argument. • A string containing a space must be enclosed in double quotation marks.

The following example illustrates **at_zone** entries in the **acp_userinfo** file. When logging in using ARA, user *cobb* is assigned to zones *bn-33net* and *bn-55net*.

```
user username=cobb
     at_zone bn-33net bn-55net end
end
```

The next example shows an **at_zone** entry that uses profile criteria. When logging in via ARA between the hours of 8:00 A.M. and 6:00 P.M, user *hobbes* is assigned to zones bn-11net and bn22-net.

```
user username=hobbes;time=“8:00am-6:00pm Sunday-Wednesday”
     at_zone bn-11net bn-22net end
end
```

at_connect_time

The **acp_userinfo** file can have an ARA connect timer defined; **at_connect_time** defines the maximum amount of time, in minutes, that an ARA connection can remain open. You can specify **at_connect_time** for a single user or for conditions that meet profile criteria. Table A-32 defines the argument for **at_connect_time** entries in the **acp_userinfo** file. The syntax is:

at_connect_time *time_value*

Table A-32. Entries for at_connect_time in the acp_userinfo File

Entry	Description
time_value	The format for this argument is <*minutes*>. For example: **at_connect_time** *120*

at_nve_filter

NVE filtering controls a remote access Apple user's view of network resources: when using *Chooser* to select resources, only the resource set defined for the user by the administrator will be visible. The administrator can specify the NVE filter on a per-user basis or for conditions that meet profile criteria. This feature complements the existing zone list, described above, by offering a higher level of control.

The administrator uses the **nve_filter** entry in the **acp_userinfo** file to specify a finite list of filter. Only one nve_filter entry per user or per profile criteria specification is permitted. The entry uses the format:

at_nve_filter [**include|exclude**] *tuple,tuple tuple...* **end**

Table A-33. Entries for at_nve_filter in the acp_userinfo File

Entry	Description
include| **exclude**	The **include** or **exclude** qualifier controls how filters are used: **include** filters allow only matching answers; **exclude** filters discard matching answers and allow non-matching answers. There is a 10-filters-per-user limit. The default is **include**.
tuple	A three-part string that identifies all network resources. The three parts of a *tuple* are: object, type, and zone. The format of a *tuple* is: **object:type@zone** with asterisks as wild cards. Any *, @, or : used as an NVE character within a *tuple* must be preceded by the Escape (\) character. Characters in a *tuple* are case-insensitive. Each field of an entity can be 32 characters long.

Following are sample **acp_userinfo** entries, including **nve_filter** information, for two users. User *frick* is allowed access only to the resources of her office Macintosh named *Frick CPU*. User *frack* cannot access *frick*’s machine, nor is she allowed access to any sales resources.

```
user username=frick
     at_passwd klot
     at_nve_filter include Frick\CPU:*@eng end
end

user username=frack
     at_passwd curly
     at_nve_filter exclude Frick*:*@ *:*@sales end
end
```

Like all other **acp_userinfo** entries, **nve_filter** information is syntax-checked by erpcd. Any errors cause the entire filter to be discarded, and an error message is generated.

This method of limiting NBP traffic is not secure, and can be circumvented by a person willing to write code to probe the network without using NBP. Also, this feature has no local RA 6300 security equivalent.

at_passwd

Each registered AppleTalk user (as opposed to a guest) must have a password defined in the **acp_userinfo** file. The definition can be for a single user or for conditions that meet profile criteria (e.g., membership in a group). Table A-34 defines the argument for the **at_passwd** entry in the **acp_userinfo** file. The syntax is:

at_passwd *string*

Table A-34. Entry for at_passwd in the acp_userinfo File

Entry	Description
string	A string of up to nine alphanumeric characters (the unencrypted password). Punctuation marks are permitted; spaces and hex values must be preceded by a backslash (\).

The following example illustrates an **at_passwd** entry in the **acp_userinfo** file:

```
#Set up the user entry
#

user username=cobb
     at_passwd ned\ ry
end
```

A guest entry in the **acp_userinfo** file looks like this:

```
#Set up a guest user entry that allows guests to connect
#for 1 hr.and hides our file servers

user username=<Guest>

     at_connect_time 60:00
     at_nve_filter exclude

        *:AFPServer@*

     end

end
```

The *Guest* entry is case-sensitive. If it is entered incorrectly, guests can log in with no restrictions because the **at_guest** parameter for this port is set to **Y**.

chap_secret

A secret token that enables CHAP authentication for PPP is defined in **chap_secret** entries in the **acp_userinfo** file. The token can be defined for a single user or for conditions that meet profile criteria (e.g., membership in a group). Table A-35 defines the argument for the **chap_secret** entry. The syntax is:

chap_secret *secret_token*

Table A-35. Entry for chap_secret in the acp_userinfo File

Entry	Description
secret_token	A string from 1 to 32 bytes long; 16 bytes is recommended due to the operation of the MD5 encryption algorithm.

The following example illustrates a **chap_secret** entry in the **acp_userinfo** file:

```
user username=smith
     chap_secret achapsecrettoken
end
```

For more details on CHAP and secret tokens, see *Challenge-Handshake Protocol (CHAP)* on page 15-295.

Limiting Access to Hosts via acp_restrict

The ERPCD can restrict any CLI (i.e., **telnet** and **rlogin**), SLIP, or PPP request for IP access to a specific host or host-port combination. This security mechanism uses a host-resident file that lists the hosts and host ports to which access is restricted and specifies the RA 6300 or the profile criteria to which the restrictions apply. By default, there are no host or host-port restrictions.

Host access security for CLI ports is enabled by setting the port parameter **connect_security** to **Y**.

Hosts or ports not listed in **acp_restrict** are considered unrestricted.

When a user issues a connection command or a SLIP or PPP link becomes active, the RA 6300, using **erpcd**, checks a restrict file for permission to connect to that host. **erpcd** expects the restrict file to be **acp_restrict** (located in the installation directory), which is an ASCII file that you create with any text editor. Table A-36 describes the arguments in each entry. The entry format is:

annex | *profile_criteria*: *restricted host* [[*ports*]] [,*restricted host* [[*ports*]], ...]
annex | *profile_criteria*~ *unrestricted host* [[*ports*]] [, *unrestricted host* [[*ports*]],...]

Table A-36. Arguments in the acp_restrict File Entries

Argument	Description
annex	The name or IP address of the RA 6300 initiating the access. This argument is supported for backward compatibility with Release 10.1 and earlier releases but is treated as if it were the profile criterion **annex=***annex* (*profile_criteria* are described next).
profile_criteria	One or more *keyword*=*value* pairs, separated by semicolons (;), specifying the conditions under which the specified hosts will be restricted or unrestricted. For information on entering *profile_criteria*, see Profile Criteria Syntax on page 15-239.
: (colon)	Indicates that the hosts listed in the same entry are restricted. White space may follow, but not precede, the colon.
~ (tilde)	Indicates that the hosts are unrestricted. White space may follow, but not precede, the tilde. For PPP and SLIP connections, hosts specified as unrestricted imply that all other hosts are restricted. (For a CLI connection, hosts specified as unrestricted have no implications for other hosts.)
restricted host	The name or IP address of a restricted host (including RA 6300s). The list of restricted hosts is separated by commas; no white space is allowed. An asterisk (*) can be used as a wild card in place of a host name or the host part of an IP address.

(continued on next page)

Table A-36. Arguments in the acp_restrict File Entries (continued)

Argument	Description
unrestricted host	The name or IP address of an unrestricted host (including RA 6300s). The list of unrestricted hosts is separated by commas; no white space is allowed. An asterisk (*) can be used as a wild card in place of a host name or the host part of an IP address.
[*ports*]	One or more TCP or UDP ports on *restricted host* or *unrestricted host*. To specify multiple ports, separate them with commas or specify them as a range separated by a hyphen (–). Enclose the port(s) in square brackets ([ ]). White space and wild cards are not allowed. The default is any TCP or UDP port.

Following are two restricted-host entries:

```
annex01: hosta,hostb,hostf,132.245.6.23
annex02: hostc,132.245.6.15,hostf,132.245.6.23,\
                         hosth,annex01
```

In the previous example, the first entry prevents SLIP, PPP, and CLI connections from *annex01* to any port on *hosta*, *hostb*, *hostf*, or the host at IP address 132.245.6.23. The second entry prevents SLIP, PPP, and CLI connections from *annex02* to any port on *hostc*, *hostf*, *hosth*, the host at IP address 132.245.6.15, the host at IP address 132.245.6.23, and *annex01*.

In the next example, which shows the use of profile criteria, user *carl* is blocked from using **telnet** or **rlogin** to access hosts *atlas* and *steam*:

```
username=carl;protocol=cli:atlas,steam
```

For profile criteria entries in which the only protocol specified is **cli**, as in the previous example, **erpcd** searches the file in sequential order and uses only the first entry whose profile criteria are met. For these types of entries, order of placement in the file is important. If permission is granted to a CLI connection request, the user follows the normal login procedure. If the request is denied, the message *Permission denied* is displayed and the session (job) is aborted.

For profile criteria specifications that explicitly specify **slip** or **ppp** (or implicitly specify them, by not specifying *any* protocol), filters are automatically generated to restrict SLIP and/or PPP connections if either protocol becomes active. Consider the following example:

```
username=*;protocol=slip: finance
```

In this example, all SLIP users on all RA 6300s are denied access to host *finance* but are allowed access to all other hosts and host ports.

Given an address of 132.245.11.4 for host *finance*, the filters generated to effect these restrictions are:

```
in include address_pair 132.245.11.4 * discard
out include address_pair 132.245.11.4 * discard
```

In the next example, the members of the group *mail_only* who connect using the PPP or SLIP protocol (as opposed to the CLI) may access the SMTP port (25) on host *mailhub* and the DNS server port (53) on the host *dns_srv*, but they cannot access anything else.

```
group=mail_only;protocol=slip~ mailhub[25], dns_srv[53]
group=mail_only;protocol=ppp~ mailhub[25], dns_srv[53]
```

To put these restrictions into effect, the RA 6300 would generate the following four filters, in which *132.245.33.1* is the address of *mailhub* and *132.245.33.2* is the address of *dns_srv*.

```
in exclude address_pair 132.245.33.1 * port_pair 25 * discard
out exclude address_pair 132.245.33.1 * port_pair 25 * discard
in exclude address_pair 132.245.33.2 * port_pair 53 * discard
out exclude address_pair 132.245.33.2 * port_pair 53 * discard
```

Filtering Restrictions

IP filtering can handle the following two cases:

- One or more hosts cannot be reached and all other hosts can.
- One or more hosts can be reached and all other hosts cannot.

However, IP filtering cannot handle the next two cases:

- A subset (e.g., a subnet or subnet group) of hosts can be reached, except for a few hosts in the subset, and all other hosts cannot be reached.
- A subset of hosts cannot be reached, except for a few hosts in the subset, and all other hosts can be reached.

For example, you cannot use **acp_restrict** to allow a user named *martha* to access all hosts on her home network (132.245.0.0), except for the finance machine at IP address 132.245.77.1, and also deny her access to hosts outside the 132.245.0.0 network. The **acp_restrict** entries for this would be:

```
user martha: 132.245.77.1
user martha~ 132.245.*
user martha: *
```

If such an entry is found, a syslog message is generated and the user is denied access.

In addition, **acp_restrict** cannot create filters from host names containing wild cards, e,g., annex*.

Finally, filters apply to IP packets only; IPX and AppleTalk packets cannot be filtered.

Using include Files in the acp_userinfo File

To reduce the task of repeating several **job** and **climask** lines in the **acp_userinfo** file, you can create an **include** file. Nested **include** files are not allowed; the only commands allowed in the **include** file are **job** and **climask**. The syntax is:

include *filename*

Table A-37. Argument for the include File

Argument	Description
filename	The name of a file located in the same directory as **acp_userinfo**.

A log message is written to the ACP log file if the **acp_userinfo** file references an **include** file that could not be opened (*Host-based Security Logging* on page B-33 provides sample ACP log file entries).

Dynamic Allocation of Network Addresses

Introduction to DHCP

The *Dynamic Host Configuration Protocol* (DHCP) enables dynamic IP addressing for remote access clients establishing a PPP connection to a Remote Annex. This eliminates the need to assign an IP address manually (and the subsequent need to reconfigure and reboot) each time that a host is added or moved to a new subnet location.

The Remote Annex acts as a DHCP “client-by-proxy”, requesting and accepting a dynamic IP address from a DHCP server on behalf of a dial-in client. The term *DHCP client* always refers to a Remote Annex acting as a DHCP client-by-proxy.

DHCP is enabled by setting the Remote Annex parameter **address_origin** (which has replaced the previously existing parameter, **dialup_addresses**) to **dhcp**, or by setting the Remote Address field in the **acp_dialup** file to **dhcp** (see *Creating the acp_dialup File* on page 15-276).

When DHCP is enabled, a DHCP client seeks to discover a DHCP server by requesting an IP address first from the DHCP server specified by **pref_dhcp1_addr**, then, if that server does not respond, from the DHCP server specified by **pref_dhcp2_addr**. The DHCP server checks the subnet of the requesting DHCP client, allocates an IP address from a pool of IP addresses made available for that subnet, and offers it to the requesting DHCP client. The DHCP client uses the allocated IP address for an interval of time called a “lease,” which is maintained for as long as the remote client connection is active, or until the DHCP client terminates the serial connection. When the lease expires, the DHCP client returns the address to the pool of dynamic addresses maintained by the DHCP server. The DHCP server can then reuse that IP address, allocating it to another DHCP client which requests an IP address.

Non-supported Features of DHCP

Some aspects of DHCP are not relevant to its use on a Remote Annex, specifically:

- A Remote Annex does not implement the BOOTP Relay function. A Remote Annex will not support host-based DHCP requests, e.g., a DHCP client operating on a PC, expecting the Remote Annex to relay host-generated DHCP protocol messages to the DHCP server.

- The DHCP client cannot be used to configure the dial-in host.
- DHCP clients do not use automatic address allocation, which assigns permanent IP addresses, nor do they retrieve statically configured IP addresses from the DHCP server.

Cautions

- If the DHCP client is invoked, but is unable to obtain an address from a DHCP server, it syslogs the condition “Client did not receive a DHCPOFFER”; the DHCP client cannot supply an address to the IPCP, and the remote connection is terminated.
- It is possible that the DHCP client will be unable to discover a DHCP server and obtain an IP address from it before the PPP connection establishment times out and terminates.

Creating the acp_dialup File

The **acp_dialup** file resides in the RA 6300 install directory. Any ACP dial-up address request that comes from the RA 6300 includes the RA 6300 address and port number, and an associated user name, which are used as keys in this file. Once the keys are matched, the corresponding dial-up addresses are returned to the caller on the RA 6300. If no match is found, the RA 6300 uses the port’s **remote_address** and **local_address** (for more details, see *Determining Dial-up Addresses using the acp_dialup File* on page 15-278).

The **acp_dialup** file contains the following fields: *User*, *RA 6300*, *Remote Address*, and *Local Address* (optional). If a local address is not specified, the local address sent back to the RA 6300 is that RA 6300's IP address. For example:

```
#User   Annex                      Remote address Local address
smith   100.30.200.39              100.30.200.45  100.30.200.46
green   *                          100.30.200.48
harris  mars                       100.30.200.55  100.30.200.40
frank   *                          dhcp
```

You can specify the RA 6300 by name, IP address, or wild card (*); the wild card means that any incoming address request with that user name will match. The file format allows one entry per line; the RA 6300 ignores any data following the comment character (#); a newline character terminates an entry.

In the previous example:

- User *smith* can make a dial-up address request from RA 6300 100.30.200.39. The remote address is 100.30.200.45; the local address is 100.30.200.46.
- User *green* can make a dial-up address request from any RA 6300. The remote address is 100.30.200.48; the local address is the address of the RA 6300 from which the request originates.
- User *harris* can make a dial-up address request from the RA 6300 named *mars*. The remote address is 100.30.200.55; the local address 100.30.200.40.
- User *frank* will obtain a remote address from a DHCP server.

Determining Dial-up Addresses using the acp_dialup File

When the port parameter **address_origin** is set to **acp**, the *local* and *remote* field settings in the **acp_dialup** file supersede the values set in the **local_address** and **remote_address** port parameters.

When **address_origin** is set to **acp**, the RA 6300 searches for the remote client's user name in the **acp_dialup** file. RA 6300 behavior at this point depends on whether or not the RA 6300 finds a matching user name in **acp_dialup**:

- If the RA 6300 does find a matching user name in **acp_dialup**, it looks at the corresponding *local* and *remote* address fields.
  - If both of these addresses are set in the **acp_dialup** file, the RA 6300 forces the use of these values over the settings in the **local_address** and **remote_address** port parameters.
  - If the *local* address field is not set, but the *remote* address field is set, the RA 6300 forces the use of the *remote* address field setting for the remote address and forces the local address setting to be the RA 6300's IP address.
  - If the *remote* address field is set to **dhcp**, a remote address will be allocated dynamically by a DHCP server (see *Dynamic Allocation of Network Addresses* on page 15-274 for a complete description).
- If the RA 6300 does not find a matching user name in the **acp_dialup** file, it looks at the **local_address** and **remote_address** port parameters.
  - If the **local_address** and **remote_address** parameters are set, the RA 6300 uses these values for the local address and remote address.

- If the **local_address** and **remote_address** parameters are not set, the RA 6300 negotiates for both the local and remote address values with the remote PPP client. (If these conditions are true for a remote SLIP client, the connection is denied.)
- If the **local_address** parameter is set but the **remote_address** parameter is not set, the RA 6300 forces the use of the value in the **local_address** parameter and negotiates for the remote address value with the remote PPP client. (If these conditions are true for a remote SLIP client, the connection is denied.)

Enhancing Password Security

The following sections describe how to configure the RA 6300 to record password histories and blacklist users. It also explains how to view and manage the database in which password histories and blacklisting information is kept.

Overview of Password History and Aging

You can configure ACP to save the encrypted form of the previous passwords a user has set. This applies to passwords set by using the **ch_passwd** utility on the security host or by responding to an RA 6300 prompt when a password expires. These passwords are stored in the UNIX database **acp_dbm** on the security host, where they are keyed on user names. If a user tries to reset his or her password to one of the stored values, ACP will reject it and display an error message.

Benefits of Password Histories

The password history mechanism helps protect against off-line, “dictionary” attacks. In this kind of attack, a user obtains the encrypted **acp_passwd** (or /**etc/passwd**) file. The user then tries to crack the passwords by taking a dictionary of words, encrypting the words (using *salted* DES encryption) and comparing them to the encrypted passwords.

Benefits of Password Aging

The longer a password is in effect, the more time an attacker has to crack its encryption. Consequently, the password history feature is most effective when used in conjunction with password aging. If password aging is enabled:

- The user *must* change passwords when a predefined amount of time has elapsed. If the user never changes passwords, there is no password history to record.
- The user *cannot* change passwords until the predefined amount of time has elapsed. This prevents potential intruders from changing passwords in rapid succession in an attempt to cycle the old passwords out of the password history and use them again.

Password aging is enabled through the use of a **shadow** file in conjunction with a **passwd** file. By default, **erpcd** uses the **acp_passwd** file alone, so password aging is initially disabled. When only the **passwd** file is used (a Berkeley standard), that file contains both the user names (UIDs) and the encrypted passwords. The **passwd/shadow** form (used with UNIX System-V) contains an *x* in place of a password in the **passwd** file and saves the encrypted passwords in a separate file called **shadow**.

If your UNIX is based on System V and you want to use the password history feature, choose the **passwd/shadow** scheme. Use the *convert* program, located in the **erpcd** directory, to change the integrated **passwd** form to the **passwd/shadow** form (and vice-versa).

If your UNIX is based on a Berkeley BSD system, password history is disabled by default. To enable it, change the value of the *STORED_PASS #define* statement in **acp_policy.h**, as described in the following section.

Enabling and Configuring Password Histories

To turn on the password history feature and (optionally) enable aging via shadow files:

1. **Use** na **or** admin **to make sure that the** enable_security **parameter is set to** Y **for the RA 6300(s) you are configuring.**
2. **Use** na **or** admin **to make sure that you have defined a security host for the RA 6300(s) you are configuring. (See *Configuring the Security Server* on page 15-230.)**
3. **Log into the security host as** root**.**
4. **Go (**cd**) to the installation directory (typically** /usr/annex**).**
5. cd **to the** src/erpcd **directory, which is within the installation directory.**
6. **In the** erpcd **directory, use a text editor to modify the** acp_policy.h **file.**

   - If you are using a shadow file, uncomment the following line in **acp_policy.h**:

     ```
     /* #define USESHADOW 1 */
     ```

     To uncomment the previous line, delete the slashes and asterisks at the beginning and end of the line, so that the line is as follows:

     ```
     #define USESHADOW 1
     ```

- Look for the following lines to define STORED_PASS, which are already uncommented. The lines define the number of passwords that will be stored to prevent them from being re-used. The variable is initialized to 6 for **passwd**/**shadow** files and 0 for **passwd** files alone. A value of 0 disables password history.

  ```
  #ifdef USESHADOW
  #define STORED_PASS 6
  #else
  #define STORED_PASS 0
  #endif
  ```

  If you are using a **shadow** file and want to change the number of passwords stored from 6 to some other value, do so. The maximum is 12.

  If you are using a **passwd** file alone and you want to enable password history, change the value of the second STORED_PASS from 0 to a number from 1 through 12.

  Specifying a non-zero value for either of the above STORED_PASS variables turns on the recording of password histories in **acp_dbm**.

- The final variable related to password history is MAX_STORED_PASS. It defines the absolute maximum number of login failures that can occur before a user is blacklisted. It is best not to change this variable, which is set to 12. If you must change it, follow the instructions in **acp_policy.h**.

7. **If you plan to use only the password history feature and not blacklisting as well, follow the instructions in Steps 8 through 11, below. If you are also using blacklisting, wait to do this until you have configured both features.**

8. **From the** /usr/annex/src **directory, recompile** erpcd**:**

   ```
   # cd /usr/annex/src
   # make install
   ```

   This automatically rebuilds **erpcd** and any other files that need to be recompiled because of the changes you have made. In addition, the following message is displayed:

   ```
   WARNING: If you have called “make install” yourself,
   then in directory /usr/annex you will have to copy
   erpcd.new to erpcd. Make sure the erpcd daemon is not
   running when that is done.

   If you used the installation script called “make
   install” then the copy will be done for you.
   ```

9. **If** erpcd **is running on the host and the host is running Berkeley BSD UNIX, kill the existing** erpcd **process as follows (your process number will vary):**

   ```
   # ps –ax | grep erpcd
   25493 ? IW 0:00 ./erpcd
   25797 p1 S 0:00 grep erpcd
   # kill -9 25493
   ```

   To kill the process on a System V host, substitute the following for the first line above:

   ```
   # ps –ef | grep erpcd
   ```

10. cd **to directory /usr/annex and copy** erpcd.new **to** erpcd**:**

    ```
    # cd /usr/annex
    # cp erpcd.new erpcd
    ```

11. **Now, restart** erpcd**.**

    ```
    # ./erpcd
    ```

After you complete Steps 1 through 11, the **acp_dbm** database is created automatically the first time a user changes a password via the **ch_passwd** utility. To list the users for which password histories exist, go to the security host’s install directory (default is **/usr/annex**) and issue the **acp_dbm -l** command:

```
# cd /usr/annex
# acp_dbm -l
List of users currently present in the acp_dbm:
                     hobbes
                     fritz

#
```

In the previous example, password histories have been saved for users **hobbes** and **fritz**.

Overview of Blacklisting

A user account is considered under attack, and therefore blacklisted, when either (or both) of the following occurs:

- A configurable number (default is 5) of consecutive failed login attempts is exceeded. In other words, if you use the default, a user is blacklisted on the sixth consecutive failed login attempt.
- A configurable number (default is 10) of non-consecutive failed login attempts is exceeded within a configurable period of time (default is 26 weeks). If you use the defaults, a user is blacklisted when the eleventh failed login attempt occurs within a period of 26 weeks.

Blacklisting enhances security by limiting the number of passwords an on-line attacker can try before the user account is automatically disabled. At this point, no one can log in with the blacklisted user name, even if someone enters the “correct” password. However, the failed login message is the same before and after blacklisting, so the user does not know that the account has been disabled.

The system administrator is informed when blacklisting occurs. First, a record is created in the ACP log file indicating that the userid has been blacklisted. This record remains unless and until you delete it manually. Second, when you invoke the **acp_dbm** utility, it immediately displays a warning identifying any blacklisted users. See *Viewing and Managing the acp_dbm Database* on page 15-288.

The data necessary for blacklisting is kept in the **acp_dbm** database, keyed on the user name. If password history and blacklisting are configured, this database is created automatically the first time a user changes passwords or attempts to login and fails. The absence of an **acp_dbm** database indicates that no password histories exist and no failed login attempts have occurred.

Blacklisting makes the RA 6300 susceptible to denial-of-service attacks. To disable a user account, a saboteur need only make a few failed login attempts. In the extreme case, a saboteur who obtains a list of employee user names could create a shell script that would automatically disable all user login capabilities.

Configuring Blacklisting

You can configure blacklisting in one of two ways:

- By editing *#define* statements in the **acp_policy.h** file.
- By issuing the **erpcd** command with the **–b**, **–x**, and **–g** options.

The **erpcd** syntax is:

erpcd [**–b***max_con*] [[**–x***max_total*] [**–g***period*]]

Do not enter any space between an option (e.g., **–b**) and the value you specify with it (e.g., *max_con*).

The **erpcd** options override the **acp_policy.h** variables. Table A-38 describes the options and their **acp_policy.h** equivalents.

For information on how to edit and rebuild (**erpcd** and the other files that have changed) in the **acp_policy.h** file to put the modifications into effect. See Steps 6 through 11 in *Enabling and Configuring Password Histories* on page 15-281.

Once you have configured and activated blacklisting, **erpcd** automatically creates the **acp_dbm** database the first time a user makes an unsuccessful login attempt. To monitor the blacklist status of one or more users, go to the directory (on the security host) that contains **erpcd** and use the **acp_dbm** utility (see *Viewing and Managing the acp_dbm Database* on page 15-288).

Table A-38. erpcd Options and acp_policy.h Variables

erpcd Option	Equivalent acp_policy.h Variable	Description
–b*max_con*	MAX_BL_CON	The number of consecutive login failures a user is permitted before being blacklisted. Valid values are 0-8. A value of 0 enables blacklisting upon any login failure (not recommended). The default, as pre-set via MAX_BL_CON, is 5. If MAX_BL_CON is undefined and you do not specify **–b***max_con*, ACP never blacklists based on consecutive login failures.
–x*max_total*	MAX_BL_ NONCON	The number of non-consecutive login failures a user is permitted before being blacklisted. Valid values are 0-20. A value of 0 enables blacklisting upon any login failure (not recommended). The default, as pre-set by MAX_BL_NONCON, is 10. If MAX_BL_NONCON is undefined and you do not specify **–x***max_total*, ACP never blacklists based on consecutive login failures.
–g*period*	MAX_BL_ PERIOD	The time period, in weeks, over which *max_total* is applied. Login failures that occurred more than this number of weeks ago do not count toward blacklisting. Valid values are 0-52. The default, as pre-set via MAX_BL_PERIOD, is 26. If MAX_BL_PERIOD is undefined or is set to 0, MAX_BL_NONCON is effectively disabled.

Viewing and Managing the acp_dbm Database

The **acp_dbm** utility lets you manage and display information about password histories and blacklisting from the **acp_dbm** database. To use this utility, you must log in with a userid of **root** or have superuser privileges. If neither is the case, **acp_dbm** immediately exits on invocation and displays the message:

```
You must have root privilege to run acp_dbm.
```

Execute the **acp_dbm** utility from the directory containing **erpcd**. Upon execution, **acp_dbm** immediately sends a warning message to standard output for each user on the blacklist. The message format is:

```
Warning: Annex user userid may be under attack; all logins
for this account have been disabled.
```

In this message, *userid* is the user name for the account that has been blacklisted.

The syntax for the **acp_dbm** utility is:

acp_dbm [**–s** *username*] [**–c** *username*] [**–d** *username*] [**–l**]

Table A-39 explains the options.

Table A-39. Options for the acp_dbm Utility

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>–s <i>username</i></td><td>Sends information about <i>username</i> from the <b>acp_dbm</b> database to standard output. The output (after the initial warning message) shows the user name, the total number and type of failures, and the date and time of each failure. The following is an example:
<pre>
User name: hobbes
Total number of consecutive failed login attempts: 2
     Login failure on Tue Dec 12 12:49:49 1995
     Login failure on Mon Dec 11 11:25:10 1995
</pre></td></tr>
<tr><td>–c <i>username</i></td><td>Clears <i>username</i> from the blacklist and deletes all records of login failures for <i>username</i>. Does not clear the password history or any other information about <i>username</i> in the <b>acp_dbm</b> database. Before using this option, investigate the account thoroughly so that you are confident it is not under attack.</td></tr>
<tr><td>–d <i>username</i></td><td>Deletes the user record from the <b>acp_dbm</b> database. Use this option, rather than <b>–c</b>, to delete the <b>acp_dbm</b> user account entirely. This option does not delete references to <i>username</i> in any other ACP files, such as <b>acp_userinfo</b> and <b>acp_passwd</b>. You must explicitly remove the user name from these files to delete the user completely.</td></tr>
<tr><td>–l</td><td>Lists all the user names contained in <b>acp_dbm</b>, including those with password histories.</td></tr>
</table>

Deleting the acp_dbm Database

The only way to delete the **acp_dbm** database is via the UNIX **rm** command.

Error Handling for Password Histories and Blacklisting

The following error conditions can occur:

- If **erpcd** cannot read or write to the **acp_dbm** database or detects incorrect protection, the event is syslogged at level LOG_CRIT and all users are denied access until **erpcd** can read and write to **acp_dbm**.

  If the wrong protection is detected, the syslogged message is:

  ```
  Security problem: Wrong protection (not 600) on acp_dbm database.
  ```

  If **erpcd** cannot read or write to **acp_dbm**, the message is:

  ```
  Cannot [read from | write to] acp_dbm database.
  ```

- If the **acp_dbm** utility fails to read or write the **acp_dbm** database, it generates the following message:

  ```
  acp_dbm: Error [reading from | writing to] acp_dbm database.
  ```

  If the utility detects the wrong protection, it generates the following message:

  ```
  acp_dbm: Wrong protection (not 600) on acp_dbm database.
  ```

- If the **ch_passwd** utility fails to read or write the **acp_dbm** database, **ch_passwd** generates the message:

  ```
  ch_passwd: Error [reading from | writing to] acp_dbm database.
  Notify System Administrator. Password change cancelled.
  ```

  If **ch_passwd** detects the wrong protection, it generates the message:

  ```
  ch_passwd: Wrong protection (not 600) on acp_dbm database.
  Notify System Administrator. Password change cancelled.
  ```

Using AppleTalk Security

The RA 6300 implementation of ARA provides three areas of security:

- ARA security.
- Zone security.
- NVE filtering.
- Logging.

ARA Security

The basic ARA security features are:

- **Username and password authentication**

  The RA 6300 authenticates the client using Apple’s DES encryption algorithm. To define a user name and password for a registered (as opposed to guest) user, see *Creating the acp_userinfo File* on page 15-249.
- **Guest access**

  The RA 6300 allows anonymous access to the network. Restrictions can be applied to *guests* by setting up an ACP *guest* profile with limitations. For more details, see *at_zone* on page 15-263.
- **Connection timer**

  The connection timer is stored in the **acp_userinfo** file. For more details, see *Creating the acp_userinfo File* on page 15-249.

Zone Security

Every user can have a zone list assigned via remote ACP. If a list is not available via ACP, the RA 6300 provides all the zones it has learned from the network. If local security is used, use the per RA 6300 parameter **default_zone_list.** For more details, see *at_zone* on page 15-263.

NVE Filtering

NVE filtering controls a remote access Apple user's view of network resources: when using *Chooser* to select resources, only the resource set defined for the user by the administrator will be visible. The administrator can specify the NVE filter on a per-user basis. This feature complements the existing zone list, described above, by offering a higher level of control.

The **nve_filter** entry in the **acp_userinfo** file specifies a list of filters on a per-user basis. For detailed information on creating **nve_filter** entries, see *at_nve_filter* on page 15-265.

This method of limiting NBP traffic is not secure, and can be circumvented by a person willing to write some code to probe the network without using NBP. Also, this feature has no local RA 6300 security equivalent.

Logging

The RA 6300 logs activity and errors from the ARA session. The log is accessed via remote ACP (for more details, see *Logging User and RA 6300 Events* on page B-33).

Using IPXCP Security

The Internet Packet Exchange Control Protocol (IPXCP) uses PPP security. For information on PPP security, see *Using PPP Security* below. For information on other aspects of IPXCP, see *Internetwork Packet Exchange (IPX) Protocol* on page A-153.

Windows ‘95 IPXCP clients must make sure that SPAP security is not enabled on their PCs. SPAP is a proprietary Microsoft security mechanism not available to other systems, such as the RA 6300.

Using PPP Security

The RA 6300 supports two authentication protocols for PPP:

- Password Authentication Protocol (PAP).
- Challenge-Handshake Protocol (CHAP).

Both of these protocols are run over the PPP link after the LCP negotiations are complete (for more details on using a PPP link, see *Point-to-Point Protocol (PPP)* on page A-87).

Password Authentication Protocol (PAP)

PAP is a two-way handshake in which an ID/password pair are exchanged in clear text. Each half of the connection can require security.

If one side of the link agrees to use PAP, after the LCP negotiations are complete, that side will send a user name/password combination to its peer. Upon receipt, the peer authenticates that combination.

When the RA 6300 requests PAP and the peer ACKs the request, the RA 6300 handles the incoming PAP user name/password combination as follows:

- If the **enable_security** and **slip_ppp_security** parameters are set to **Y**, the RA 6300 first tries to authenticate the user name/ password combination using ACP. ACP checks the regime file to determine the regime and password file to use (see *Configuring the acp_regime File* on page 15-245). If the ACP server is unavailable, the RA 6300 falls back to local security (i.e., it compares the remote end's user name/password against the global port parameters **user_name** and **port_password**).
- If the **enable_security** parameter is set to **Y** and the **slip_ppp_security** parameter is set to **N**, the RA 6300 uses local security (i.e., it compares the remote end's user name/password against the global port parameters **user_name** and **port_password**).
- If the user name/password combination is valid, the RA 6300 sends a *PAP Authenticate-ACK* message. If the combination is not valid, the RA 6300 sends a *PAP Authenticate-NAK* message.

When the RA 6300 agrees to PAP, it sends the PAP user name/password combination as follows:

- It uses the global port parameter **ppp_username_remote** as the user name.
- It uses the global port parameter **ppp_password_remote** as the password.
- If the user name/password combination is valid, the peer sends a *PAP Authenticate-ACK* message. If the combination is not valid, the peer sends a *PAP Authenticate-NAK* message.

Challenge-Handshake Protocol (CHAP)

CHAP is a three-way handshake that depends on a secret token. The secret token is known to both sides of the peer-to-peer link.

When the challenge is sent by the RA 6300, the peer responds with a one-way encrypted value. The authenticator then runs the same encryption on the challenge message using the peer's secret token. It then compares the result to the received value. If they match, the authenticator sends a *success* message; otherwise, it sends a *failure* message. Currently, the only encryption algorithm supported is MD5.

The secret token must be distributed to both sides of the link by an external mechanism.

ACP is used only when the RA 6300 is authenticating a peer.

CHAP does not use the **acp_regime** file.

The secret token is defined within an *entry* option called **chap_secret** in the **acp_userinfo** file (for more details, see *Creating the acp_userinfo File* on page 15-249 and *chap_secret* on page 15-268).

In the following example, user *smith*, when logging into an RA 6300 running CHAP, will have the secret token, *achapsecrettoken*, used in verifying the *response*. The mechanism of receiving a challenge, determining a secret based on the user, and sending the result back to the challenger, is analogous to the user name/password paradigm.

```
user username=smith
    chap_secret achapsecrettoken
end
```

If the **slip_ppp_security** parameter is set to **Y**, the RA 6300 sends the username, challenge message, and challenge response to ACP for authentication. The RA 6300 uses local security when ACP is unavailable and the **port_password** parameter is set; local security ignores the user name and checks the *response* against **port_password** using the **port_password** to encrypt the challenge message. If the **port_password** parameter is not set, the link fails.

Receiving a CHAP Challenge

When the RA 6300 receives a *challenge*, the challenge and the secret token (the **ppp_password_remote** parameter value) are used to generate a *response* message (the *name* field is set to the **ppp_username_remote** parameter value). The value in the *response* message is a result of running MD5 encryption on the secret token and the value in the *challenge* message. If the RA 6300 receives a *success* message, the link enters (or remains in) NCP negotiation; otherwise, the link is terminated.

The RA 6300 negotiates an authentication *challenge* from a peer only if the **ppp_password_remote** and **ppp_username_remote** parameters are set for this session.

CHAP does not use the **acp_regime** file.

Sending a CHAP Challenge

When the RA 6300 sends a *challenge* to the peer (remote node requesting a link), it includes the **chap_auth_name** parameter value as the *name* field and a randomly generated number as the *value* field.

If ACP is used, after receiving the peer's *response*, the RA 6300 passes the following items to the ACP server: chap username, challenge, and the peer's response (id and challenge response).

The ACP server combines the secret, challenge, and id to create an expected response. The ACP server then compares the response it created with the one it received from the RA 6300.

If the responses are identical, the ACP server sends a success code to the RA 6300. If not, it sends a failure code.

Upon receiving a success code, the RA 6300 allows the link to be established. When receiving a failure code, the RA 6300 prevents it from being established.

The Annex sends a *challenge* only if the **enable_security** and **slip_ppp_security** parameters are set to **Y**, the **ppp_security_protocol** parameter is set to **chap**, and CHAP is ACKed during LCP. If the RA 6300 is ACKed for CHAP, it will seek only one valid *response*.

The RA 6300 terminates a link if it cannot authenticate a *challenge*. If the RA 6300 does not receive a *response* to a *challenge* within the allotted time-out, it re-issues the *challenge* for the defined number of retries.

ACP logging for CHAP includes the standard PPP login and reject. It also logs whether or not a chap secret was found in the acp_userinfo file (for more details on security logging, as well as a sample log file, see *Host-based Security Logging* on page B-33).

Re-issuing a CHAP Challenge

By default, the RA 6300 sends a *challenge* only once, at the time the link is established. Optionally, you can configure the RA 6300 to re-issue a challenge at random intervals ranging from one second to the maximum number of seconds you specify. To do so, set the Annex security parameter **max_chap_chall_int** to a value between 1 and 65535 (approximately 18.2 hours). The following example sets the maximum interval to 3600 (two hours). The RA_6300 will send a *challenge* at random intervals between 1 second and two hours over the course of the connection.

```
admin: set annex max_chap_chall_int 3600
```

The **max_chap_chall_int** default is 0, which disables the re-issuing of challenges.

Using the PPP Security Parameters

There are a variety of settings one can choose when configuring the RA 6300 for PPP security. Table A-40 lists the possible combinations of PPP security parameter settings and their effect on RA 6300 activity.

The following two statements are true for all cases listed in Table A-40. First, if a remote side of a link demands PAP, the RA 6300 uses **ppp_username_remote** and **ppp_password_remote** for the username and password. Second, if **ppp_username_remote** and **ppp_password_remote** are not set, the connection fails.

Table A-40. PPP Security Parameters and their Effect on RA 6300 Activity

If:	Then:

enable_security = N
ppp_security_protocol = n/a
slip_ppp_security = n/a
	Request no PPP security incoming.
	Do not log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = none
slip_ppp_security = Y
	Request no PPP security incoming.
	Log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = none
slip_ppp_security = N
	Request no PPP security incoming.
	Do not log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = pap
slip_ppp_security = Y
	Use ACP for incoming user name and password. Log accesses in the ACP log file.

(continued on next page)

Table A-40. PPP Security Parameters and their Effect on RA 6300 Activity (continued)

If:	Then:

enable_security = Y
ppp_security_protocol = pap
slip_ppp_security = N
	Use **port_password** for incoming password. Do not log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = chap
slip_ppp_security = Y
	Use ACP for authentication, sending username challenge, and challenge response.

enable_security = Y
ppp_security_protocol = chap
slip_ppp_security = N
	Use **port_password** for incoming secret token. Do not log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = chap-pap
slip_ppp_security = Y
	Request CHAP in negotiation; if it is NAKed by peer, request PAP.
	If using PAP, use ACP for incoming user name and password. If using CHAP, use ACP for authentication, sending username, and challenge response.
	Log accesses in the ACP log file.

enable_security = Y
ppp_security_protocol = chap-pap
slip_ppp_security = N
	Request CHAP in negotiation; if it is NAKed by peer, request PAP.
	Use **port_password** for incoming password/secret token and ignore incoming user name.
	Do not log accesses in the ACP log file.

Using Filters for Security

The RA 6300 implementation of filtering allows you to improve the security of an internal network by preventing potentially dangerous traffic from crossing it. For example, you might want to prevent an outside host from using the Network File System (NFS) protocol or the Trivial File Transfer Protocol (TFTP) to access an internal network, since these protocols have no built-in security and can alter local data. Or, you might want to use filtering to prevent users on your internal network from accessing external hosts and services.

An effective way to provide this kind of protection is to pick one RA 6300 on the internal network to be the network's *chokepoint* or *firewall* through which all traffic to and from external networks must pass. Then, configure filters on that RA 6300 to block undesirable packets.

You can also use filtering to log (in the **syslog** file) traffic for security or network-management purposes.

For the RA 6300, filtering applies only to the following interfaces:

- lo0
- en0

The only option available when enabling or disabling filters for the RA 6300 through cli command is **all (*)** which means that you can either enable or disable all filters for the specified interface.

Filters can be applied to individual users on a session-specific basis through the acp_userinfo file. See the subsection *filter* on page 15-259 in the *Creating the acp_userinfo File* section of this chapter.

For more details on filtering, see *Filtering* on page A-143.

Using Kerberos Authentication

The default ACP configuration authenticates a user by checking the user name and password against entries in the **acp_passwd** file. You can configure ACP to use Kerberos instead of the default authentication process.

When building the ACP/**erpcd** process, a Kerberos library routine (**libkrb.a**) is linked with the ACP code. ACP prompts the user for a user name and password. However, instead of validating the user name and password via the **acp_passwd** file, ACP opens a connection to the Kerberos server and passes the user name and password to the Kerberos library routine for authentication. The Kerberos library routine returns a ticket to ACP indicating whether or not the user is authenticated.

If the Kerberos server authenticates the user, it encrypts the ticket with the user's password before returning it to ACP. If the Kerberos server rejects the user, it returns an error code, and ACP refuses the login attempt. In either case, ACP calls a separate Kerberos routine to destroy the returned ticket after the validation process.

Enabling Kerberos Authentication

To enable Kerberos authentication, you must rebuild the **erpcd** process, and then use this process instead of the default version. To rebuild **erpcd**:

1. **Edit the** make.config file in the /annex_root/src directory **and look for the keyword CFG_STUBLINKING, at the bottom of the file. The line will look like this:**

   ```
   CFG_STUBLINKING = L. -lstubs
   ```

2. **Modify the line in Step 1 to include the libkrb.a file, as follows:**

   ```
   CFG_STUBLINKING = Kerberos_lib_path/libkrb.a
   ```

   For *Kerberos_lib_path*, specify the name of the directory containing the libkrb.a file. This file is located in the directory in which Kerberos was installed.

3. **Rebuild** erpcd **(see *Re-compiling erpcd* on page 15-336).**
4. **Install the new** erpcd **in the usual place (saving the old version as a back-up in case of problems).**
5. **Terminate the executing** erpcd **and start up the new version.**

If both the primary and secondary ACP servers are defined, it is important that both the primary and secondary ACP servers support Kerberos authentication for consistency.

Configuring the RA 6300 for Use with Kerberos Authentication

To configure the RA 6300 for use with Kerberos authentication, you must set the parameters as indicated in Table A-41.

Table A-41. Kerberos Parameter Settings

Parameter	Setting
enable_security	Yes
security_broadcast	No
port_server_security	Yes
vcli_security	Yes
vcli_password	<unset>
password	<unset>
cli_security	Yes (on each serial port)
port_password	<unset> (on each port)

Using the ACE/Server

The ACE/Server token is an access control security token which is used to positively identify users of computer systems and secure TCP/IP networks. Used in conjunction with the SecurID card hardware or software access control modules (ACMs), the ACE/Server token automatically generates a unique, unpredictable access code every 60 seconds. The ACE/Server, a daemon that interfaces with the user database, also allows the system administrator to monitor access by running reports of all attempted logins.

Supported ACE/Server Releases

Remote Annex R4.1 and later releases offer support for ACE/Server Releases 2.1.1 and 2.2.

ACE/Server is supported using ACP and is limited to those UNIX platforms for which the vendor provides client libraries.

Using the SecurID Card

To use the SecurID card feature, you must purchase the ACE/Server software from Security Dynamics. The ACE/Server software includes client software and the SecurID card. The ACE/Server system is designed to prevent any unauthorized access to your network.

SecurID Card Description

The SecurID card is a credit-card sized card containing a microprocessor and an LCD display. This card generates, at a designated interval, a one-time-only, unpredictable code on the LCD display. At the usual system prompt from your RA 6300, SecurID card users enter a passcode in order to access your protected system.

ACE/Server Authentication

The ACE/Server system provides a unique code, such as the user’s PIN number.

Each SecurID card has a unique serial number that identifies it to the ACE/Server.

Assigning a SecurID Card to a User

When you receive the ACE/Server software and SecurID cards, one of the cards is already assigned to the login name *adm* and is enabled for your system administrator. When you become *adm* and execute **sdadmin**, it determines that you have assumed that login name and uses it to find the **adm** card and your correct authority level. The **sdadmin** does not require a passcode entry. By using the one card with administrator authority, at least one person in your SecurID system has the authority to manage the ACE/Server system and all its databases, including changing any relevant information.

Clients

An ACE/Server UNIX Client is a TCP/IP machine connected via a network to the ACE/Server. Whenever a client sends a user-authentication request, the ACE/Server looks up the client's name. For this name to be found, all clients network addresses must be entered into the database, and all the network addresses must be known to the server via the **/etc/hosts** file or your NIS name server.

The SecurID Card User Interface

When a user tries to log into your system, the ACE/Server prompts for the user name and passcode. The user enters the PIN number followed by the current SecurID card code displayed on the SecurID card.

Access Types

The ACE/Server utility authenticates two access types:

- Port-to-port RA 6300
- Network-to-port

To authenticate SLIP, PPP, and IPX users:

Authenticate SLIP, PPP, and IPX

1. **Log into a CLI port.**
2. **Issue the CLI command** slip, ppp**, or** ipx**.**

   Or
3. **Log into** auto_detect **and** auto_adapt **ports.**
4. **Press** Return **to enter CLI mode, and then issue the** slip, ppp**, or** ipx **command.**

Generating PINs

When...	The...
a SecurID card is assigned to a user	card is set to New-PIN mode in the ACE/Server database.
a user attempts to log on for the first time to a network via a Remote Annex	user enters only the code on the SecurID card (if the PIN has been cleared).
SecurID requires a unique PIN	user must enter a new PIN (user- or system-generated).

The ACE/Server software provides three options related to generating a new PIN:

- CANNOT_CHOOSE_PIN
- MUST_CHOOSE_PIN
- USER_SELECTABLE

Before installing the ACE/Server software, you must determine which of the above options your site will use. The following is an overview of the available options. See the *ACE/Server Manual* for more information.

CANNOT_CHOOSE_PIN

The new PIN is generated by the system and does not give the user the option to select a new PIN. The user is prompted to allow the system to generate and display the new PIN or exit and leave the SecurID card in New PIN mode.

MUST_CHOOSE_PIN

The user must select a new PIN and is not given the option of having the system generate the new PIN. The user is prompted to enter a new PIN containing 4 to 8 alphanumeric characters or exit and leave the SecurID card in New PIN mode.

USER_SELECTABLE

The user is given the option to select a PIN or have the system generate and display a new PIN. The user is prompted to enter a new PIN containing 4 to 8 alphanumeric characters or have the system generate a new PIN and display it or leave the SecurID card in New-PIN mode.

Installation

Copy Files to src/sdclient

During the Remote Annex software installation, you must copy the following library and files from your ACE/Server distribution media to the **src/sdclient** directory:

- sdclient.a library
- *.h files

The ACE/Server UNIX Client must be installed on each host running **erpcd**. For more detailed information, see the *ACE/Server Installation Guide*.

Each UNIX host running a SecurID-enabled **erpcd** must be enabled as a client in the ACE/Server. For more detailed information, see the *ACE/Server Administration Manual*.

Makefile Switches

Define Makefile Switches

Define a new set of switches in the **Makefile** by uncommenting the two lines that define ACE1_2 or ACE2_0 in **erpcd/Makefile** for ACE/Server V2.1.1 or V2.2. Also, comment out the flag (**PASSFLAG**) that causes the Annex password prompt to appear:

Example

```
#SECURIDFLAG=-DSECURID_CARD -DACE2_0

#SECURIDFILES=../sdclient/sdiclient.a

...

PASSFLAG = -DPASS_SEC
```

to

```
SECURIDFLAG=-DSECURID_CARD -DACE2_0

SECURIDFILES=../sdclient/sdiclient.a

...

#PASSFLAG = -DPASS_SEC
```

To integrate SecurID into ACP, you must make changes in the **erpcd** utility. When you have made the necessary changes to the **Makefile**, rebuild the Remote Annex software. See *Re-compiling erpcd*, later in this chapter.

Define Makefile Switches for non-ANSI Standard Compiler

With a non-ANSI standard C compiler, uncomment the following lines in the **Makefile**:

```
#SECURIDCFILES=fflush.c

#SECURIDOFILES=fflush.o
```

to

```
SECURIDCFILES=fflush.c

SECURIDOFILES=fflush.o
```

New-PIN Mode

If the site allows a user to select a PIN, ACP displays the following text:

```
Enter your new PIN containing 4 to 8 digits,
     or
Press <Return> to generate a new PIN and display it,
     or
<Ctrl-D> <Return> to leave your card in New-Pin mode.
```

The minimum and maximum PIN lengths and the choice between digits only or alphanumeric characters is determined by the system administrator when installing the ACE/Server.

If the user enters a PIN, ACP prompts for the code's re-entry (the typed characters are not echoed back to the terminal). The re-entry prompt looks like this:

```
Please re-enter PIN:
```

If the user is not allowed to choose the PIN, the following text is displayed:

```
Press <Return> to generate a new PIN and display it,
     or
<Ctrl-D> <Return> to leave your card in New-Pin mode.
```

If the user presses **Return**, the terminal displays the assigned PIN. If the user incorrectly re-enters the PIN or chooses to leave the card in New-Pin mode, the login attempt is terminated.

Configuring the RA 6300 for Use with SecurID

To use the SecurID card, security must be enabled on the RA 6300:

1. **Set the following RA 6300 parameters to** Y**:**
   - **enable_security**
   - **vcli_security**
2. **Set the following RA 6300 port parameters to** Y **on the global port:**
   - **cli_security**
   - **port_server_security**
3. **Set the RA 6300 port parameter** ppp_security_protocol **to** none **on each port.**

   If **ppp_security_protocol** is set to **none**, the user will be prompted again for user name and passcode when trying to use the CLI **ppp** command. The user must enter the PIN and SecurID card code for the passcode.

   If you do not want to be prompted a second time, set **ppp_sec_auto** to **Y**.
4. **Set the RA 6300 parameters** password **and** vcli_password **and the port parameter** port_password **to the null string ("") if you want the ACE/Server system to authenticate all login attempts before allowing access to the RA 6300. Also, do not set a port password in the** acp_passwd **file when using SecurID.**
5. **Enter a host name or IP address for the** pref_secure1_host **and** pref_secure2_host **parameters for each RA 6300 using a SecurID card. The host addresses where each ACP process runs must be activated in the ACE/Server database as clients.**

6. **Set the RA 6300 parameter** security_broadcast **to** N **so that the RA 6300 does not inadvertently contact an ACP process that does user authentication via the** acp_passwd **file unless all the ACP server processes in your network are configured and installed to do user authentication by calling the ACE/Server.**
7. **Set the RA 6300 parameter** acp_key **to its assigned value and enter this value into the** acp_keys **file on the host. Then ACP and the RA 6300 exchange user names and passcodes encrypted with the key.**
8. **Activate valid RA 6300 users in the ACE/Server database with permissions (individual or group) to access the ACP servers. If two ACP servers are used, each user must be allowed access to both servers since either of them can authenticate a user by calling the ACE/Server host.**

Integrating SecurID into ACP

Integrating the ACE/Server software into ACP requires changes to the **erpcd** utility. The following instructions assume that the ACE/Server software is installed in a directory called **/usr/ace** and the RA 6300 software is installed in **/usr/annex**; if your code is installed in different directories, substitute the appropriate pathnames where applicable.

These instructions assume that the software is installed on a UNIX system and that the host tools have been compiled (as opposed to using the binaries from the RA 6300 distribution tape). Also, the target UNIX system requires a development environment (*C* compiler, libraries, etc.).

Make sure the host clock is set correctly.

1. **As a superuser, change into the** /usr/annex/src **directory:**

```
# cd /usr/annex/src
```

2. **Create a directory called** sdclient**:**

```
# mkdir sdclient
```

3. **Copy the required header files and libraries from the ACE/Server directories:**

   If you have ACE/Server Release 2.1.1 or 2.2:

   ```
   # cp /usr/ace/sdiclient.a sdclient
   # cp /usr/ace/prog/*.h sdclient
   ```

   This sequence requires that these files are installed on the slave/client system from the ACE/Server host.

   Make sure the ACE/Server UNIX Client is installed on the system that is running **erpcd**.

4. **Edit the** Makefile **file in the** /usr/annex/src/erpcd **directory:**

   ```
   # vi Makefile
   ```

5. **Kill the existing** erpcd **process (your process number will vary):**

   ```
   # ps -ax | grep erpcd
   25493 IW 0:00 ./erpcd
   25494 IW 0:00 ./erpcd

   25797 p1 S 0:00 grep erpcd
   # kill 25493
   ```

6. **Rebuild** erpcd **(see *Re-compiling erpcd* on page 15-336).**

7. **If you have linker errors try running the** ranlib **utility on the** sdiclient.a **library:**

   ```
   # ranlib sdclient/sdiclient.a
   ```

   Then rebuild **erpcd** (see *Re-compiling erpcd* on page 15-336).

8. **Make sure that ACP is enabled in the** eservices **file (the default is ACP disabled). The default file looks like this:**

```
# erpc  remote programs
#
# prog noverlo verhi  name
#
  1       0     0     bfs
# 3       0     99    acp
```

Enable ACP by removing the pound sign (#) from its entry. The edited file looks like this:

```
# erpc  remote programs
#
# prog noverlo verhi  name
#
  1       0     0     bfs
  3       0     99    acp
```

9. **Run** erpcd **from the current directory or install the newly built** erpcd **in the** /usr/annex **directory by entering:**

```
# ./erpcd
```

or

```
# mv /usr/annex/erpcd /usr/annex/erpcd.old
# make install
# /usr/annex/erpcd
```

10. **Now follow the procedures in the ACE/Server documentation for registering clients and users. The hosts where erpcd is running must be registered as clients, and all users with SecurID cards that will log into the RA 6300(s) must be allowed to access the host clients.**

11. **On the RA 6300, enable security, configure the preferred security server, and enable CLI security on the ports to be protected by SecurID. If you have a secondary server, the new** erpcd **must be installed on that host and that host must be registered as a client in the ACE/Server database.**

    A sample **admin** session looks like this:

    ```
    admin: set annex enable_security Y security_broadcast N
    admin: set annex pref_secure1_host calvin
    admin: set port cli_security Y
    admin: reset port
    ```

SecurID Backup Security

The RA 6300 uses the following procedures if the server running SecurID and **erpcd** is down:

- If the RA 6300 finds another server running **erpcd** but not SecurID, ACP will control RA 6300 security.
- If the RA 6300 cannot find another server running **erpcd**, the RA 6300 uses local security.

Using SafeWord AS Security

Enigma Logic’s SafeWord AS software verifies the identity of CLI users to permit access to protected systems. When you install SafeWord version 4.x on a central network server and link SafeWord to **erpcd**, SafeWord provides and authenticates fixed or dynamic passwords. SafeWord also supports the RADIUS server.

The difference in the application of SafeWord AS for this release is that the client/server approach now allows **erpcd** to communicate only with the SafeWord server through a client API. The server then interfaces with the database. Also, another difference is that clients are allowed to be on different hosts.

ACP hosts serve as clients to SafeWord AS.

You can use SafeWord software for:

- SLIP, PPP, IPX, and ARAP sessions only when you start a session from a CLI port. IPX users must connect from Fastlink II in terminal mode.
- The ARAP Remote Access client’s CCL scripts in versions 1.0 and 2.0 as long as you do not use a SafeWord challenge as part of a dynamic password.

  ARAP does not use the **acp_regime** file.
- Macintosh or dial-back users only when the SafeWord user name matches the user name listed in the **acp_userinfo** file.

The RA 6300 supports SafeWord for user authentication only, therefore authorization is not supported. Therefore, when you dial in to the network through an RA 6300, or dial out from an RA 6300 (e.g., if you telnet to a port in slave mode), the RA 6300 does not display the SafeWord Failed Access Report. In addition, the RA 6300 does not run the user's SafeWord execute program at the end of the authentication process.

Installing SafeWord AS

To integrate SafeWord into ACP, you must make changes in the **erpcd** utility. You must install SafeWord:

- On a host running **erpcd**.
- On a UNIX system that has a development environment. You compile the host tools on this system (as opposed to using the binaries from the RA 6300 distribution tape).
- In a directory named **/safelog** and Annex software in the **/usr/annex** directory. If you do not use these directory names, you must substitute pathnames.

Copy the following files during installation from the SafeWord AS installation directory into your **src/enigma** directory:

- NETWORKAPI/swecapi.h
- NETWORKAPI/swecapi.a
- LOCALAPI/custfail.h
- LOCALAPI/custpb.h

For a successful NETWORKAPI installation, you must also install the SafeWord AS client for UNIX machine (option 2 in the installation script) during the installation.

Makefile Switches

Define **Makefile** Switches

Define a new set of switches in the **Makefile** by uncommenting the following lines in **erpcd/Makefile**:

```
#ENIGMAFLAG=-DENIGMA_SAFEWORD -DNET_ENIGMA_ACP
#ENIGMAFILES=../enigma/swecapi.a
#ENIGMACFILES=acp_safeword.c
#ENIGMAOFILES=acp_safeword.o
```

“__assert” comes up undefined (the default). You must uncomment the following line as well:

```
#ENIGMAFLAG = -DENIGMA_SAFEWORD -DNET_ENIGMA_ACP
-DNEED_ENIGMA_ASSERT_PATCH
```

Configuration Management

Place a new file, called **safeword.cfg**, in the annex installation directory. This file is created as **sid.cfg** when you install the SafeWord AS client.

Move and Rename the **sid.cfg** File

To place the new file in the installation directory:

12. **Copy the** sid.cfg **file into the installation directory.**
13. **Rename the** sid.cfg **file to** safeword.cfg**.**

Create the **safeword.cfg** File

To create the **safeword.cfg** file, use the following example and replace *yourservername* with the name of your SafeWord server:

```
02 Authen. Server (host weight connects port):
     yourservername 0 0 7482

09 User ID Source (USER/SYSTEM): USER

10 Server's System Name: STANDARD

15 Send Status Messages to User: NONE

16 Send Status Messages to Console: ERROR

17 Send Status Messages to log File: NONE

18 Status Message Log Filename: sid.log

23 Status Message Label: sid-7482
```

Integrating SafeWord into ACP

Before you use SafeWord, you need to integrate SafeWord into ACP:

1. **As a superuser, change to the** /usr/annex/src **directory:**

   ```
   # cd /usr/annex/src
   ```

2. **Create a directory called** enigma**:**

   ```
   # mkdir enigma
   ```

3. **Copy the** libidpb.a, custpb.h**, and** custfail.h **files into the enigma directory:**

   ```
   # cp /safelog/swecapi.a enigma/swecapi.h
   # cp /safelog/custpb.h enigma/custpb.h
   # cp /safelog/custfail.h enigma/custfail.h
   ```

SafeWord's standard installation provides the **libidpb.a**, **custpb.h**, and **custfail.h** files in the **/safelog** directory.

4. **Edit the** make.config **file in the /annex/root/src directory:**

   ```
   # vi make.config
   ```

5. **Locate the following line, which is near the bottom of make.config:**

   ```
   CFG_STUBLINKING = -L. -lstubs
   ```

   Change the line to include the SafeWord library, as follows:

   ```
   CFG_STUBLINKING = ../enigma/libidpb.a -L. -lstubs
   ```

6. **If** erpcd **is running on the host, kill the existing** erpcd **process (your process number will vary):**

   ```
   # ps –ax | grep erpcd
   25493 ? IW 0:00 ./erpcd
   25797 p1 S 0:00 grep erpcd
   # kill -9 25493
   ```

7. **Rebuild** erpcd**:**

   ```
   # make erpcd. If you have linkage errors, try running
   the ranlib utility on the sdclient.a library:
   ```

   ```
   # ranlib enigma/lipidpb.a
   # make erpcd
   ```

8. **Install** erpcd **into the** usr/annex **directory:**

   ```
   # make install
   ```

9. **Restart** erpcd**:**

   ```
   # /usr/annex/erpcd
   ```

10. **On the RA 6300, use** admin **or** na **to** set pref_secure1_host **to the Internet address of the host running SafeWord and** erpcd**.**

    You can enter the backup host's address in the **pref_secure2_host** parameter.

SafeWord Passwords

SafeWord provides fixed and dynamic passwords to verify user access to protected systems.

While SafeWord's IDUTIL program allows administrators to create up to three levels of authentication for each user, RA 6300 access allows you to combine one dynamic and one fixed password: you cannot use two dynamic or two fixed passwords for a single authentication process.

If you configure a SafeWord startup file, it will not run when a user accesses an RA 6300.

Fixed Passwords

System administrators can generate a user's initial fixed password and can set the password's expiration date. When an existing password expires, RA 6300 users can choose a new fixed password:

1. **If the expiration message appears after you enter your username and password, press the** Escape **key and then press** Return.

   The *Old Fixed Password* message appears.

2. **Enter your old password and press** Return.

   The *New Fixed Password* message appears.

3. **Type your new password and press** Return.

   The *Repeat New Fixed Password* message appears.

4. **Type your new password again and press** Return.

   If you completed these steps correctly, the *Permission Granted* message appears. If you did not, SafeWord displays an error message.

Dynamic Passwords

SafeWord generates dynamic passwords using a hand-held password generator called a “token.” The token generates new passwords each time a user wants to access protected systems. Network administrators can configure SafeWord’s dynamic passwords in Synchronous, Semi-synchronous, and Asynchronous modes:

- In Synchronous mode authentication, the token generates a dynamic password that you enter at your terminal.
- In Semi-synchronous mode:
  - Enter the password from your previous session into the token, which then displays a new password.
  - Enter the dynamic password at your terminal.
- In Asynchronous mode:
  - The token displays a string, called a “challenge” before you enter a dynamic password.
  - Enter the challenge into the token, which generates a dynamic password.
  - Enter the dynamic password at your terminal.

You cannot use this (or any other authentication technique that uses a challenge) with PAP or IPX security, because neither PAP nor IPX allow challenges.

For detailed information about configuring and generating fixed and dynamic passwords, refer to Enigma Logic’s SafeWord documentation.

SafeWord Backup Security

The RA 6300 uses the following procedures if the server running SafeWord and **erpcd** is down:

- If the RA 6300 finds another server running **erpcd** but not SafeWord, ACP will control RA 6300 security.
- If the RA 6300 cannot find another server running **erpcd**, the RA 6300 uses local security.

Configuring Security for the RA 6300 FTP Daemon

When a new FTP session is initiated, the FTP daemon registers the source host with the **who** database. A subsequent **who** displays:

```
annex: who

 Port   What   User   Location   When   Idle Address
 1      PSVR   ---    jdcm       4:06am 3:13 192.9.200.60
                      console
 v1     CLI    hobbes ---        4:07am      192.9.200.60
 v2     FTPD   ---    ---        ---    :01  bryce
 annex:
```

In the above sample command display, since the user has not yet logged into the **ftp** session, no user name appears in the *User* field. If the **enable_security** parameter is set to **Y** but a preferred security server is not configured, or if **enable_security** is set to **N**, the user is prompted for a user name and a password. The RA 6300 will accept any user name, but grants FTP access only after checking the password against its administrative password. If the RA 6300 grants access, the user’s name appears in the **who** command display.

If the **enable_security** parameter is set to **Y** and a preferred security server is configured, the RA 6300 calls the **ppp_security** function in the **acp_policy.c** file with the user’s name and password as entered and the service set to SERVICE_FTP. If ACP grants access, the FTP daemon will ask for an “account.” The RA 6300 compares the text entered at this prompt against its administrative password for an added level of security.

If the **enable_security** parameter is set to **Y** and the preferred security server is not reachable, the RA 6300 denies access to the FTP daemon.

When the validation process is complete, the RA 6300 logs FTP access in the ACP logfile (see *Host-based Security Logging* on page B-33) and updates the **who** command display to look something like this:

```
annex: who

 Port   What   User    Location    When    Idle  Address
 1      PSVR   ---     jdcm        4:06am  3:37  192.9.200.60
                       console
 v1     CLI    hobbes  ---         4:07am        192.9.200.60
 v2     FTPD   hobbes  ---         2:43pm        bryce
 annex:
```

The RA 6300 FTP daemon is compatible with all versions of UNIX **ftp.**

You can completely disable the RA 6300 FTP daemon by setting **ftpd** in the **disabled_modules** parameter.

Configuring the IP Basic Security Option (IPSO)

The Department of Defense Basic Security Option for IP identifies the U.S. classification level at which an IP datagram is to be protected and the authorities whose protection rules apply to each datagram, as defined in RFC 1108. The RA 6300 partially implements this security option by adding the IPSO classification level to packets generated by **telnet** or **rlogin** running on an RA 6300 dedicated, adaptive, or CLI port. (The CLI port can be an **auto_detect** or **auto_adapt** port that the user has put into **cli mode** by pressing **Return** when first connected to the port.) The RA 6300 does not add the option to locally generated system packets, such as ICMP messages and RIP updates. Nor does the RA 6300 check incoming packets for the presence of the IP Security Option.

To set the IPSO for packets generated on a port:

1. **Use the** na **utility, the superuser CLI** admin **command, or SNMP to set the RA 6300 parameter** enable_security **to** Y **(the default is** N**).**
2. **Use** na**,** admin**, or SNMP to set the serial line port parameter** ipso_class **to one of the following values:** topsecret**,** secret**,** confidential**,** unclassified**, or** none**. If you specify** none **(the default), the RA 6300 does not add the option to packets.**

The **ipso_class** parameter is also an object in the private-enterprise MIB and can be set via SNMP (for more details, see *Simple Network Management Protocol (SNMP)* on page B-57).

The following sample **su** session causes a basic security option of secret to be included in all packets generated by ports 1 and 2.

```
annex: su
Password:
annex# admin
Annex administration Remote Annex Rx.x, 72 ports
admin: set port=1,2 ipso_class secret
admin: set port mode cli
admin:
```

When a router that fully implements IPSO receives a packet with an unacceptable classification level, it sends an ICMP security discard message to the packet's originator. If the RA 6300 receives a discard message, it passes it to the application running on the port that generated the IPSO packet.

Logging Security Events

Host-based security can generate audit trails of user activity. Each time the security server grants or denies a request for user access, the security server logs it. Each event is logged as a message in an ACP log file.

The ACP log file can be the default **acp_logfile** located in the **/usr/annex** directory or an RA 6300-specific log file. An RA 6300-specific log file is created by uncommenting the following statement in the **acp_policy.h** file:

```
#define SEPARATE_LOGS
```

Once this statement is uncommented, an RA 6300-specific log file is created with the name **acp_logfile.*Annex_IPaddress*** in the **/usr/annex** directory.

Each logged message in the ACP log file contains the following fields:

- *IP address of the RA 6300.*
- *Sequence number.*
- *Port type.*
- *Date.*
- *Time.*
- *Module.*
- *Event.*
- *Packets in.*
- *Packets out.*

- *Bytes in.*
- *Bytes out.*
- *Protocol-dependent information.*
- *Username*.

For the RA 6300, the following additional fields are added:

- *Calling number.*
- *Called number.*
- *Called subaddress.*
- *Bearer.*

All fields are separated by colons and are encoded for use by UNIX utilities that sort, merge, select, or filter streams. *Host-based Security Logging* on page B-33 provides a sample log file.

The parser of the **acp_userinfo** file generates log messages if an error is detected when processing a user’s profile.

Modifying the Supplied Security Application

You can modify the supplied security policy to create a security scheme that meets the needs of your network. Some simple modifications involve changing system definitions in the file */annex_root/***src/erpcd/acp_policy.h**. More elaborate security policies may require modifying or replacing functions in the file */annex_root/***src/erpcd/acp_policy.c**.

Do not change the function declarations or the description of the interface; these are fixed by the calls made into this library. Before making even the smallest change, save the base version of the file requiring modification.

If you modify the default policy, you must re-compile **erpcd**, kill the current version, and start the new version (see *Modifying the Code* on page 15-335).

Disabling User Name and Password Validation

When security is enabled, users must provide a user name and password. You can disable this policy by modifying the */annex_root*/**src/erpcd/acp_policy.h** file.

To disable the user name requirement, change the line that defines user validation from:

```
#define USER_VALIDATION 1   to  #define USER_VALIDATION 0
```

Messages are logged to the security server host when users access the CLI, but the message does not include a user name.

To disable the port password requirement, make sure the following line is commented out (i.e., enclosed in asterisks), as follows:

```
/* #define PORT_PASSWORD 1 */
```

Linking NIS Password File Verification to ACP

You can enable several options in the **acp_policy.h** file by removing the slash (/) and asterisk (*) at the beginning and the end of the definition line.

To use the NIS password file for verification through ACP, change (uncomment) the following lines:

```
/* #define NATIVEPASSWD 1 */ to #define NATIVEPASSWD 1
/* #define NATIVESHADOW 1 */ to #define NATIVESHADOW 1
```

You can change several other options in the same way:

```
/*
     * Uncomment this line to select the use of the\
     * standard syslog(3) facility in addition to or in\
     * place of the logfile -- the value of "USE_SYSLOG"\
     * is used to identify the daemon.(Comment the
     * second line out to disable the normal acp log file.)
*/
/* #define USE_SYSLOG "annex" */
#define USE_LOGFILE 1
```

```
/*
     * Uncomment this line to use decoded Annex peer names,\
     * rather than numeric IP addresses, in the log file
     * and in syslogging.
*/
/* #define USE_ANAME 1 */
```

Modifying Message Formats in the ACP Log File

The USE_SECONDS option in the **acp_policy.h** file enables messages in the ACP log file to use a *seconds-since-1970* (ten decimal digits) format. This format is most useful for automatic ACP log file parsing programs since these programs frequently need to do comparisons and arithmetic on dates. This option is disabled by default.

You can enable USE_SECONDS by changing (uncommenting) the following line:

```
/* #define USE_SECONDS 1*/ to #define USE_SECONDS 1
```

The standard message format in the ACP log file is:

```
<annex_name>:<logid>:#<port>:<yymmdd>:<hhmmss>:<service>:\
<event>:<pkts in>:<pkts out>:<bytes in>:<bytes out>:<msg>
```

When USE_SECONDS is enabled, the message format in the ACP log file is:

```
<annex_name>:<logid>:#<port>:<seconds_since_1970>:\
<service>:<event>:<pkts in>:<pkts out>:<bytes in>:\
<bytes out>:<msg>
```

Changing the Expected File Names Used by ACP

The supplied policy uses names for various files. For example: **acp_passwd**, **acp_keys**, **acp_restrict**, and **acp_logfile**. You can change the names of any of these files in the */annex_root/***src/erpcd/acp_policy.h** file.

If you decide to use either an existing system or a network-wide password file instead of the **acp_passwd** file, change the following lines in the **acp_policy.h** file:

```
#define ACP_PASSWD (str) \
   sprintf(str,"%s/acp_passwd",install_dir)

#define ACP_PTMP (str) \
   sprintf(str,"%s/acp_ptmp",install_dir)
```

To change only the filename:

```
#define ACP_PASSWD (str) \
   sprintf(str,"%s/new_filename",install_dir)

#define ACP_PTMP (str) \
   sprintf(str,"%s/new_tempfile",install_dir)
```

To change the full pathname:

```
#define ACP_PASSWD (str) \
   sprintf(str,"new_path/new_filename")

#define ACP_PTMP (str) \
   sprintf(str,"new_path/new_tempfile")
```

The *new_filename* is the name of the new password file, and the *new_tempfile* is a temporary file used by the **ch_passwd** command. Since you do not need the temporary file if you are using an existing system file, comment out the line for the temporary file.

The **install_dir** is defined in the file */annex_root/***src/make.config** with the leading quote supplied by the makefile. Since the trailing quote is required by the two strings, double quote the names for the new password and temporary files.

You can change the names of several other files in the **acp_policy.h** file in the same way:

```
#ifdef NATIVESHADOW

#define ACP_SHADOW(str)\
     strcpy(str,"/etc/shadow")
#define ACP_STMP(str)\
     strcpy(str,"/etc/shadow.tmp")
#define ACP_LOCKFILE(str)\
     strcpy(str,"/etc/.pwd.lock")
#define ACP_GROUP(str)\
     strcpy(str,"/etc/group")
#else

#define ACP_SHADOW(str)\
     sprintf(str,"%s/acp_shadow",install_dir)

#define ACP_STMP(str)\
     sprintf(str,"%s/acp_stmp",install_dir)

#define ACP_LOCKFILE(str)\
     sprintf(str,"%s/.pwd.lock",install_dir)

#define ACP_GROUP(str)\
     sprintf(str,"%s/acp_group",install_dir)

#endif

/*   define pathname of accounting file*/

#define ACP_LOGFILE(str) \

     sprintf(str,"%s/acp_logfile",install_dir)

/*   define pathname for restrictions file*/
```

(continued on next page)

```
#define ACP_RESTRICT(str) \

     sprintf(str,"%s/acp_restrict",install_dir)

/*   define pathanme for annex acp_keys file */

#define ACP_KEYS(str) \

     sprintf(str,"%s/acp_keys",install_dir)

/*   define pathanme for annex dialup addresses file */

#define ACP_DIALUP(str) \

     sprintf(str,"%s/acp_dialup",install_dir)

/*   define pathname for user profile file */

#define ACP_USERINFO(str) \

     sprintf(str,"%s/acp_userinfo",install_dir)

#define ACP_ESERVICES(str) \

     sprintf(str,"%s/eservices",install_dir)
```

In the same way, you can also change the expected prompts for default applications:

```
#ifndef SECURID_CARD

#define ACP_USERPROMPT "Annex username: "
#define ACP_PASSPROMPT "Annex password: "
#define ACP_PERMGRANTD "\nPermission granted\n"
#define ACP_PERMDENIED "\007\nPermission denied\n"
#define ACP_INCORRECT "\nUsername/Password Incorrect\n"

#else

#define ACP_USERPROMPT "Username: "
#define ACP_PASSPROMPT "Enter PASSCODE: "
#define ACP_PERMGRANTD "\nPASSCODE accepted\n"
#define ACP_PERMDENIED "\007\nAccess Denied\n"
#define ACP_INCORRECT "\nUsername/PASSCODE Incorrect\n"

#endif
```

(continued on next page)

```
#define ACP_TIMEDOUT "\007\nLogin Timed Out\n"
#define ACP_WARNING "\007\nYour password will expire \
     in %ld days unless changed.\n"
#define ACP_WARNINGM "\007\nYour password expires after\
     tomorrow unless changed.\n"
#define ACP_WARNINGT "\007\nYour password expires after \
     today unless changed.\n"

#define ACP_AWARNING "\007\nYour account will expire in\
     %ld days.\n"
#define ACP_AWARNINGM "\007\nYour account expires after\
     tomorrow.\n"
#define ACP_AWARNINGT "\007\nYour account expires after\
     today.\n"
#define ACP_EXPIRED "Your password has expired.\n"
#define ACP_NEWPASS "Enter a new password:  "
#define ACP_NEWPASS2 "Re-enter new password:  "

#define ACP_PASSMATCH "Entered passwords do not match.\
     Try again.\n"
#define ACP_ACCESSCODEPROMPT "Access Code: "
#define ACP_PHONEPROMPT "Telephone Number: "
#define ACP_DIALBACKGRANTD "\nRequest accepted,dialback in\
     progress\n"
#define ACP_CLINODIALBACK "\nPermission granted, no\
     dialback\n"

/*   define messages used by Securid Card application*/

#ifdef SECURID_CARD
#define ACP_NEXTCODEPROMPT "Enter next card code: "
#define ACP_PINCHAR "characters"
#define ACP_PINDIGIT "digits"
#define ACP_PINSIZE "%d"
#define ACP_PINSZRANGE "%d to %d"
#define ACP_NEWPINPROMPT "Enter your new PIN containing %\
     %s,\n"
```

(continued on next page)

```
#define ACP_OR "\t\tor\n"
#define ACP_NEWPIN_2 "Press Return to generate new PIN and\
     display it\n"
#define ACP_NEWPIN_3 "<Ctrl d> to leave your card in New-PIN\
     mode.\n"
#define ACP_SYSGENPIN "\t\t%s\n"
#define ACP_PINREENTRY "Please re-enter PIN: "
#endif#ifdef PORT_PASSWORD
/* only if PORT_PASSWORD is set and a port password exists
in acp_passwd */
#define ACP_PORTPROMPT "Port password: "
#endif

/* miscellaneous defines for default application */

#define INPUT_TIMEOUT 30
#define INPUT_POLL_TIMEOUT 3
#define RETRIES_MAX 3
```

Locking the ACP Log File

To prevent two or more host processes from logging a record simultaneously, the RA 6300 **erpcd** code uses the host system call **lockf** to lock the ACP log file. This lock prevents other processes from writing the file until the file update is complete.

There are two ways to use the system **lockf** call. You can select either mechanism via a switch in the **acp_policy.h** file. The following explanation of the switch resides in this file. The default method, T_LOCK, is reliable but not very efficient; F_LOCK is more efficient but does not work on all hosts (some host manufacturers have issued patches that resolve this issue).

```
/*
* Uncomment this line to select the F_LOCK method to lock the
* ACP log file for updating.
*
* A file must be locked for update in order to block other
* processes from writing to it simultaneously.
*
* F_LOCK - Passing the F_LOCK as the cmd value when making
* system lockf call is the most efficient and preferred manner
* to lock a file for exclusive write access. In this scenario
* a process is put to sleep until the resource is available.
* Once available the process is preempted owning the resource.
*
* T_LOCK - When the T_LOCK cmd argument is passed, the process
* must repeatedly send the lockf call the until the resource
* is available.Once available the system call returns a
* success and the resource is acquired.
*
* The F_LOCK cmd has been determined to be faulty on many
* hosts. Failures can not be narrowed down to any particular
* hardware manufacturer or UNIX system. There are to many OS
* revs and variables to sense the correct lockf method to
* use at installation time. The default, T_LOCK was chosen
* simply because it has been proven reliable. SEE
* 'log_message()'
*/
/* #define USE_F_LOCK 1 */
```

Masking CLI Commands

When the security subsystem is enabled, you can mask (disable) user access to specific CLI commands by modifying the CLI_MASK line in the **acp_policy.h** file.

To disable **rlogin** and **telnet** for all users that enter the system through ACP security, modify the definition line to read:

```
#define CLI_MASK (unsigned long) (MASK_RLOGIN | MASK_TELNET)
```

To disable the CLI **who** and **su** commands for all users that enter the system through ACP security, modify the definition line to read:

```
#define CLI_MASK (unsigned long) (MASK_WHO | MASK_SU)
```

You can extend this to any set of commands by adding masks to that line separated by the vertical bar (|).

If the user enters the masked command, the CLI displays an error message. The superuser CLI commands cannot be masked individually. They can all be disabled by masking the **su** command.

Superuser CLI mode overrides ACP command masking.

You can disable several other CLI commands in the same way:

```
/*   define bit to disable each maskable CLI command*/

#define MASK_BG       0x00000001
#define MASK_CALL     0x00000002
#define MASK_FG       0x00000004
#define MASK_HANGUP   0x00000008
#define MASK_HELP     0x00000010
#define MASK_HOSTS    0x00000020
#define MASK_JOBS     0x00000040
#define MASK_KILL     0x00000080
#define MASK_NETSTAT  0x00000100
#define MASK_RLOGIN   0x00000200
#define MASK_STATS    0x00000400
#define MASK_STTY     0x00000800
#define MASK_TELNET   0x00001000
#define MASK_WHO      0x00002000
#define MASK_LOCK     0x00004000
#define MASK_SU       0x00008000
#define MASK_SLIP     0x00010000
#define MASK_CONNECT  0x00020000
#define MASK_SERVICES 0x00040000
```

(continued on next page)

```
#define MASK_PPP      0x00080000
#define MASK_ARAP     0x00100000
#define MASK_NONE     0x80000000
```

After changing the code, **cd** to the **/src** directory and recompile **erpcd**.

For more specific command disabling, e.g., by user name, you must edit the distribution policy file */annex_root*/**src/erpcd/acp_policy.c**.

Modifying the Code

You can create a more elaborate security policy application by modifying the code in the files */annex_root*/**src/erpcd/acp_policy.c** and */ annex_root*/**src/erpcd/acp_policy.h**. The program that executes ACP starts a new version of itself each time a security request is received from an RA 6300. A call is made to an ACP remote procedure, which makes calls to functions in the ACP library to prompt for user names, passwords, etc. When ACP gathers the information required to perform the authorization algorithm, it again calls functions in the library to grant or deny the request. The program then exits.

The distribution policy file **acp_policy.c** is documented in the form of *C* programming language comments. The file **policy.doc** provides a complete description of the available library functions.

Re-compiling erpcd

You must re-compile **erpcd** if you modify the supplied policy and the **ch_passwd** utility if you changed the name of the ACP password file from **acp_passwd**. The source files are in */annex_root*/**src/erpcd**, where *annex_root* is the directory to which the RA 6300's source code was copied. To re-compile:

1. **cd to** /annex_root/src**.**
2. **To re-compile only** erpcd**, enter the command:**

   ```
   # make erpcd
   ```

3. **To re-compile both** erpcd **and** ch_passwd**, enter the command:**

   ```
   # make all
   ```

4. **To install, enter the command:**

   ```
   # make –f ../make.config –f Makefile install
   ```

   This saves the old version of **erpcd** as **OLDerpcd** in the installation directory.

5. **Kill the current** erpcd **and start the new one.**

Using the ch_passwd Utility

The **ch_passwd** utility enables users to change their passwords when accessing an RA 6300 through the Access Control Protocol (ACP) security system. This utility affects only passwords in the **acp_passwd** or **acp_shadow** file. Table A-42 describes the supported argument for **ch_passwd**.

To change an RA 6300 user password, the *username* in the **acp_passwd** file must match the *username* in the **/etc/passwd** (or **/ etc/shadow**) file on the ACP host.

If ACP is configured to record password histories, it saves the passwords set via the **ch_passwd** command. ACP keeps these passwords in the **acp_dbm** database on the security host, keyed by user name. The value of the STORED_PASS variable in **acp_policy.h** determines the number of passwords saved. This variable is initialized to 6 for **passwd/shadow** files and 0 for **passwd** files alone. A value of 0 disables password history. For more information, see *Enabling and Configuring Password Histories* on page 15-281.

The **ch_passwd** utility first prompts for the old password, and then for the new one. The syntax is:

ch_passwd

A superuser can change the password for any user. The superuser syntax is:

ch_passwd [*username*] [**–s** *directory*]

If you change the name of the ACP password file, you must recompile both **erpcd** and the **ch_passwd** utility. The source files for both are provided with the RA 6300 software distribution and are located in the /*annex_root*/**src/erpcd** directory. For instructions on recompiling both, see *Configuring Hosts and Servers* on page A-209.

Table A-42. Supported Argument for ch_passwd

Argument	Description
–s *directory*	Specifies the directory for the security files (**acp_passwd** and, if configured, **acp_shadow**); defaults to the defined install-annex directory (usually **/etc/annex/**).

Chapter 1 Network Administration

This chapter discusses typical network administration tasks you can perform using Remote Annex 6300 tools and utilities. Table B-1 lists typical tasks, related commands or parameters, and where to look in this chapter for more information.

Table B-1. Administration Tasks, Commands, and Parameters

Task	Related CLI Command and/or Parameter	Instructions for Use
Display PRI information	pri command	See *Displaying PRI Information* on page 1-3
Display PRI B channel assignment information	pri b command	See *Displaying PRI B Channel Assignments* on page 1-4
Control PRI calls	pri call command	See *Controlling PRI Calls Made to the RA 6300* on page 1-5
Display session parameter blocks	sessions command	See *Displaying Active Session Parameter Blocks* on page 1-7
Manage and display information about modems	modem command and its arguments	See *Displaying Internal Modem Information* on page 1-39
Display network statistics	netstat command and its arguments	See *Displaying Network Statistics* on page 1-8
Test RA 6300 network links	ping command and its arguments	See *Using the ping Command to Test Network Links* on page 1-30
Display RA 6300 statistics	stats command and its arguments	See *Displaying RA 6300 Statistics* on page 1-41

(continued on next page)

Table B-1. Administration Tasks, Commands, and Parameters (continued)

Task	Related CLI Command and/or Parameter	Instructions for Use
Display user activity	who command	See *Displaying User Activity* on page 1-38
Monitor session activity	tap and control commands	See *Monitoring Session Activity* on page 1-48
Log security events	enable_security and syslog_port parameters	See *Logging User and RA 6300 Events* on page 1-33
Display entries in the RA 6300 host table	hosts and reset annex nameserver commands, host_table_size parameter	See *Managing the Host Table* on page 1-49
Display and maintain the Host ARP (Address Resolution Protocol) Table	arp command and its arguments	See *Managing the ARP Table* on page 1-32
Disable RA 6300 Software modules	disabled_modules parameter	See *Disabling Software Modules* on page 1-52
Troubleshoot configuration problems	Commands and parameters based on problem experienced	See *Typical Configuration Problems* on page 1-53

Displaying PRI Information

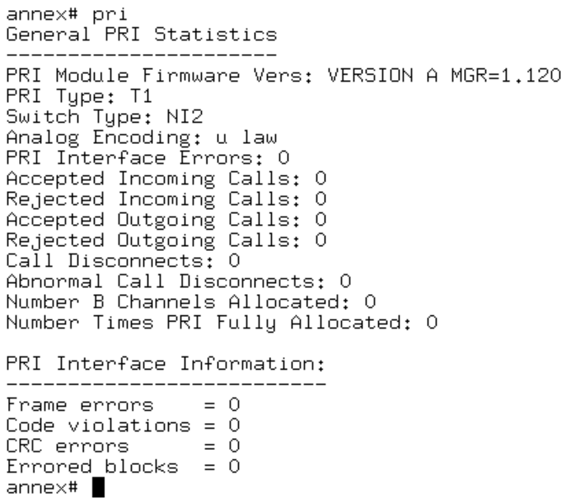
The superuser CLI **pri** command displays information about the RA 6300 PRI module and its connections. Using **pri** you can display information that includes:

- PRI module firmware version information (module version and firmware revision).
- PRI type (T1 or E1).
- Switch Type (Telco switch from which PRI line originates)
- Analog encoding method (mu_law or a_law)
- Number of PRI interface errors.
- Call information.
  - Incoming calls accepted.
  - Incoming calls rejected.
  - Outgoing calls accepted.
  - Outgoing calls rejected.
  - Calls disconnected normally.
  - Calls disconnected abnormally (e.g., interface errors).
  - Number of times B channel capacity reached (23 or 30).
- Other PRI interface information
  - Frame errors
  - Code violations
  - CRC errors
  - Error blocks

pri Command Syntax

```
annex# pri
```

pri Sample Display



```
annex# pri
General PRI Statistics
----------------------
PRI Module Firmware Vers: VERSION A MGR=1.120
PRI Type: T1
Switch Type: NI2
Analog Encoding: u law
PRI Interface Errors: 0
Accepted Incoming Calls: 0
Rejected Incoming Calls: 0
Accepted Outgoing Calls: 0
Rejected Outgoing Calls: 0
Call Disconnects: 0
Abnormal Call Disconnects: 0
Number B Channels Allocated: 0
Number Times PRI Fully Allocated: 0

PRI Interface Information:
--------------------------
Frame errors     = 0
Code violations = 0
CRC errors       = 0
Errored blocks   = 0
annex# █
```

Displaying PRI B Channel Assignments

The superuser CLI **pri b** command displays information about active B channels that includes:

- Call setup information elements received during call establishment that includes: called number, calling number, and bearer.
- Name of the SPB associated with the session operating over the B channel.

- Session duration.
- Assigned port name.

pri b Command Syntax

```
annex# pri b
```

pri b Sample Display

```
annex# pri b
b    called_#       calling_#     br spb_name   time      port
--------------------------------------------------------------
2    0100           6175551212    vo modem      21m 3s    asy8
```

Controlling PRI Calls Made to the RA 6300

The superuser CLI **pri call** command controls whether calls are accepted or denied by the RA 6300.

The command has two options:

- **Allow** which lets the RA 6300 accept calls made to it.
- **Stop** which denies acceptance of new calls, but lets currently established calls to continue.

By default the RA 6300 accepts calls made to it. The RA 6300 reverts to this default following a restart.

These options must be used with one of the following keywords:

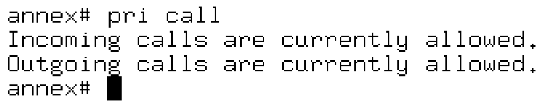
- **Incoming** which are calls made to the RA 6300
- **Outgoing** which are calls made from the RA 6300

The RA 6300 does not currently support outgoing calls. However, the command does display outgoing call information.

pri call Command Syntax

To perform this task...	Issue this command
Learn current status of call acceptance/denial	pri call
Allow incoming calls	pri call allow incoming
Allow outgoing calls	pri call allow outgoing
Stop acceptance of new incoming calls	pri call stop incoming
Stop acceptance of new outgoing calls	pri call stop outgoing

pri call Sample Display



```
annex# pri call
Incoming calls are currently allowed.
Outgoing calls are currently allowed.
annex# █
```

Displaying Active Session Parameter Blocks

The superuser CLI **sessions** command displays the SPBs that are active on an RA 6300.

sessions Command Syntax

To display...	Issue this command
SPB name and setup criteria for all active SPBs	sessions
SPB name, setup criteria, and parameter settings for all active SPBs	sessions -a
SPB name and setup criteria for a single active SPB	sessions -l <spb_name>
SPB name, setup criteria, and parameter settings for a single active SPBs	sessions -al <spb_name>

sessions Sample Display

```
annex# sessions
Session Name: modem
-------------------
       Called Number: 6172380300
              Bearer: voice
         Call Action: modem

Session Name: v120
------------------
       Called Number: 6172380300
              Bearer: data
         Call Action: v120

Session Name: sync
------------------
       Called Number: 6172380301
         Call Action: sync
```

Displaying Network Statistics

The CLI **netstat** command displays information that the RA 6300 has obtained from the network. Using **netstat** you can display:

- Active connections.
- Ethernet statistics.
- PPP statistics.
- SLIP statistics.
- AppleTalk statistics.
- IPX statistics.
- RIP statistics.
- Routing table information.
- Route cache information.
- Dial-out route statistics.
- Rotary information.
- Filtering statistics.
- Memory statistics.
- Protocol statistics.

netstat Command Syntax

annex# **netstat**

Table B-2 describes the display observed using the netstat command with each of its arguments.

Table B-2. Arguments for the netstat Command

Argument	Description
none	Displays default active connection information. Information displayed includes: local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol for all active connections
-A	Adds the protocol control block (PCB) addresses to the default display.
-a	Adds the sockets used by server processes to the default display; can be used in combination with –A.
-C	Displays the contents of the route cache.
-i	Displays the state of the hardware interfaces, e.g., AppleTalk, SLIP, PPP, as well as a dial-out route’s interface name.
-ia *port*	Displays statistics for a specific RA 6300 ARA interface
-ip *port*	Displays the current state of a PPP interface.
-iQ	Displays interface queues.
-iS	Displays the state of the hardware interfaces plus additional information about the SLIP interfaces.
-f	Displays filtering statistics.

(continued on next page)

Table B-2. Arguments for the netstat Command (continued)

Argument	Description
-n	Displays all network addresses as numbers rather than names or symbols; can be used in combination with –A, –a, –i, –r, –t. It displays the IP addresses and TCP ports in decimal notation.
-g	Displays RIP statistics.
-m	Displays statistics for memory buffer allocation.
-R	Displays information about rotaries.
-r	Displays the routing tables, including dial-out routes.
-ra	Displays only Appletalk routes.
-ri	Displays only IP routes
-s	Displays network protocol statistics. LAT statistics display only if the correct lat_key value is set. AppleTalk statistics display only if the correct option_key value is set.
-rs	Displays routing statistics.
-t	Displays the default active connection information along with the attached device name.
-x	Displays information about IPX.
-xi	Displays information about RA 6300s currently in use for dial-in or LAN-to-LAN routing.
-x?	Displays information about using the netstat –x command.

(continued on next page)

Table B-2. Arguments for the netstat Command (continued)

Argument	Description
-xm	Displays information about the amount of memory available in the large and small IPX buffer pools.
-xr	Displays the routes defined in the RA 6300’s IPX routing table.
-xr *network_number*	Displays the RA 6300 route for that network.
-xs	Displays server names, types, and addresses.
-xs *server_name*	Displays information for the specified server (the *server_name* argument is case-sensitive).
-z	Displays the network zone list for AppleTalk.

netstat Sample Displays

The following sections include a representative set of displays presented by the netstat command and its arguments. Not all command and argument combinations are presented.

Active Connections

Entering the **netstat** command without arguments displays the local and remote addresses, send and receive queue sizes (in bytes), protocol, and the internal state of the protocol for all active connections. Table B-3 lists the arguments for this command.

Table B-3. Arguments for the netstat Command

Argument	Description
–A	Adds the protocol control block (PCB) addresses.
–a	Includes sockets used by server processes; can be used in combination with –A.

For example, the **netstat –a** command display looks like this:

```
annex01# netstat –a

Active connections (including servers)
Proto Recv-Q Send-Q Local Address Foreign     (state)
                                  Address
tcp   0      2      annex1.telnet test1.4759  ESTABLISHED
tcp   0      0      annex1.883    gibbs.login ESTABLISHED
tcp   0      0      annex1.1085   ale.telnet  ESTABLISHED
tcp   0      0      annex1.1081   opus.telnet ESTABLISHED
tcp   0      0      annex1.1022   test1.login ESTABLISHED
tcp   211    0      annex1.953    xzyx.login  ESTABLISHED
tcp   0      0      annex1.1021   test1.login ESTABLISHED
tcp   0      0      *.finger      *.*         ESTABLISHED
tcp   0      0      *.printer     *.*         ESTABLISHED
tcp   0      0      *.telnet      *.*         LISTEN
udp   0      0      *.biotope     *.*
udp   0      0      *.snap        *.*
udp   0      0      *.who         *.*
udp   0      0      *.erupt       *.*
udp   0      0      *.route       *.*
```

Interface Statistics

The **netstat –i** command displays interface statistics for an RA 6300 running on an Ethernet LAN. Table B-4 describes the hardware interface statistics for Ethernet.

Table B-4. Hardware Interface Statistics for Ethernet)

Statistic	Description
Frames Received	The number of packets received from the network interface.
Frames Transmitted	The number of packets transmitted on the network interface.
Bytes Received	The number of bytes received from the network interface.
Bytes Transmitted	The number of bytes transmitted on the network interface.
CRC Errors	The number of frames received from the network interface with a bad CRC.
Alignment Errors	The number of frames received from the network interface that were both misaligned and have a CRC error.
Bad Type/ Length Fields	The number of frames received from the network interface that have an unrecognized type field (ethernet) or an illegal length field (802.3).
Buffer Drops	The number of frames received from the network interface that were good, but dropped because no buffers were available.
FIFO Drops	The number of frames received from the network interface that were lost since the local system bus was not available.
Interface Resets	The number of times the network interface has been initialized from reset; typically, one.
TX DMA Underruns	The number of times a frame transmission is terminated due to lack of data.

(continued on next page)

Table B-4. Hardware Interface Statistics for Ethernet) (continued)

Statistic	Description
RX DMA Overruns	The number of times a frame reception is terminated due to lack of system bus bandwidth.
Carrier Sense Losses	The number of times a frame transmission is terminated due to loss of the Carrier Sense signal. The transceiver cable may have a short or an open.
Clear to Send Losses	The number of times a frame transmission is terminated due to loss of the Clear to Send signal.
Collisions Detected	The number of times a frame transmission is terminated due to a collision.
Max Collision Retries	The number times consecutive collisions for a frame exceed the maximum collision retry limit.

The **netstat –i** command display looks like this:

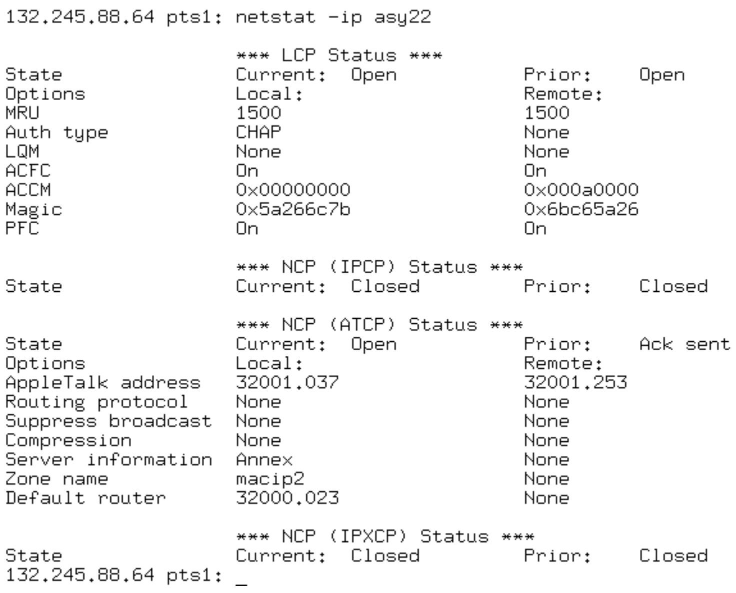
```
annex01# netstat –i

Name   Mtu   Network      Address    Ipkts  Ierrs  Opkts   Oerrs  Collis
en0    1500  132.245.66.0 worm       26563  0      15085   744    0
en0    1500  10000–20000  18062.79   1626   0      823     0      0
lo0    1536  127          127.0.0.1  0      0      0       0      0
asy2   604   18358        18062.79   0      0      0       0      0
asy16  1006  132.245.6    annex01    14770  0      7468    0      0
asy3   1500  192.9.200    zipwad     3453   0      3002    0      0

                    *** Hardware Interface Statistics ***

Ethernet Address:      00-80-2d-00-00-9b
Frames Received:       39861             Frames Transmitted:   45239
Bytes Received:        33965470          Bytes Transmitted:    29453
CRC Errors:            2                 Alignment Errors:     10
Bad Type/Length Fields:6                 Buffer Drops:         0
FIFO Drops:            1                 Interface Resets:     1
TX DMA Underruns:      241               RX DMA Overruns:      0
Carrier Sense Losses:  451               Clear to Send Losses: 0
Collisions Detected:   17526             Max Collision Retries:125
```

PPP Statistics

The **netstat –ip** *port-number* command displays a summary of a PPP interface and its current state. Table B-5 describes the fields in the **netstat –ip** command display.

The **netstat –ip** command display for an **asy** port (asynchronous port) on the RA 6300 looks like this.

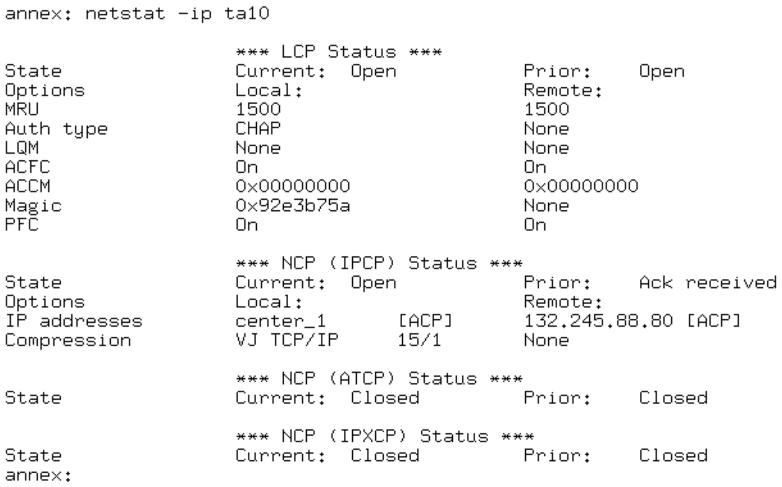
```
annex# netstat -ip asy22
```



```
132.245.88.64 pts1: netstat -ip asy22

                    *** LCP Status ***
State               Current:  Open           Prior:    Open
Options             Local:                   Remote:
MRU                 1500                     1500
Auth type           CHAP                     None
LQM                 None                     None
ACFC                On                       On
ACCM                0x00000000               0x000a0000
Magic               0x5a266c7b               0x6bc65a26
PFC                 On                       On

                    *** NCP (IPCP) Status ***
State               Current:  Closed         Prior:    Closed

                    *** NCP (ATCP) Status ***
State               Current:  Open           Prior:    Ack sent
Options             Local:                   Remote:
AppleTalk address   32001.037                32001.253
Routing protocol    None                     None
Suppress broadcast  None                     None
Compression         None                     None
Server information  Annex                    None
Zone name           macip2                   None
Default router      32000.023                None

                    *** NCP (IPXCP) Status ***
State               Current:  Closed         Prior:    Closed
132.245.88.64 pts1: _
```

The **netstat –ip** command display for a **ta** port (terminal adapter/V.120 port) on the RA 6300 looks like this:

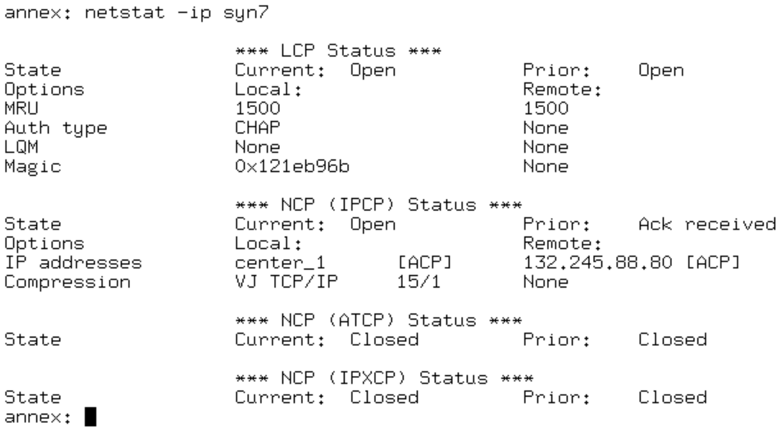
```
annex# netstat -ip ta10
```



```
annex: netstat -ip ta10

                    *** LCP Status ***
State               Current:  Open           Prior:    Open
Options             Local:                   Remote:
MRU                 1500                     1500
Auth type           CHAP                     None
LQM                 None                     None
ACFC                On                       On
ACCM                0x00000000               0x00000000
Magic               0x92e3b75a               None
PFC                 On                       On

                    *** NCP (IPCP) Status ***
State               Current:  Open           Prior:    Ack received
Options             Local:                   Remote:
IP addresses        center_1      [ACP]      132.245.88.80 [ACP]
Compression         VJ TCP/IP     15/1       None

                    *** NCP (ATCP) Status ***
State               Current:  Closed         Prior:    Closed

                    *** NCP (IPXCP) Status ***
State               Current:  Closed         Prior:    Closed
annex:
```

The **netstat –ip** command display for a **syn** port (synchronous PPP) on the RA 6300 looks like this:

```
annex# netstat -ip syn7
```



```
annex: netstat -ip syn7

                    *** LCP Status ***
State               Current:  Open           Prior:    Open
Options             Local:                   Remote:
MRU                 1500                     1500
Auth type           CHAP                     None
LQM                 None                     None
Magic               0x121eb96b               None

                    *** NCP (IPCP) Status ***
State               Current:  Open           Prior:    Ack received
Options             Local:                   Remote:
IP addresses        center_1      [ACP]      132.245.88.80 [ACP]
Compression         VJ TCP/IP     15/1       None

                    *** NCP (ATCP) Status ***
State               Current:  Closed         Prior:    Closed

                    *** NCP (IPXCP) Status ***
State               Current:  Closed         Prior:    Closed
annex: █
```

If *compression* is set, the values that appear in the display are *Max–slot–id* and *Comp–slot–id*. These values are sub-options of VJ compression.

Table B-5. Field Definitions for the netstat –ip Command

Field	Definition
local	Refers to the RA 6300.
remote	Refers to the peer.
[xxx]	The origin of the value for *ip–addresses*: ANX=param; REM=peer–defined; and ACP=from security server ACP dial–up addresses.
LCP and NCP (IPCP) Options	Shows the current and the prior state of the connection. Any *current* setting other than *Open* indicates the link is not up. The states are:
Closed	The layer has shut down via an administrative or peer request.
Request sent	The RA 6300 has sent a configure request and is waiting for an answer.
ACK received	The RA 6300 has received a configure ACK and is waiting for a configure request.
ACK sent	The RA 6300 received and answered a configure request.
Open	Layer negotiation has completed successfully.
Closing	The link is in process of closing. The RA 6300 has sent a terminate request and is waiting for a terminate ACK.
Security	Shows the states based on the last security messages sent and received; this field appears only in superuser mode.

(continued on next page)

Table B-5. Field Definitions for the netstat –ip Command (continued)

Field	Definition
Possible local states for PAP security:	
Initial	No PAP security has been initiated.
AREQ received	The RA 6300 has received the Authenticate-Request message and currently is processing it.
ANAK sent	The RA 6300 has rejected the peer's Authenticate-Request; the link will be coming down.
AACK sent	The RA 6300 has authenticated the peer.
Possible remote states for PAP security:	
Initial	No PAP security has been initiated.
AREQ sent	The RA 6300 has sent the Authenticate-Request message and is waiting for the response.
ANAK received	The RA 6300's Authenticate-Request has been rejected by the peer; the link will be coming down.
AACK received	The peer has authenticated the RA 6300.
Possible states for CHAP security:	
CHAP AACK Rcvd	CHAP Authentication Acknowledged Received.
CHAP AACK Sent	CHAP Authentication Acknowledged Sent.
CHAP CHAL Rcvd	CHAP Challenge Received.
CHAP CHAL Sent	CHAP Challenge Sent.
CHAP RESP Rcvd	CHAP Response Received.
CHAP RESP Sent	CHAP Response Sent.

SLIP Statistics

The **netstat –iS** command displays SLIP data after the hardware interface statistics:

```
annex01# netstat –iS
```

```
Name   Mtu   Network      Address   Ipkts  Ierrs Opkts  Oerrs  Collis
en0    1500  192.9.200    annex1    648918 0     352845 0      0
lo0    1536  127          127.0.0.1 0      0     0      0      0
asy6   1006  192.9.200    annex1    0      0     0      0      0
asy13  256   192.9.200    annex1    0      0     0      0      0

                    *** Hardware Interface Statistics ***

Ethernet Address:      00-80-2d-00-14-3d
Frames Received:       705482            Frames Transmitted:   35283
Bytes Received:        62425605          Bytes Transmitted:    193578
CRC Errors:            0                 Alignment Errors:     0
Bad Type/Length Fields:0                 Buffer Drops:         0
FIFO Drops:            0                 Interface Resets:     1
TX DMA Underruns:      0                 RX DMA Overruns:      0
Carrier Sense Losses:  0                 Clear to Send Losses: 0
Collisions Detected:   2389              Max Collision Retries:0

         SLIP rcvr:
              intrs 0, loops 0, bytes 0, pkts 0
              bytes/intr 0, bytes/loop 0, bytes/pkt 0
              hiwaters 0, overflows 0, mbuf waits 0, mbuf kicks 0
              overruns 0, ipintrq full 0
              FRAME_ENDs 0, FRAME_ESCs 0, proto errs 0, last proto err 0
         SLIP xmit:
              intrs 13, starts 22, vectors 108, bytes 1874, pkts 13
              FRAME_ENDs 22, FRAME_ESCs 32
              bytes/intr 144, bytes/vec 17, vec/pkt 8, bytes/pkt 144
```

AppleTalk Statistics

The **netstat –i** command displays interface statistics. AppleTalk addresses display as *net.node* in hexadecimal, where *net* is 16 bits and *node* is 8 bits. ARAP interfaces display as *ara* plus the unit number. Table B-6 defines the arguments for displaying AppleTalk statistics.

Table B-6. Displaying AppleTalk Statistics using the netstat Command

Argument	Description
–i	Displays interface statistics.
–ip *port number*	Displays a specific RA 6300 PPP interface (see *PPP Statistics* on page 1-15).
–z	Displays the network zone list.

The **netstat –i** command display looks like this:

```
annex01# netstat –i
```

```
Name  Mtu   Network     Address    Ipkts Ierrs Opkts  Oerrs Collis
en0   1500  132.245.66.0worm       26563 0     15085  744   0
en0   1500  10000–20000 18062.79   1626  0     823    0     0
lo0   1536  127         127.0.0.1 0      0     0      0     0
asy2  604   18358       18062.79   0     0     0      0     0
```

```
           *** Hardware Interface Statistics ***
Ethernet Address:  00-80-2d-00-
                   00-9b
Frames Received:   39861          Frames             452397
                                  Transmitted:
Bytes Received:    33965470       Bytes Transmitted: 2945335
CRC Errors:        2              Alignment Errors:  10
Bad Type/Length    6              Buffer Drops:      0
Fields:
FIFO Drops:        1              Interface Resets:  1
TX DMA Underruns:  241            RX DMA Overruns:   0
Carrier Sense      451            Clear to Send      0
Losses:                           Losses:
Collisions         17526          Max Collision      125
Detected:                         Retries:
```

(continued on next page)

```
       *** IEEE 802.2 Data Link Layer Statistics ***
802.2 packets received:1   802.2 packets sent:      0
ATALK packets sent:    0   AARP packets sent:       0
ATALK packets received:0   AARP packets received:   0
Unknown 802.2 types:   0   Unknown 802.2 SAP’s:     0
Unknown SNAP org codes:0   Unknown SNAP ether types:0
```

RIP Statistics

The **netstat –g** command displays RIP statistics. Table B-7 describes the field definitions for the command display.

The **netstat –g** command display looks like this:

```
annex01# netstat –g
Input packets: 19942, Output packets:   0
Interface triggers:   2, Timer events: 4818   Load trips: 0

Sources:

132.245.33.22:   4661 packets      132.245.33.34:   5632 packets
132.245.33.228:  4822 packets      132.245/33/238:  4816 packets
132.245.33.138:  9                 132.245.33.254:  1 packet

Rooting Changes: 1  Queries received: 0

Intf  Bad    Bad   Trigg.  Recv’d  Sent  Disc’d  Update  Queries
      Pkts   Rtes
en0   0      0     0       19942   0     0       22      4
```

Table B-7. Field Definitions for the netstat –g Command

Field	Definition
Intf	Displays the interface.
Bad Pkts	Displays the number of packets the interface dropped due to invalid format or data.
Bad Rtes	Displays the number of routes the interface dropped due to invalid format or data.

(continued on next page)

Table B-7. Field Definitions for the netstat –g Command (continued)

Field	Definition
Trigg.	Displays the number of triggered updates transmitted over the interface. The RA 6300 sends triggered updates whenever it changes the hop count of a route. It transmits them immediately, even if it is not yet time for one of the regular update messages to be transmitted.
Rec’d	Displays the number of packets (with or without errors) received over the interface.
Sent	Displays the number of output packets the RA 6300 tried to send over the interface. This number includes packets that were dropped because the RA 6300 ran out of buffers or the link’s output queue was full.
Disc’d	Displays the number of input packets discarded due to protocol errors or restrictions set by configuration parameters (e.g., rip_accept).
Update	Displays the number on lines in the routing table that were modified due to packets received on that interface.
Queries	Displays the number of routing-table queries received on the interface.

Routing Table Information

The **netstat –r** command displays statistics and information about all available routes in the kernel routing table, including dial-out routes; dynamic dialing routes that do not have a phone connection established appear with a *w* at the end of the route entry. Table B-8 lists the **netstat** command arguments that display routing information. Table B-9 describes the field definitions for the **netstat –r** command display.

Table B-8. Displaying Routing Table Information using the netstat Command

Field	Definition
–r	Displays statistics and information about all available routes in the routing table. A route comprises a destination host or network and the gateway through which data is forwarded. If the dial-out route currently is not active, only *do<route number>* appears in the *Interface* field. If the route currently is active, *asy<port number>* appears in the *Interface* field.
–ra	Displays only AppleTalk routes.
–ri	Displays only IP routes.

The **netstat –r** command display looks like this:

```
annex: netstat –r
Routing tables
Destination      NextHop       Flags  Usage  UseCount Mtr  Interface
4400 - 4499      4475.129      UHF    1      3        0    en0
Apple default    4400.22       UGF    0      0        0    en0
IP default       132.245.44.22 US     +0     0        2    en0
127.0.0.0/8      *             UI     fixed  0        2    lo0
132.245.1.0/24   132.245.44.22 UR     -114   0        3    en0
132.245.2.0/24   132.245.44.22 UR     -114   0        2    en0
132.245.9.0/24   132.245.44.22 UR     -78    36       2    en0
132.245.10.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.11.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.12.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.22.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.33.0/24  132.245.44.22 UR     +33    147      2    en0
132.245.34.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.44.0/24  *             UI     fixed  8382     1    en0
bermuda          132.245.44.22 USH    -114   0        2    en0
132.245.66.0/24  132.245.44.22 UR     -114   0        2    en0
132.245.77.0/24  132.245.44.22 UR     -114   0        2    en0
```

Table B-9. IP Fields in the netstat –r Command Display

Field	Explanation
Destination	The IP address of the route's destination, followed by a slash (/), followed by the number of 1 bits, counting from left to right, in the Destination's subnet mask. For example, the */24* following the IP address *132.254.1.0* indicates a subnet mask of 24 bits (eight octets), or 255.255.255.0. If *IP Default* appears in the Destination field, the entry specifies the route the RA 6300 uses if it can find no other route for a destination. If a name appears in the Destination field, the entry is for a host route; name servers do not have names for network routes. (However, the RA 6300 does not always know a host's name.)
NextHop	The next router to which packets with the given Destination are sent. If the Destination is a local interface, this field displays an asterisk (*); interface routes have no next hop.
Flags	The following three flags:
First flag (Status)	
U	The route is valid (up) and in use.
Q	The route is valid but the interface is quiescent, i.e., the interface is not up yet or was brought down by expiration of the timer set by the **net_activity** port parameter.
D	The route is invalid (down) and has a metric of 16 (RIP infinity). It will stay in the routing table for two more minutes so that other routers can learn that it is invalid.

(continued on next page)

Table B-9. Fields in the netstat –r Command Display (continued)

Field	Explanation
Second flag (Source)	
C	The route was learned via an ICMP redirect. This can occur only when IP routing is disabled (by setting the **routed** parameter to **N**).
I	The route is an interface route.
R	The route was learned via RIP.
S	The route is a static route, learned from a route defined in the **gateway** section of the RA 6300 configuration file or a route entered using the CLI superuser **route** command.
Third flag	
H	The route is a hardwired static route.
Usage	A positive or negative integer indicating a route’s usage. When RIP adds a route to the routing table, it sets its usage value to 0. Every time the route is used RIP adds one to the value; every thirty seconds RIP subtracts one from the value. When the routing table reaches its maximum size of 256 entries, RIP removes the route with the lowest usage value. If there is a tie, RIP removes the first route listed. The values range from -9999999, for a route that has not been used in 9.5 years, to +9999999, for a very frequently used route. Interface, hardwired, and *extremely* frequently used routes contain the word *fixed* in this field instead of a number.
UseCount	A positive integer indicating the number of times the route has been used to transmit a packet. If you subtract the value in this field from the value of *Usage*, you can determine how long a route has been in the routing table.
Mtr	The metric for the route.
Interface	The interface over which the RA 6300 can reach the next hop.

Route Cache Information

The **netstat –C** command displays the contents of the cache route, including both static routes added from the **gateways** section of the configuration file and routes added by the **route** command. Table B-10 describes the flags for the command display.

Table B-10. Flag Descriptions for the netstat –C Command

Flag	Definition
intf *x*	An interface route, where *x* is the interface name and number, e.g., asy8. This can be a back-up route for a an interface that has a duplicate definition in the routing table. For example, if you define a subnet mask for a Proxy-ARP serial interface, and that mask is the same as the RA 6300’s en0 subnet mask, the routes to that interface will be considered duplicates. As a result, the RA 6300 will store the en0 interface route in the routing table and the serial interface route in the cache, thus making the serial interface unreachable. The example below shows a dial-out route, *do67*.
hardwired	Route added either by the **route –h** command or a route defined as ***hardwired*** in the **gateway** section of the RA 6300 configuration file.

The **netstat –C** command display looks like this:

```
annex01# netstat -C

Destination      Subnet Mask     Gateway         Metric  Flags
default          0.0.0.0         132.245.33.22    1
74.68.67.0       255.255.255.0   0.0.0.0          1      intf do67
132.245.124.0                    132.245.71.72    2      hardwired
```

Filtering Statistics

The **netstat –f** command displays filtering statistics. The statistics are cumulative for the Ethernet ports, i.e., changing filters does not reset the counters. The counters for a SLIP and PPP line reset each time the connection resets. Table B-11 describes the field definitions for the command display.

The **netstat –f** command display looks like this:

```
annex01# netstat –f

Int    In-hits    Out-hits    Drop       ICMP      Syslog
en0    0          0           0          0         0
asy1   0          0           0          0         0
asy2   0          0                      0         0
```

Table B-11. Field Definitions for the netstat –f Command

Field	Definition
Int	Displays the interface.
In-hits	Displays the number of packets that matched an input filter.
Out-hits	Displays the number of packets that matched an output filter.
Drop	Displays the number of discarded filtered packets.
ICMP	Displays the number of filtered packets that sent an ICMP message.
Syslog	Displays the number of filtered packets that were syslogged.

Memory Statistics

The **netstat –m** command displays statistics for the memory management routines:

```
annex01# netstat –m
1127/3599 mbufs in use:
        7 mbufs allocated to data
        2 mbufs allocated to packet headers
        9 mbufs allocated to socket structures
        14 mbufs allocated to protocol control blocks
        3 mbufs allocated to routing table entries
        2 mbufs allocated to socket name
        2 mbufs allocated to interface address
          64 mbufs allocated to incoming network i/f packets
        1024 mbufs allocated to SPD Layer RX Data/Status
899 Kbytes allocated to network (31% in use)
0 requests for memory denied
```

Protocol Statistics

The **netstat –s** command displays statistics for the following protocols: ICMP, UDP, TCP, IP, TMux, LAT, and DDP. The LAT statistics display only if the correct **lat_key** value is set; TMux statistics display only if the **tmux_enable** parameter is set to **Y**; DDP statistics display only if the correct **option_key** value is set. A truncated view looks something like this:

```
annex01# netstat –s

    tcp:
     3097 data packets sent
     394865 packets sent
                  309577 data packets (1011910 bytes)
                  87 data packets (22401 bytes) retransmitted
    udp:
     0 incomplete headers
     0 bad data length fields
     0 bad checksums
     2755 no listening port
     77148 packets received
     956 packets sent
```

```
ip:
 613422 total packets received
 0 bad header checksums
 4 output packets we did frag
 5 output fragments we created
icmp:
 2359 calls to icmp_error
 0 errors not generated ’cuz old message too short
 0 errors not generated ’cuz old message was icmp
 Output histogram:
       destination unreachable: 2358
 Input histogram:
       echo reply: 41
tmux:
 65 packets from upper levels
       0 TMUX packets sent
       0 not suitable to TMUX
       0 dropped by TMUX
       65 not able to TMUX
 0 packets from IP
       0 encapsulated packets received
       0 TMUX checksum fails
       0 TMUX other fails
 1 TMUX ENQ packets sent
lat:
 241 Total run messages received
 228 Total run messages transmit
 56382 Total service messages recv.
 3796 Total service messages used
ddp:
 0 short header packets received
 13838 long header packets received
 12120 no checksum
 1 packet too short
 5 not enough data
 13671 packets forwarded
 0 packets encapsulated
```

Using the ping Command to Test Network Links

The superuser CLI **ping** command tests and measures the LAN. Also, it can isolate a single-point hardware or software failure.

The **ping** command sends out an Internet Control Message Protocol (ICMP) echo request packet each second, or until input from the terminal terminates the command.

After completing, **ping** displays a summary of all echo replies received. This display includes a calculation of the time, in milliseconds, that it takes to return the message (if the number of data bytes is 8 or greater).

ping Command Syntax

ping [–**artv**] *host* [*databytes* [*count*]]

Table B-12 lists the arguments for this command.

Table B-12. Arguments for the ping Command

Argument	Description
–a	Generates AppleTalk Echo Protocol (AEP) echo request packets to a target node.
–r	Bypasses the normal routing tables and sends the message directly to a host on an attached network. An error returns if the host is not on a directly attached network.

(continued on next page)

Table B-12. Arguments for the ping Command (continued)

Argument	Description
–t	Traces the path of a packet from the local host to the destination host and back, displaying information about each router in the path. This option allows you to see whether a packet arrived at and/or returned from its remote destination and, if not, where it stopped. The option is based on the Traceroute facility described in RFC 1393. You can use **–t** with the **–r** and/or **–v** argument(s), but not with **–a**.
–v	Displays the IP and ICMP packet headers for the reply from the host.
host	The host, router, or RA 6300 to which the **ping** is sent.
databytes	The number of bytes of data in the ICMP Echo Request message. The default is **56**.

ping Sample Display

The **ping** command display looks like this:

```
annex01# ping caddy
PING caddy: 56 data bytes
64 bytes from 132.245.6.25: icmp_seq=0. time=37. ms
64 bytes from 132.245.6.25: icmp_seq=1. time=12. ms
64 bytes from 132.245.6.25: icmp_seq=2. time=12. ms
64 bytes from 132.245.6.25: icmp_seq=3. time=12. ms
----caddy PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round-trip (ms) min/avg/max = 12/20/37
```

Managing the ARP Table

The Address Resolution Protocol (ARP) maps Internet addresses to hardware addresses. Hosts implementing ARP maintain a translation table for these address mappings. When an RA 6300 receives a request for a host that does not have a translation entry in the ARP table, it broadcasts for the hardware address. The superuser CLI **arp** command displays and modifies entries in this translation table.

Since the RA 6300 automatically builds the ARP table dynamically, you rarely need to modify the table. You can use **arp** to modify the table for hosts that do not implement ARP, enabling communications between the host and the RA 6300. Using **arp**, you can delete a specified entry and/or create an entry for a host.quit

A created entry is permanent unless it is defined as temporary, in which case the entry is deleted after 20 minutes. An entry defined as published causes the RA 6300 to respond with its hardware address for the specified host, even though the IP address is not the RA 6300's. Publishing a hardware address for another host frequently is done to route data to a host connected to the RA 6300 through a SLIP or PPP link.

Monitoring RA 6300 Activity

The CLI commands assist in monitoring RA 6300 activities (see *Using the CLI Commands* on page C-111 for more information). These activities include:

- Logging user and RA 6300 activities.
- Displaying user activity.
- Displaying RA 6300 statistics.
- Monitoring port activity.

Logging User and RA 6300 Events

The RA 6300 provides two mechanisms for logging events: host-based security and a 4.3BSD-style **syslog** daemon (see *Using RA 6300 Security* on page A-211 for details on host-based security and ACP).

Host-based Security Logging

Host-based security provides logging capabilities that maintain audit trails of user activity. The security server logs each event as a message to its ACP log file. Security logging is enabled automatically when host-based security is enabled for an RA 6300 (using the parameter **enable_security**).

Events are logged to the security server that responded to the security request, either granting or denying access requests. When using back-up security servers, the ACP log file is located on each server.

To change the name and/or format of theACP log file, see *Modifying the Supplied Security Application* on page A-325.

Each logged message in the ACP log file can contain any of the following fields:

- *IP address of the RA 6300.*
- *Sequence number.*
- *Port number.*
- *Date.*
- *Time.*
- *Module.*
- *Event.*
- *Packets in.*
- *Packets out.*

- *Bytes in.*
- *Bytes out.*
- *Protocol-dependent information*
- *Username*.

For the RA 6300, the following additional fields are added:

- *Calling number.*
- *Called number.*
- *Called subaddress.*
- *Bearer.*

All fields are separated by colons (:) and are encoded for use by UNIX utilities that sort, merge, select, or filter streams.

When more than one host functions as a security server, the log files can be merged and sorted by the date and time fields.

Following is a sample from a log file that pertains to PRI calls:

```
132.245.11.14:210201c5:#24:960603:104555:PRI manager:call
accept:clg# 6175552536: cld# 0300: cldsa : br voice

132.245.11.14:210201c6:#24:960603:104649:cli hook:login:smith

132.245.11.14:210201ca:#24:960603:105032:cli hook:logout:smith

132.245.11.14:210201cb:#24:960603:105032:cli
hook:acct:0:0:72:2978:smith

132.245.11.14:210201cc:#24:960603:105032:PRI manager:call
disconnect:clg# 6175552536: cld# 0300: cldsa : br voice
```

Element descriptions for the first entry in the above sample are the following:

Entry Element.	Description
132.245.11.14	This is the IP address of the RA 6300 logging the entry.
210201c5	This is the entry number in the log file
#24	This is the RA 6300 internal port number. The # symbol indicates that the port type is asynchronous, the letter t indicates that the port type is V.120, and the letter s indicates that the port type is synchronous. The number 24 indicates that this is twenty fourth port configured as an asynchronous port
960603	This is the date by year (96), month (06), and day (03) (i.e., June 3, 1996)
104555	This is the time by hour (10), minutes (45), and seconds (55) (i.e., 10:45:55)
PRI Manager	This is the name of the software module performing the logging to ACP
call accept	This indicates what actio is being taken by the logging element
clg#	This us the calling number (In this sample, 617-555-2536
cld#	This is the called number
cldsa	This is the called number
br	This indicates whether the bearer is voice or data

Other element descriptions for the fourth entry in the above sample are the following:

Entry Element.	Description
0	Packets in
0	Packets out
72	Bytes in
2978	Bytes out

Events are written continuously to the ACP log file. To prevent this file from overwhelming the file system on the hosts, and still obtain the record information for generating reports, move and compress the file at regular intervals. The size of your network, the number of RA 6300s, and the amount of activity generated at each RA 6300 determines the frequency for moving and compressing the file.

Events written while using ARA or the dial-back security feature have their own messages:

- **bad access code**

  Users entered an unidentified access code for the defined username – the login was terminated.
- **call-back**

  Users logged in with a known username and access code – the RA 6300 calls back a pre-defined phone number (this log can be generated by any of the features that perform a call-back, including dial-back security and ARA); possible status values are **request**, **ok**, **no answer**, or **no device**.

Including the call-back message, the new messages generated by ACP while processing ARA logins are:

- **login**

  User is authenticated and session is started.
- **logout**

  Session exited via user hang-up, time-out, or administrator **reset**.
- **reject**

  Authentication failed.

Event Logging Using syslog

The RA 6300 can log events for a system running a 4.3BSD-style **syslog** daemon or syslog to a port on the RA 6300. The RA 6300 parameter **syslog_port** defines the port to which logged messages are sent (for more details, see *Using Event Logging* on page A-40. The logged message includes:

- The date and time of the event.
- The name or IP address of the RA 6300 on which the event occurred.
- The name of the event and PID of the RA 6300 process.
- A description of the event.

In the following example, on May 5, at 9:19 a.m., a user named *Worth* on port 8 of *annex01* issued the **rlogin** command to host *galago*.

```
May 5 9:19:03 annex01 cli[598]:Job-Begin:8:rlogin
galago:Worth
```

The information display differs, depending on the event. In the following example of a typical message, a time server updates the RA 6300’s time. The time server host’s address displays in hexadecimal longword. Times are expressed in hexadecimal as the number of seconds since 00:00:00 January 1, 1970.

```
Jan 5 9:56:5 annex timed[38]:adjusting time from host
      5fc809c0: old=25bf1398, new=25bf1399, delta=1
```

The next example shows a user on port 9 of *annex* issuing the **telnet** command to access another RA 6300 or other Remote Annex.

```
May 5 8:56:3 annex telnet_cmd[35]:Telnet-Begin:9:telnet
annex1
```

Displaying User Activity

When the CLI **who** command is issued for an RA 6300, it displays the user name, the jobs the user is running, when the connection began, any idle time, and the source of the connection. This command also displays current users on other RA 6300s, and on other hosts, if those hosts have **fingerd** running for **who** *user@host*.

The **who** command display looks like this:

```
annex01# who

Port   What  User   Location     When    Idle Address
pts1   CLI   bob    Ext 528      8:44am       [local]
pts2   CLI   ---    ---          9:02am       [local]
pts4   LPD   ---    ---          9:45am       oaxaca
pts6   ARAP  cobb   P-01-03-con  9:59am       [local]
pts16  PSVR  cody   lpq port     10:00am :43  support
syn1   CLI   ellis  Ext 632      10:00am :41  192.9.200.133
syn2   CLI   carey  ---          10:43am      192.9.200.60
```

When the command is issued for a 4.3BSD host, the display is the same as for the **finger** command executed at the host. Using the **who** command, you can obtain a significant amount of information on users and their activities in the network. For example:

- All users connecting to or from a specific host(s).
- A single user or a group of users connected to the RA 6300.
- All users connected to specific port or virtual CLI.
- A specific user (**who** *user@host*) or all users (**who** *@host*) logged into a specific host.

Using abbreviations, you can display a range of hosts or user names.

Displaying Internal Modem Information

The superuser CLI **modem** command displays information about the RA 6300 internal modems. Using the **modem** command and its arguments you can display status information that includes:

- Type of modem.
- Whether a modem is available, in-use, busied-out, or failed.
- Make a failed modem usable again

modem Command Syntax

modem[–**aum** <*number-range*>]

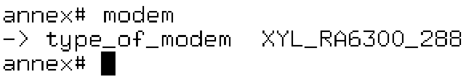
Table B-13 describes the function the modem command with and without its arguments.

Table B-13. Arguments for the modem Command

Argument	Description
none	Displays a information that identifys the type of modems internal to the RA 6300.
-a	Displays the configuration settings for the internal modems.
-u <*modem-range*>	Makes failed modems available. The command has no effect on any modems available, busied-out, or in use.
-m <*modem-range*>	Displays information about each modem specified in the <*number-range*> depending on its current state. If <*number-range*> is not specified, then information about all installed modems displays. The following status information is displayed about each modem. (Only one of the following states displays for each modem) Available In-use Busied-out Failed

modem Sample Display

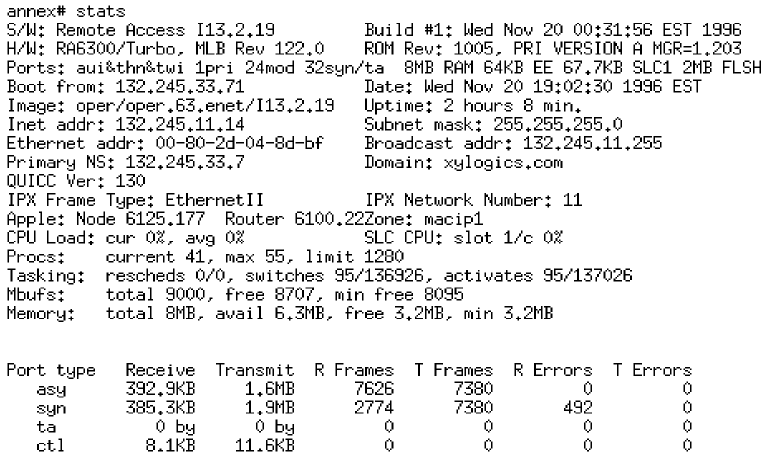
The **modem** command display looks like this:



```
annex# modem
-> type_of_modem  XYL_RA6300_288
annex#
```

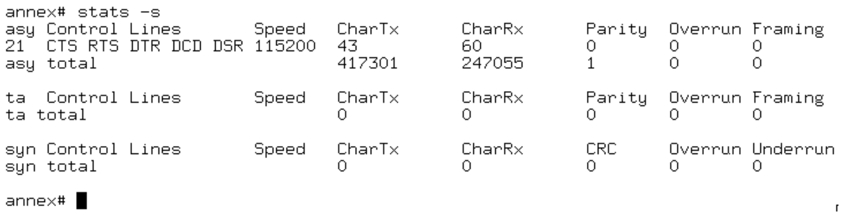
Displaying RA 6300 Statistics

The CLI **stats** command displays general RA 6300 statistics, or statistics for one or more port types. It can also display statistics for the internal CSU (if present). A typical **stats** command display for an RA 6300 on an Ethernet network looks like this:



```
annex# stats
S/W: Remote Access I13.2.19         Build #1: Wed Nov 20 00:31:56 EST 1996
H/W: RA6300/Turbo, MLB Rev 122.0    ROM Rev: 1005, PRI VERSION A MGR=1.203
Ports: aui&thn&twi 1pri 24mod 32syn/ta  8MB RAM 64KB EE 67.7KB SLC1 2MB FLSH
Boot from: 132.245.33.71            Date: Wed Nov 20 19:02:30 1996 EST
Image: oper/oper.63.enet/I13.2.19   Uptime: 2 hours 8 min.
Inet addr: 132.245.11.14            Subnet mask: 255.255.255.0
Ethernet addr: 00-80-2d-04-8d-bf    Broadcast addr: 132.245.11.255
Primary NS: 132.245.33.7            Domain: xylogics.com
QUICC Ver: 130
IPX Frame Type: EthernetII          IPX Network Number: 11
Apple: Node 6125.177  Router 6100.22Zone: macip1
CPU Load: cur 0%, avg 0%            SLC CPU: slot 1/c 0%
Procs:    current 41, max 55, limit 1280
Tasking:  rescheds 0/0, switches 95/136926, activates 95/137026
Mbufs:    total 9000, free 8707, min free 8095
Memory:   total 8MB, avail 6.3MB, free 3.2MB, min 3.2MB


Port type   Receive  Transmit  R Frames  T Frames  R Errors  T Errors
   asy      392.9KB     1.6MB      7626      7380         0         0
   syn      385.3KB     1.9MB      2774      7380       492         0
   ta          0 by      0 by         0         0         0         0
   ctl        8.1KB    11.6KB         0         0         0         0
```

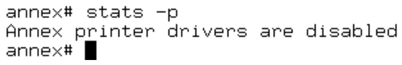
The **stats –s** command displays statistics for all port types:



```
annex# stats -s
asy Control Lines       Speed   CharTx      CharRx      Parity  Overrun Framing
21  CTS RTS DTR DCD DSR 115200  43          60          0       0       0
asy total                       417301      247055      1       0       0

ta  Control Lines       Speed   CharTx      CharRx      Parity  Overrun Framing
ta total                        0           0           0       0       0

syn Control Lines       Speed   CharTx      CharRx      CRC     Overrun Underrun
syn total                       0           0           0       0       0

annex# █
```

The **stats –p** command displays statistics for all parallel ports:

The RA 6300 does not support parallel ports.



```
annex# stats -p
Annex printer drivers are disabled
annex# █
```

The superuser **stats –c** command clears all serial line statistics to zero.

The **stats –m** command displays statistics for active control lines, but displays the modem controls for inactive control lines rather than displaying *idle*.

The **stats –m** <*number_range*>command displays statistics for asynchronous ports in specified number range. For example, specifying a range of 1 through 8 (**stats -sta1-8**) displays statistics for the first eight synchronous (asy) ports.

The **stats –sta**<*number_range*> displays statistics for terminal adapter (ta) ports in the specified number range. For example, specifying a range of 1 through 8 (**stats -sta1-8**) displays statistics for the first eight terminal adapter (ta) ports.

The **stats –ssyn**<*number_range*> displays statistics for synchronous (syn) ports in the number range. For example, specifying a range of 1 through 4 (**stats -ssyn1-4**) displays statistics for the first four synchronous ports.

For RA 6300s that support internal CSUs, the **stats –T** command displays an extensive set of T1/PRI line statistics, alarm indications, and other line performance monitoring information. This information is obtained through the RA 6300 implementation of the Facilities Data Link (FDL) protocol and from the PRI module. Table B-14 describes the arguments for **stats –T**. The syntax is as follows:

annex: **stats –T** {**clear**|**current**|**total**|**all**}

Table B-14. Arguments for the stats –T Command

Argument	Description
current	Displays T1/PRI statistics information for the current 15-minute interval.
total	Displays a summary of T1/PRI statistics information for the last 24 hours. This option is available only if the **pri** *fdl_type* parameter is set to **att** (see Table A-4 on page A-53).
all	Displays the T1/PRI statistics for each valid interval. There are up to 96 intervals (15 minutes per interval) for a 24-hour period. The most recently recorded interval is displayed as the highest-numbered one. This option is available only if the **pri** *fdl_type* parameter is set to **att** (see Table A-4 on page A-53).
clear	Clears the alarm condition saved in the history buffer. Also clears the previous 24-hour statistics. Once the alarm history buffer is cleared, the next alarm event is captured and stored in the history buffer. Subsequent alarm events will not overwrite the history buffer. This option is available only to the Superuser.

Table B-15 describes the fields that **stats –T** displays. The following is a sample display:

```
annex: stats -T current

Alarm History:
        Thu Oct 10 12:29:33 1996 EDT
        NO SYNC
Current Alarms:
        RED
        NO SYNC
        LOSS OF SIGNAL
Circuit ID:
T1 info:
Loopback mode: no loopback

Current Statistics:
        Fri Oct 11 10:08:31 1996 EST
        Number of valid seconds: 124
        CRC6 Error Event: 0
        Out of Frame: 1
        ESF Error Event: 2
        Errored Seconds: 104
        Severely Errored Seconds: 104
        Unavailable Seconds: 104
        Bursty Errored Seconds: 0
        Loss of Frame Count: 1
        Controlled Slip Seconds: 0
        Unavailable Signal State: Declare
```

Table B-15. Fields in the stats –T Command Display

Field	Description
Alarm History	Displays the first alarm that occurred after a boot or a **stats -T clear** command, preceded by the time at which the alarm occurred.
Current Alarms	When *NO SYNC* is displayed, the T1/PRI engine has lost frame synchronization and enters the Red Alarm state. When nothing is displayed for Sync, the T1/PRI engine is detecting frame synchronization.
	When *LOSS OF SIGNAL* is displayed, the T1/PRI engine is not detecting pulses on the T1/PRI network interface receiver. When *LOSS OF SIGNAL* is not displayed, the T1/PRI network interface is receiving pulses. The loss of signal condition causes the T1/PRI engine to transmit AIS (all ones unframed) on the network interface.
	When *BLUE* is displayed, the T1/PRI engine is receiving AIS (all ones unframed) from the network. When *BLUE* is not displayed, the T1/PRI engine is not receiving AIS.
	When *RED* is displayed, the T1/PRI engine has detected loss of frame synchronization. During this condition, the T1/PRI interface is sending a Yellow Alarm to the network. When *RED* is not displayed, the T1/PRI engine is in frame synchronization.
	When *YELLOW* is displayed, the T1/PRI engine is receiving a Yellow Alarm from the network. When *YELLOW* is not displayed, the T1 engine is not receiving a Yellow Alarm from the network.

(continued on next page)

Table B-15. Fields in the stats –T Command Display (Continued)

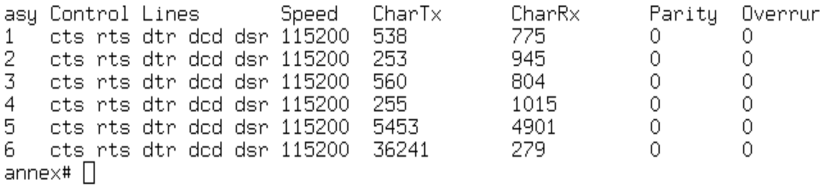
Field	Description
Circuit ID	The T1/PRI engine’s Circuit ID displayed from the **tni_circuit_id** parameter.
Loopback mode	The loopback status has the following possible states: **Loopback** – Loopback is in progress. **No Loopback** – There is no loopback in progress. The test LED on the front panel is not illuminated.
Number of Valid Seconds	Part of the **current** report that indicates the number of seconds for which statistics data has been collected.
CRC6 Error Event	A CRC6 error occurs when the six-bit CRC field calculated by the customer installation, based on the incoming DS1 signal, does not agree with the CRC field contained in the DS1 signal received from the network.
Out of Frame	An Out of Frame (OOF) event begins when any two of four consecutive frame synchronizing bits are received from the network interface. An OOF state ends when reframe occurs.
ESF Error Event	An Extended Superframe CRC error.
Errored Seconds	An Errored Second is a second with one or more ESF error events; that is, one or more CRC error events or one or more Out of Frames (OOFs).

(continued on next page)

Table B-15. Fields in the stats –T Command Display (Continued)

Field	Description
Severely Errored Seconds	A Severely Errored Second is a second with 320 or more CRC error events or one or more OOFs.
Unavailable Seconds	Unavailable Seconds is a count of 1-second intervals during which service is unavailable.
Bursty Errored Seconds	A Bursty Errored Second (BES) is a second with more than one, but less than 320, CRC error events.
Loss of Frame Count	Loss of Frame Count is the number of times that frame synchronization has been lost.
Controlled Slip Seconds	A Controlled Slip Second is a second with one or more Controlled Slips. A Controlled Slip is the occurrence of a replication or deletion of a DS1 frame by the receiving terminal.

For example, the display that reports statistics for the first six asynchronous ports looks like this:



```
asy Control Lines       Speed   CharTx      CharRx      Parity  Overrur
1   cts rts dtr dcd dsr 115200  538         775         0       0
2   cts rts dtr dcd dsr 115200  253         945         0       0
3   cts rts dtr dcd dsr 115200  560         804         0       0
4   cts rts dtr dcd dsr 115200  255         1015        0       0
5   cts rts dtr dcd dsr 115200  5453        4901        0       0
6   cts rts dtr dcd dsr 115200  36241       279         0       0
annex# ▯
```

Monitoring Session Activity

The RA 6300 provides two superuser CLI commands that display information about the state of the RA 6300’s serial ports: **control** and **tap**.

The superuser CLI **control** command is a diagnostic tool that, for a specified port, allows you to set DTR and RTS or output a short test message. The superuser CLI **tap** command accesses (wire taps) a serial port from a terminal.

The **tap** command will not work with PPP.

Using **tap**, you can:

- Observe the output to the port. The command also displays keystrokes entered from your terminal as output to the port you are tapping as if they had been entered on the port.
- Find out exactly what users are seeing on their terminals from a remote location.

- Provide on-line advice and instructions to users at their terminals.
- Monitor traffic in both directions on the port, especially incoming special conditions, such as line breaks and special characters.

Under certain circumstances, the order of displayed data may not match the actual time sequence of the events. All input and output data is displayed. Special characters and control line changes are stored in a limited buffer. If these changes occur too rapidly, they may be lost.

Managing the Host Table

The host table contains this information for each host:

- Host name.
- Aliases (if any).
- IP address.
- Multiple IP addresses (if any).
- System status (if the entry is updated by RWHO).
- Load factor (if the entry is updated by RWHO).
- Number of users (if the entry is updated by RWHO).

The CLI **hosts** command displays all entries in the host table. The RA 6300 can build and update the host table from RWHO messages and from responses to DNS and/or IEN-116 queries. Entries are updated according to information received. Information for a host will be updated if new information received is different from what is currently in the host table. The RA 6300 considers information from a DNS server the most reliable source; it considers an IEN-116 as the next reliable source; and it considers RWHO broadcasts as the least reliable source. Thus, information from a DNS server always updates current information received from either an IEN-116 server or an RWHO broadcast; information from an IEN-116 server always updates current information received from an RWHO broadcast.

The RA 6300 also deletes entries. The criteria for deletion depend on the source of the entry. Each DNS response includes a time to live (TTL). When an entry reaches its full life (default=60 minutes), the DNS server is queried again. If a DNS server recognizes the name, the entry is re-entered in the host table; otherwise, it is deleted. The RA 6300 keeps track of how often each IEN-116 host table entry is referenced. If a name server entry has not been used for 32 days, it is deleted.

The RA 6300 expects to receive an RWHO message from a host at least every six minutes; if no message is received in that time period, the host table status entry for that host is changed to *down?*. If there is no message for 12 minutes, the status is changed to *down*, and if no message is received for 60 minutes, the entry is removed from the table.

If the host table acquires a new entry after it is full, the RA 6300 deletes the oldest, least-used entry to make room for the new one. If the host table is too small, it frequently changes. Increasing the size of the host table using the RA 6300 parameter **host_table_size** reduces these changes.

Other tools for managing the host table are:

- The CLI **hosts –n** and **hosts –f** commands.

  The hosts **–n** command displays name server information; **hosts –f** flushes all, or specified, entries in the host table.

- The **na** or CLI **admin** command **reset annex nameserver**.

  The **reset annex nameserver** command resets all name server parameters discussed in this section and flushes all entries from the host table.

> Flushing the host table and resetting the name server does not remove down-loaded entries from the **gateway** section of the configuration file.
>
> Additionally, the **gateway** section of the configuration file permits a line entry containing a host name that is associated with an IP Address. This entry is identical to the **/etc/hosts** file entry, except aliasing is not supported. When the RA 6300 boots, it adds this host name entry to the host table. Each entry lives in the host table until a nameserver overrides the information or until the administrator resets the nameserver using the **na** or CLI **admin** commands.

Disabling Software Modules

The RA 6300 parameter **disabled_modules** allows you to disable individual software modules to free memory space. If you enter more than one module, separate module names using commas. Valid options are **admin**, **atalk**, **dialout**, **edit**, **fingerd**, **ftpd**, **ipx, lat**, **nameserver**, **ppp**, **slip**, **snmp**, **tn3270**, **tstty**, **vci**, **all**, or **none**. The default is **vci** (disables the RA 6300 VMS interface).

The syntax for disabling several modules is:

```
set annex disabled_modules lat,snmp,ppp,slip
```

You should exercise extreme caution when disabling modules:

- If **disabled_modules** is set to a value other than **none** and **server_capability** includes the operational image, no modules are disabled; a syslog message announces this override.
- The **vci** option disables the RA 6300 interface for VMS environments along with the following commands: **backwards**, **change**, **clear**, **crash**, **define**, **disconnect**, **forwardlis**, **forward**, **list**, **logout**, **resume**, **set**, **show**.
- If **lat_key** is invalid and **server_capability** is set to **none**, the LAT code is freed for use by the system.
- Disabling LAT also disables the CLI commands **services**, **connect**, and **queue**.
- Disabling **admin** and **snmp** can cause problems if host-based **na** is not available. To change parameters in this case, return to monitor mode, erase the parameters in non-volatile memory, and reconfigure the RA 6300.

Typical Configuration Problems

Each RA 6300 hardware platform provides a hardware installation guide that contains troubleshooting information. Many problems that occur after an RA 6300 is running are due to improper configuration of the RA 6300 or a host. The following subsections describe the symptoms of several common configuration problems.

Connection Delays When Using Name Servers

RA 6300 users may notice connection delays under certain circumstances. If **name_server_1** and **name_server_2** are defined, and **name_server_1** is down or does not exist, there will be a 15–30 second delay until **name_server_2** resolves the name during a connect to a host using **rlogin** or **telnet**. If both name servers are down or they do not exist, there will be up to a 45 second delay. If the host to which the user ID is trying to connect is in not in the RWHO host table, an error occurs; the terminal displays a message informing the user that the name server is unreachable.

Hosts not Appearing in Hosts Display

The **hosts** command should list any hosts that broadcast RWHO packets if the configuration parameter **rwhod** is set to **Y**. If you expect to see a host in the **hosts** display and it does not appear, wait several minutes and then re-issue the **hosts** command before assuming there is a problem; the time between broadcasts can vary. Before proceeding, verify that the host not appearing in the **hosts** display is sending RWHO packets correctly by entering **ruptime** on another host on the network, or by checking that the host in question is running **rwhod**.

If the host is sending RWHO packets correctly, incompatible broadcast addresses may be causing the problem. Originally, a broadcast packet used a host address of all *zeros* (*network.0*). Later refinements required a change to the broadcast address, specifying a host address of all *ones* (*network.255*). A host configured with a *network.255* address will accept *network.0* broadcasts. Hosts configured with *network.0* addressing will not see *network.255* broadcasts. You can configure the RA 6300 for either method of addressing by setting the **broadcast_addr** parameter.

Wrong Host Address in Host Table

The RA 6300 assumes that the host described in the data part of the RWHO packet sent the packet, and the IP header's *source-Internet-address* field contains the host's address. Usually, this assumption is correct because routers do not forward broadcast packets. Some RWHO daemons do forward RWHO packets.

You can turn off RWHO at the RA 6300 by setting the **rwho** parameter to **N**. RWHO entries are not added to the RA 6300's host table.

Network Logins to BSD Hosts are Invisible

A user can **rlogin** or **telnet** to a host, but the pseudo-terminal does not show up in a **who** command display. This problem is caused by a mismatch between pseudo-terminals configured in the **/dev** directory and pseudo-terminal entries in **/etc/ttys**. Update the **/etc/ttys** file to contain the proper number of pseudo-terminals as indicated by the actual device entries in **/dev**.

All Network Ports are in Use

The **rlogin** or **telnet** command is rejected after the user name is entered in response to the login prompt. The error message *all network ports in use* indicates that all available pseudo-terminals are in use. On BSD hosts, update **/etc/ttys** and create more pseudo-terminals in **/dev**.

Chapter 2 Simple Network Management Protocol (SNMP)

This chapter describes the Simple Network Management Protocol (SNMP) and the SNMP agent provided by the RA 6300. This chapter includes the following sections:

- *SNMP Protocol Overview.*
- *Setting up the RA 6300 for SNMP.*
- *Standard MIB Support.*
- *Parameters vs. Private Enterprise MIB.*

SNMP Protocol Overview

SNMP is a heavily used management protocol. It operates over the User Datagram Protocol (UDP), which is part of the TCP/IP protocol suite. SNMP provides an easier and more efficient means of managing the RA 6300.

- The SNMP protocol can send queries to the SNMP agents located in each RA 6300.
- Each SNMP agent collects information about its RA 6300 and provides that information to the Network Management Station running the RA 6300. The agent process acts as a server in a typical client-server model.
- Management Information Bases (MIBs) located on the SNMP Network Management Station describe the information that comes from the agents.

SNMP Management Stations

An SNMP Network Management Station is a dedicated or shared network device that is the client in the client-server model. The management station can run an application specifically written for the RA 6300 (or other Remote Annexes) and its MIBs (e.g., Annex Manager), or a generic application that communicates with other non-Bay Networks devices (e.g. SunNet Manager,™, HP/OpenView, ™ NetView for AIX™). The generic application must include the definitions of the MIBs supported by the RA 6300.

The SNMP agent processes **get**, **set**, **get-next** commands, returns a response indicating the command's success or failure, and returns the requested data for the **get** and **get-next** commands (*SNMP Commands* on page 2-62 describes these commands in greater detail).

Message Delivery

SNMP messages are encapsulated in UDP datagrams. The UDP layer does not guarantee delivery. The RA 6300 uses a time-out and retry mechanism to guarantee the SNMP command's delivery. If a time-out occurs, the RA 6300 does not know if the agent did not receive the command or if the agent's response was lost.

The SNMP agent can generate an unsolicited trap command and send it to one or more network addresses. Receivers of traps, i.e., trap hosts, do not respond to the SNMP agent (for more details, see *Defining Trap Hosts and Traps* on page 2-60).

The RA 6300 supports only the cold-start, link-up, and link-down traps defined in MIB-II.

Configuring the RA 6300 for SNMP

Before an SNMP network management application can monitor or manage the RA 6300, you must define certain configuration data, including the SNMP agent and related RA 6300 parameters.

Configuring the SNMP Agent

Entries in the **gateway** section of the configuration file, which is downloaded during RA 6300 initialization, both enable the SNMP agent and define the operating characteristics of the SNMP daemon that controls the SNMP agent.

The **gateway** section of the configuration file contains four optional keywords for configuring the RA 6300 SNMP agent:

- **community**
- **traphost**
- **contact**
- **location**

The following subsections detail each of these keywords as well as the parameters required for use with SNMP.

A sample entry in the **gateway** section of the configuration file looks like this:

```
annex 132.245.6.34
     host 132.245.1.01 gateway 132.245.7 metric 1 hardwired
     net 132.245.9.0 gateway 132.245.2.3 metric 1 hardwired
     snmp contact john smith ext 370
     snmp location computer room
end
snmp community public
snmp traphost 132.245.6.50
```

Defining the Community String

Each SNMP message contains a community string in its header. The receiving SNMP agent tries to match the message's string with an existing community string list. If there is no match, the SNMP agent discards the message without responding to the sender.

The keyword **community** defines an SNMP community name from which the RA 6300 responds to requests. At system start-up, the SNMP agent requires at least one community string to be defined in the configuration file. If the file does not contain a community string, the RA 6300 defaults to the community name *public* (unless SNMP is disabled in the parameter **disabled_modules**). There is no notion of read-only or read-write communities.

You can specify up to four SNMP community names in the **gateway** section of the configuration file, but each community requires a separate line. The RA 6300 adds these communities to the SNMP agent's community table. The syntax is:

snmp community *name*

Defining Trap Hosts and Traps

The RA 6300 employs two methods for defining the host addresses it uses when generating SNMP trap messages.

- The first method defines up to ten static trap hosts in the configuration file using the SNMP trap host syntax.
- The second method loads the trap hosts (if any) from the configuration file into the Trap Host Table (i.e., the anxTrapHostTable objects in the proprietary MIB). You can modify this table by adding or deleting trap hosts. However, the changes you make directly through the table will be lost when the RA 6300 reboots. If you want your changes to be permanent, you must use the configuration file.

Traps are unsolicited administrative messages generated by SNMP agents on the network. The keyword **traphost** defines the host to which SNMP traps are sent. For the RA 6300 to generate traps, one or more trap host addresses must be defined in the **gateway** section of the configuration file along with the SNMP community string. All generated trap messages use the first community string defined in the configuration file (if the file does not contain a community string, the RA 6300 defaults to *public).*

You can specify up to ten static trap hosts in the configuration file, but each host requires a separate line. Specify the trap host using its IP address (RFC 1157 provides more details on communities and traps). Table B-16 describes the supported SNMP traps. The syntax is:

snmp traphost *ipaddr*

Table B-16. Supported SNMP Traps

Trap	Description
coldstart	Upon initialization of the SNMP agent at boot time.
link-up	Upon initialization of each network interface.
link-down	Upon de-configuration of any network interface.

Defining the Contact String

The keyword **contact** defines the object that identifies the person responsible for managing the RA 6300, as supported by MIB-II. The syntax is:

snmp contact *string*

The *string* can include information about how to contact the person; e.g., *M. Law, x 370.*

Defining the Location String

The keyword **location** defines the object that describes the RA 6300's location; e.g., *computer room*. The syntax is:

snmp location *string*

Defining the disabled_modules Parameter

The parameter **disabled_modules** allows you to turn off certain features during software initialization (e.g., enter *LAT, PPP, SLIP* to turn these features off). If you disable SNMP, the RA 6300 will discard all SNMP messages it receives. By default, the SNMP agent on the RA 6300 is enabled (for more details, see *disabled_modules* on page C-51).

Defining the allow_snmp_sets Parameter

The RA 6300's default setting for the **allow_snmp_sets** parameter does not permit parameter value changes because the SNMP **set** command's header transmits the community string in clear text, which may be a security risk. To modify parameters through SNMP, you must first set **allow_snmp_sets** to **yes** using the **na** utility or the **admin** command. You cannot set this parameter using SNMP (for more details, see *arap_v42bis* on page C-41).

SNMP Commands

The SNMP agent software in the RA 6300 supports the SNMP commands **get**, **get-next**, **set**, and **trap** as defined in RFC 1157. Table B-17 describes these commands.

Table B-17. Supported SNMP Commands

Action	Description
get	Retrieves the value of a specific object from one of the supported MIBs.
get-next	Traverses the MIB tree to retrieve the next object's management information.
set	Modifies the values of MIB objects. The RA 6300 private enterprise MIB and several objects in the standard MIBs allow you to configure the RA 6300 from an SNMP management station on the network rather than using the **na** utility or CLI **admin** command.
trap	Asynchronously reports significant events.

When the **allow_snmp_sets** parameter is enabled, the RA 6300 accepts SNMP **set** commands from any source and processes them. When disabled, the RA 6300 rejects all SNMP **set** commands; the RA 6300 SNMP agent returns the error *no such name* for the first object in the **set** command (for more details, see *arap_v42bis* on page C-41).

The specifics of using the SNMP commands depend on the management station (see your SNMP management station documentation). The MIB definitions in the files provided in the directory /*annex_root*/**src/snmp** must be compiled and included in your management station database before you can manage the RA 6300.

Using SNMP set to Send Commands to the RA 6300

The private enterprise MIB objects allow you to change the configuration of the RA 6300. These configuration changes do not take effect until the RA 6300 is rebooted.

Using the SNMP **set** command, you can broadcast a message, reset a port or subsystem, and reboot the RA 6300.

- To broadcast a message, use SNMP **set** to write the message to the MIB object **anxcBcastMsg** and then **set** the broadcast type to the MIB object **anxcBcast**.
- To reset an RA 6300 subsystem, use SNMP **set** to write the desired type (**all**, **macros**, **motd**, **nameserver**, **security**) to the MIB object **anxcReset**.
- To reset the RA 6300, use SNMP **set** to write the desired value to the MIB object **anxcReset**.
- To reset a single serial port, use SNMP **set** to write the appropriate value to the character MIB object **charPortReset** (defined in RFC 1316) that corresponds to the serial port to reset.
- To reboot the RA 6300, **set** the desired image name to the MIB object **anxcBootImage** and **set** any boot warning message to the MIB object **anxcBootMsg**. For a delayed boot, **set** the boot time to the MIB object **anxcBootTime**. Then **set** the boot type to the MIB object **anxcBoot.**

To change the RA 6300's configuration using **set**, SNMP must be enabled at boot time. Make sure the argument **snmp** is not disabled in the **disabled_modules** parameter. For more details, see *disabled_modules* on page C-51).

You cannot configure filters through SNMP.

RA 6300 Standard MIB Support

The RA 6300 supports the following standard MIBs with some restrictions:

MIB	Defined in...	For information on restrictions...
MIB-II	RFC 1213	See Table B-18 on page B-67
AppleTalk MIB	RFC 1243	See Table B-19 on page B-68
Dot3 Ethernet-like Statistics MIB	RFC 1389	See Table B-20 on page B-69
Rip2 MIB	RFC 1389	See Table B-20 on page B-69
DS1 MIB	RFC 1406	See Table B-24 on page B-72

The following standard MIBs are supported as read objects only:

MIB	Defined in...	For information on restrictions...
Character MIB	RFC 1316	See Table B-22 on page B-70
RS232-like MIB	RFC 1317	See Table B-23 on page B-71
Ethernet MIB	RFC 1398	See Table B-21 on page B-70

The Capabilities Statement (filename: **xylo.cap**) in the */annex_root/***src/snmp** directory contains additional information about support for specific MIB objects.

Most parameters do not map to standard MIB objects. Instead, they map to MIB objects in a proprietary (or private enterprise) MIB specific to the RA 6300 and other Remote Annexes. The private MIB also contains objects that provide status and statistics information to the network manager (see *Parameters vs. Private Enterprise MIB* on page 2-73).

This section explains the relationship between the RA 6300 (and other Remote Annexes) and standard MIBs, listing the exceptions and restrictions placed on standard MIBs by the RA 6300 SNMP agent. This section includes:

- *MIB Object Hierarchy*
- *Describing and Naming Objects*
- *Restrictions on Standard MIBs*

MIB Object Hierarchy

MIBs define the hierarchy of managed objects. MIB objects represent data that the RA 6300 can retrieve or configuration information that it can modify.

Describing and Naming Objects

RFC 1155 (*Structure and Identification of Management Information for TCP/IP-based internets*) describes the layout and encoding of exchanged data objects. The SMI (Structure of Management Information) uses the ISO standard ASN.1 (Abstract Syntax Notation One) to define a method for describing a hierarchical name space for managed information.

Each object has:

- A name (also referred to as an Object Identifier (OID)).
- A syntax and an encoding. In addition to the basic integer and octet string data types, several special types are defined (e.g, *IP Address, Network Address, Counter, Gauge, TimeTicks)*. RFC 1212 (Concise MIB Definitions) is an easier-to-read form used in most standard MIBs today. It is used to define the private enterprise MIB.

Restrictions on Standard MIBs

The SNMP Agent does not use all objects in the supported standard MIBs. This section lists the supported standard MIBs and outlines the differences between the Annex parameters and specific standard MIB objects. *RA 6300 Standard MIB Support* on page 2-65 lists the supported standard MIBs.

RFC 1213 MIB-II Restrictions

The RA 6300 supports RFC1213's *system*, *interfaces*, *at*, *ip*, *icmp*, *tcp*, *udp*, and *snmp* groups. It does not support the *egp* group. In addition, some individual objects have the restrictions outlined in Table B-18.

Table B-18. RFC 1213 MIB-II Objects

Object Name	get/set Restrictions	Read Object Limitations
ifAdminStatus	read only	Returns only *up (1)* and *down (2)*
ifOperStatus	none	Returns only *up (1)* and *down (2)*
atEntry	Cannot create new rows	none

(continued on next page)

Table B-18. RFC 1213 MIB-II Objects (continued)

Object Name	get/set Restrictions	Read Object Limitations
ipRouteEntry	Cannot create new rows	none
ipRouteProto	none	Returns only *local (2)*, *icmp (4)*, and *rip (8)*
ipRouteType	none	Returns only *invalid (2)*, *direct (3)*, *indirect (4)*
ipNetToMediaEntry	Cannot create new rows	none
ipNetToMediaType	Writes only *invalid (2)*, *dynamic (3)*, and *static (4)*	Returns only *dynamic(3)* and *static(4)*

RFC 1243 AppleTalk MIB Restrictions

The RA 6300 does not support the *llap*, *rtmp*, *kip*, *zip*, and *nbp* groups. It supports the *aarp*, *atport*, *ddp*, and *atecho* groups with the restrictions listed in Table B-19.

Table B-19. RFC 1243 AppleTalk

Object Name	Restrictions	Read Object Limitations
atportType	Read only	None
atportNetStart	Not supported	None
atportNetEnd	Not supported	None
atportNetAddress	Not supported	None

(continued on next page)

Table B-19. RFC 1243 AppleTalk (continued)

Object Name	Restrictions	Read Object Limitations
atportStatus	Read only	None
atportZone	Read only	None
atportIfIndex	Read only	None
ddpOutRequests	Not supported	None
ddpInLocalDatagrams	Not supported	None
ddpNoProtocolHandlers	Not supported	None
ddpBroadcastErrors	Not supported	None
ddpShortDDPErrors	Not supported	None
ddpHopCountErrors	Not supported	None

RFC 1389 RIPv2 MIB Restrictions

The RA 6300 supports *rip2GlobalGroup*, *rip2IfStatTable*, and *rip2IfConfTable*. It does not support *rip2PeerTable*. Table B-20 describes additional restrictions.

Table B-20. RFC 1389 RIPv2 MIB Objects

Object Name	Restrictions	Read Object Limitations
rip2IfStatStatus	Read only	None
rip2IfConfDomain	Not supported	None
RipIfConfAuthKey	Not supported	None
ripIfConfStatus	Read only	None

RFC 1398 Ethernet MIB Restrictions

The RA 6300 supports RFC 1398’s *dot3StatsTable* and *dot3CollTable* with the restrictions outlined in Table B-21.

Table B-21. RFC 1398 Ethernet MIB Objects

Object Name	Restrictions	Read Object Limitations
dot3StatsSQETestErrors	Not supported	None
dot3StatsInternalMacReceiveErrors	Not supported	None

RFC 1316 Character MIB Restrictions

The RA 6300 supports the *char* group with the restrictions outlined in Table B-22.

Table B-22. RFC 1316 Character MIB Objects

Object Name	Restrictions	Read Object Limitations
charPortAdminStatus	Read only	Returns only *enabled (1), disabled (2), off (3)*
charPortOperStatus	None	Returns only *up (1), down (2), active (5)*
charPortInFlowType	Supports only *none (1), xonXoff (2),* and *hardware (3)*	None
charPortOutFlowType	Supports only *none (1), xonXoff (2),* and *hardware (3)*	None

(continued on next page)

Table B-22. RFC 1316 Character MIB Objects (continued)

Object Name	Restrictions	Read Object Limitations
charPortAdminOrigin	Read only	None
charPortName	Read only	None
charPortSessionMaximum	Maximum value is *16*	None
charSessKill	Read only	None
charSessState	None	Returns only *connected (2)*
charSessConnectionId	None	Returns only null
charPort objects for virtual ports	Read only, read-write objects apply only to physical ports	None

RFC 1317 RS-232 MIB Restrictions

The RA 6300 supports this MIB with the restrictions described in Table B-23.

Table B-23. RFC 1317 RS-232 MIB Objects

Object Name	Restrictions	Read Object Limitations
rs232AsyncPortParity	*none (1)*, *mark (4)*, or *space (5)* all map to *none (1)*	Returns only *none (1)*, *odd (2)*, or *even (3)*
rs232AsyncPortStopBits	*dynamic (4)* maps to *one (1)*	Returns only *one (1)*, *two (2)*, or *one-and-half (3)*
rs232SyncPortTable	Not supported	

RFC 1406 DS1 MIB Restrictions

All DS1 MIB objects necessary to configure the RA 6300 PRI interface are supported, but, for some objects you are limited to setting default values. The RA 6300 supports this MIB with the restrictions described in Table B-24.

Table B-24. RFC 1406 DS1 MIB Objects

Object Name	Restrictions
MIB Tables	
dsx1CurrentTable	Not supported
dsx1IntervalTable	Not supported
dsx1TotalTable	Not supported
DSX1ConfigTable	
dsxTimeElapsed	Not supported
dsxValidIntervals	Not supported
dsx1SendCode	Not supported
dsx1CircuitIdentifier	Not supported
dsx1TransmitClockSource	Not supported
dsx1Fd1	Not supported

Parameters vs. Private Enterprise MIB

The private enterprise MIB file provides the object descriptions for the hardware, software, ports, parameters, and commands groups for all Remote Annex products including the RA 6300.

The RA 6300 supports a subset of the private MIB objects. For example, objects related to parallel ports are not supported by the RA 6300.

Location of Private MIB Files

The standard MIBs reside in the /*annex_root*/**src/snmp** directory.

Private MIB Filenames

The software distribution provides this information in the following files:

MIB FIlename	Description.
xylo.smi	Describes the structure of Xylogics management information bases
xylo-ports.mib	Contains all MIB object groups related to the management and configuration of ports.
xylo-anx.mib	Contains all MIB objects related to configuring the RA 6300 and other Remote Annex products (i.e, Annex-wide configuration settings)
xylo-protocol.mib	Contains all of the protocol-related private MIB groups
xylo-wan.mib	Contains all of the MIB objects related to WANs

Most of the configuration parameters are provided as objects with read-write access permission in the private enterprise MIB. A number of these parameters can be found in the standard MIBs that the SNMP agent supports.

Most MIB object names for the parameters in the private enterprise MIB are preceded by the string "*.iso.org.dod.internet.private.enterprises.xylogics.annex.*".

One of the many exceptions is the object corresponding to the **image_name** parameter. The MIB object name for **image_name** is preceded by the string "*.iso.org.dod.internet.private.enterprises.xylogics.annex.annexcmds.*".

MIB Prefixes

All MIB object names have a prefix that indicates the MIB in which it is defined; Table B-25 lists these prefixes and their corresponding MIB. Table B-26 lists the configuration parameters and the corresponding MIB object names.

There are other settable MIB objects included in the standard MIBs supported by the SNMP agent. The read-only objects defined in the various MIBs allow the SNMP management station to monitor many MIB variables.

Table B-25. Prefixes for MIB Object Names Related to the RA 6300

Prefix	Corresponding MIB file
anx	xylo-anx.mib
gp	xylo-wan.mib
call	xylo-wan.mib
anxt1	xylo-wan.mib

Configuration Parameters vs. MIB Objects

Table B-26 lists the configuration parameter and the corresponding MIB Object Name for the RA 6300.

Table B-26. Parameter vs. MIB Object Name

Configuration Parameter	MIB Object
acp_key	anxAcpKey
allow_snmp_sets	** not applicable **
a_router	anxAppleTalkRouter
authoritative_agent	anxAuthAgent
broadcast_addr	anxBcastAddr
cli_prompt	anxCliPrompt
config_file	anxConfigFile
daylight_savings	anxDaylightSavings
default_zone_list	anxAppleTalkDefZones
disabled_modules	anxDisabledModules
enable_security	anxEnableSecurity
host_table_size	anxHostTableSize
image_name	anxDefaultImageName
inet_addr	anxInetAddr
ipencap_type	anxIpEncapType
ip_forward_broadcast	anxIpFwdBcast

(continued on next page)

Table B-26. Parameter vs. MIB Object Name (continued)

Configuration Parameter	MIB Object
lat_key	anxLatKey
load_broadcast	anxLoadBcast
load_dump_gateway	anxLoadDumpGateway
load_dump_sequence	anxLoadDumpSeq
loose_source_route	anxLooseSrcRoute
max_vcli	anxMaxVcli
min_unique_hostnames	anxMinUniqueHostNames
motd_file	anxMotdFile
name_server_1	anxNameServer1Type
name_server_2	anxNameServer2Type
nameserver_broadcast	anxNameServerBcast
network_turnaround	anxNetTurnAround
node_id	anxAppleTalkNodeId
option_key	anxOptionKey
password	anxPassword
pref_dump_addr	anxPrefDumpAddr
pref_load_addr	anxPrefLoadAddr
pref_name1_addr	anxNameServer1Addr
pref_name2_addr	anxNameServer2Addr

(continued on next page)

Table B-26. Parameter vs. MIB Object Name (continued)

Configuration Parameter	MIB Object
pref_secure1_host	anxSecurServer1Addr
pref_secure2_host	anxSecurServer2Addr
rip_auth	anxRipAuth
rip_routers	anxRipRouteList
routed	anxRouted
rwhod	anxRwhod
security_broadcast	anxSecurBcast
server_capability	anxServerCap
subnet_mask	anxSubnetMask
syslog_facility	anxSysLogFacility
syslog_host	anxSysLogHost
syslog_mask	anxSysLogMask
syslog_port	anxSysLogPort
tcp_keepalive	anxTcpKeepAlive
tftp_dump_name	anxTftpDumpName
tfpt_load_dir	anxTftpDirName
time_broadcast	anxTimeBcast
timezone_minuteswest	anxTimeZone
vcli_password	anxVcliPassword
vcli_security	anxVcliSecurity
zone	anxAppleTalkZone

LAT-specific Configuration Parameters vs. MIB Objects

Table B-27 lists the LAT-specific configuration parameters and the corresponding MIB object names. The string "*.iso.org.dod.internet.private.enterprises.xylogics.annex.*" precedes the MIB object names.

Table B-27. LAT-specific Configuration Parameters vs. MIB Object Name

LAT-specific na Parameter	MIB Object
circuit_timer	anxCircuitTimer
facility_num	anxFacilityNum
group_value	anxLatGroupVal
keep_alive_timer	anxKeepAliveTimer
lat_queue_max	anxLatQueueMax
retrans_limit	anxReXmitLimit
server_name	anxServerName
service_limit	anxServiceLimit
sys_location	anxLatLocation
vcli_groups	anxLatVcliGroupVal

LAT Statistic Objects

Table B-28 lists the LAT statistic objects; these objects provide the same information available in the CLI **netstat** command. The following string precedes the MIB object names: “*.iso.org.dod.internet.private.enterprises.xylogics.annex.*”.

Table B-28. LAT Statistic Objects

MIB Object Name	Description
anxLatRecvRunMsgs	total received run messages
anxLatXmitRunMsgs	total transmitted run messages
anxLatRecvSlots	total received slots
anxLatXmitSlots	total transmitted slots
anxLatRecvBytes	total received bytes
anxLatXmitBytes	total transmitted bytes
anxLatDupMsgs	total duplicate messages
anxLatRexmitMsgs	total retransmitted messages
anxLatBadCircuitMsgs	total bad circuit messages
anxLatBadSlotMsgs	total bad circuit slots
anxLatAcceptHostInits	total accepted host-initiates
anxLatRejectHostInits	total rejected host-initiates
anxLatMultipleNodes	total multiple nodes seen
anxLatCreatedCircuits	total created circuits
anxLatCreatedSessions	total created sessions

(continued on next page)

Table B-28. LAT Statistic Objects (continued)

MIB Object Name	Description
anxLatRecvFrames	total received frames
anxLatXmitFrames	total transmitted frames
anxLatIllegalFrames	total illegal frames
anxLatCircuitTimeouts	total circuit time-outs
anxLatXmitSvcMsgs	total transmitted service messages
anxLatRecvSvcMsgs	total received service messages
anxLatUsedSvcMsgs	total used service messages

TMux-specific Parameters vs. MIB Objects

Table B-29 lists the TMux-specific parameters and their corresponding MIB object names.

The following string precedes the MIB object names: “*.iso.org.dod.internet.private.enterprises.xylogics.annex.*”.

Table B-29. TMux-specific Parameters vs. MIB Objects

TMux Parameter	MIB Object Name
tmux_delay	anxTmuxDelay
tmux_enable	anxTmuxEnable
tmux_max_host	anxTmuxMaxHost
tmux_max_mpx	anxTmuxMaxMpx

IPX-specific Parameters vs. MIB Objects

Table B-30 lists the IPX-specific parameters and their corresponding MIB object names.

The following string precedes the MIB object names: “*.iso.org.dod.internet.private.enterprises.xylogics.annex.*”.

Table B-30. IPX-specific Parameters vs. MIB Objects

IPX Parameter	MIB Object Name
ipx_do_checksum	anxIpxDoChecksum
ipx_dump_password	anxIpxDumpPasswd
ipx_dump_path	anxIpxDumpPath
ipx_dump_username	anxIpxDumpUsername
ipx_file_server	anxIpxFileServer
ipx_frame_type	anxIpxFrameType

Interface Parameters vs. MIB Objects

Table B-31 lists the interface parameters and the corresponding MIB object names. The string “*.iso.org.dod.internet.mgmt.mib-2.interfaces.*” precedes the MIB object names.

Table B-31. Interface Parameters vs. MIB Objects

Interface Parameter	MIB Object
rip_accept	interfaceRipAccept
rip_advertise	interfaceRipAdvertise
rip_default_route	interfaceRipDefRoute
rip_horizon	interfaceRipHorizon
rip_recv_version	interfaceRipRecvVersion
rip_send_version	interfaceRipSendVersion
rip_sub_accept	interfaceRipSubAccept
rip_sub_advertise	interfaceRipSubAdvertise

Global Port Parameters vs. MIB Objects

Table B-32 lists the global port parameters corresponding to the MIB object names. Table B-33 lists the PPP and SLIP port parameters and the corresponding MIB object names. Table B-29 lists the T1-specific parameters and the corresponding MIB object names.

- All global port private MIB object names are preceded by the string “*.iso.org.dod.internet.private .enterprises.xylogics.annex.ports.portTable.portEntry.*” and appended by the port instance number.
- The following string precedes the object names that are in the rs232 MIB: “*.iso.org.dod.internet.mgmt.mib-2.transmission.rs232.*”
- The string “*.iso.org.dod.internet.mgmt.mib-2.char.*” precedes the MIB object names that are in the charlikeMIB.

Table B-32. Global Port Parameters vs. MIB Object Names

Parameter	MIB Object
allow_broadcast	gpGenericAllowBcast callGenericAllowBcast
arap_v42bis	gpArapV42bis callArapV42bis
at_guest	anxpAtGuest
at_nodeid	anxpAtNodeid
at_security	anxpAtSecurity
attn_string	gpLineEditAttnChar callLineEditAttnChar
authorized_groups	gpLatAuthGroupVal

(continued on next page)

Table B-32. Global Port Parameters vs. MIB Object Names (continued)

Parameter	MIB Object
backward_key	gpSignalBackwardKey callSignalBackwardKey
banner	gpGenericBanner callGenericBanner
broadcast_direction	gpBcastDirection callBcastDirection
char_erase	gpLineEditCharErase callLineEditCharErase
cli_imask7	gpGenericCliImask callGenericCliImask
cli_inactivity	gpTimerCliInactivity callTimerCliInactivity
cli_security	anxpCliSecurity
connect_security	anxpConnectSecurity
control_lines	gpSignalCtrlLines callSignalCtrlLines
data_bits	rs232AsyncPortTable.rs232AsyncPortEntry.rs232 AsyncPortBits
forward_key	gpSignalForwardKey callSignalForwardKey
line_erase	gpLineEditLineErase callLineEditLineErase
dedicated_arguments	gpGenericDedicatedArgs callGenericDedicatedArgs

(continued on next page)

Table B-32. Global Port Parameters vs. MIB Object Names (continued)

Parameter	MIB Object
echo	gpLineEditEcho callLineEditEcho
erase_char	gpLineEditEraseChar callLineEditEraseChar
erase_line	gpLineEditEraseLine callLineEditEraseLine
erase_word	gpLineEditEraseWord callLineEditEraseWord
forwarding_count	gpTimerForwardCount callTimerForwardCount
forwarding_timer	gpTimerForwardTimer callTimerForwardTimer
hardware_tabs	gpLineEditHardwareTabs callLineEditHardwareTabs
imask_7bits	anxpImask7Bits
inactivity_timer	gpTimerInactivityTimer
input_flow_control	charPortTable.charPortEntry.charPortInFlowType
input_is_activity	gpTimerInputIsActivity callTimerInputIsActivity
input_start_char	gpSignalInputStartChar callSignalInputStartChar
input_stop_char	gpInputStopChar callInputStopChar
ixany_flow_control	anxpIxanyFlowCtl

(continued on next page)

Table B-32. Global Port Parameters vs. MIB Objects (continued)

Parameter	MIB Object
latb_enable	anxpLatbEnable
map_to_lower	gpLineEditMapToLower callLineEditMapToLower
map_to_upper	gpLineEditMapToUpper gpLineEditMapToLower
max_cap_chall_int	anxMaxChapChallInt
max_session_count	charPortTable.charPortEntry. charPortSessionMaximum
mode	gpGenericMode callGenericMode
modem_var	gpGenericModemVar callGenericModemVar
need_dsr	gpNeedDsr callSignalNeedDsr
newline_terminal	gpLineEditNewLineTerm callLineEditNewLineTerm
net_inactivity	gpNetInactivity callNetInactivity
net_inactivity_units	gpNetInactivityUnits callNetInactivityUnits
output_flow_control	charPortTable.charPortEntry. charPortOutFlowType

(continued on next page)

Table B-32. Global Port Parameters vs. MIB Objects (continued)

Parameter	MIB Object
output_is_activity	gpTimerOutputIsActivity callTimerOutputIsActivity
output_start_char	gpSignalOutputStartChar callSignalOutputStartChar
output_stop_char	gpSignalOutputStopChar callSignalOutputStopChar
parity	rs232AsyncPortTable.rs232AsyncPortEntry. rs232AsyncPortParity
port_password	anxpPortPassword
port_server_security	anxpPortServerSecurity
tcp_keepalive	gpGenericTcpKeepAlive call
stop_bits	rs232AsyncPortTable.rs232AsyncPortEntry. rs232AsyncPortStopBits
tcp_keepalive	gpGenericTcpKeepAlive callGenericTcpKeepAlive
telnet_crlf	gpLineEditTelnetCRLF callLineEditTelnetCRLF
telnet_escape	gpLineEditTelnetEscape callLineEditTelnetEscape

(continued on next page)

Table B-32. Global Port Parameters vs. MIB Objects (continued)

Parameter	MIB Object
term_var	gpGenericTermVar callGenericTermVar
tn3270_printer_host	gpTn3270PrinterHost callTn3270PrinterHost
tn3270_printer_name	gpTn3270PrinterName callTn3270PrinterName
toggle_output	gpLineEditToggleOutput callLineEditToggleOutput
user_name	gpSecurityUserName callSecurityUserName

Table B-33. PPP and SLIP Port Parameters vs. MIB Objects

PPP/SLIP Parameter	MIB Object
allow_compression	gpNetAllowCompression callNetAllowCompression
address_origin	gpNetPppDialupAddr callNetPppDialupAddr
do_compression	gpNetDoCompression callNetDoCompression
local_address	gpNetLocalAddr callNetLocalAddr
metric	gpNetMetric callNetMetric

(continued on next page)

Table B-34. PPP and SLIP Port Parameters vs. MIB Objects (continued)

PPP/SLIP Parameter	MIB Object
ppp_acm	gpPPPAcm callPPPAcm
ppp_mru	gpPPPMru callPPPMru
ppp_ncp	gpPPPNcp callPPPNcp anxPppNcp anxSyncPppNcp
ppp_password_remote	gpPPPRemotePasswd callPPPRemotePasswd
ppp_security_protocol	gpPPPSecurityProto callPPPSecurityProto
ppp_username_remote	gpPPPRemoteUser callPPPRemoteUser
remote_address	gpNetRemoteAddr callNetRemoteAddr
slip_mtu_size	gpSlipMtuSize callSlipMtuSize
slip_no_icmp	gpSlipNoIcmp callSlipNoIcmp
slip_ppp_security	gpNetSlipSecure callNetSlipSecure
slip_tos	gpSlipTos callSlipTos
subnet_mask	gpSlipSubnetMask callSlipSubnetMask

Table B-35. Multi-Link PPP Parameters vs. MIB Objects

Multi-Link PPP Parameter	MIB Object
mp_mrru	mpMrru
mp_endpoint_option	mpEndPointClass
mp_endpoint_value	mpEndPointValue

Book C

Chapter 1 na Commands

The **na** utility provides commands for managing the RA 6300. These commands allow you to:

- Set and display the operating characteristics of the RA 6300.
- Reboot or reset the RA 6300, reset internal ports, and reset sessions.
- Broadcast administrative messages to RA 6300 users.

The **na** utility is stored on and accessed from a UNIX host.

The RA 6300 stores the parameters set using **na** in non-volatile memory. After a reboot or a reset, the RA 6300 updates its run-time parameters with the non-volatile parameters changed by **na**. The **na** utility can communicate with the RA 6300 only when the RA 6300 is running its operational code.

All **na** commands are taken from the **na** standard input: you can run **na** interactively or provide it with input through a file or pipeline. You can create a script file containing **na** commands to configure an RA 6300. This script file can save the configuration information for a specific RA 6300 and, when required, restore the configuration.

This chapter illustrates command names, parameter names, and keywords in their long forms. Examples of **na** commands sometimes appear without the interactive command prompt, and with embedded comments that describe the functions being performed. This format resembles the appearance of **na** scripts; the portion of the script entered at the terminal in response to the command prompt appears in bold type.

Command Notation

Interactive **na** sessions allow you to enter **na** commands with or without arguments or parameters. If you enter the command without arguments or parameters, **na** prompts for them. The conventions for an interactive session are:

- You can abbreviate commands and parameter names to the minimum number of characters that uniquely distinguish the name from any other name that may appear in the same context.
- Type a new-line character to end a command entry. To continue an entry onto the next line, type a backslash (\) character immediately preceding the new-line character.
- To enter a space as an argument, enclose it in double quotes (" "). Otherwise, the space is assumed to be a delimiter.
- The UNIX interrupt character (usually CTRL-C) returns you to the command prompt.

Additionally, **na** permits comments when the # character is present at the beginning of a comment line. All characters between the # and the next new line are ignored. Table C-1 describes the supported arguments for **na**.

Table C-1. Arguments for the na Commands

Argument	Description
annex_identifier	A symbolic name or an IP address assigned to an RA 6300: `lab` or `132.245.254.38` or `0xC0.0x9.0xC8.0x64`
annex_list	A list of one or more *annex_identifiers* separated by commas: `support,132.245.254.42,lab`
annex_parameters	A list of one or more RA 6300 parameters and values separated by white space (space, tab, new line): `pref_load_addr 132.245.254.66\` `pref_dump_addr 132.245.254.66`
interface_identifier	Either **en0** or **port**.
interface_parameters	A list of one or more interface parameters, with or without values, separated by white space (space, tab, newline): `rip_sub_advertise Y`
interface_set	A list of one or more *interface_identifiers* separated by semicolons. An *interface_set* can include interfaces on different RA 6300s: `en0@132.245.254.42`
port_parameters	A list of one or more global port parameters, with or without values, separated by white space: `input_flow_control eia`

Commands

Table C-2 lists the **na** commands; the following subsections describe them.

Table C-2. The na Commands

Command	Description
annex	Defines a default *annex_list* used with subsequent commands.
boot	Boots the RA 6300.
broadcast	Sends a broadcast message to one or more users on internal asynchronous ports.
copy	Copies configuration parameters.
dumpboot	Boots the RA 6300 and produces a dump.
echo	Writes the remainder of the line to the standard output.
help or ?	Displays help for commands and parameters.
interface	Defines a default interface used with subsequent commands.
password	Defines a default administrative password used to communicate with an RA 6300.
port	Specifies the global port.
quit	Terminates **na**.
read	Reads and executes a script file.

(continued on next page)

Table C-2. The na Commands (continued)

Command	Description
reset	Resets an internal port, interface, or subsystem.
set	Defines or modifies the value of a parameter.
show	Displays the current value of a parameter.
write	Writes the current configuration to a script file.

After installing **na** on a UNIX host, type **na** at a terminal connected to this host. No arguments or command line options are available.

```
% na
Annex network administrator Rx.x
command:
```

Seven of the **na** commands use standard UNIX superuser protection – only a superuser at the host can execute these commands: **boot**, **broadcast**, **copy**, **dumpboot**, **read**, **reset**, and **set**.

annex

The **annex** command establishes a default *annex_list* that is used in subsequent commands. Before issuing an **na** command, specify the RA 6300 to which the executed command refers. The RA 6300s you specify using the **annex** command become the default *annex_list*. You can group several RA 6300s into a single list, and then issue one command for the entire group of RA 6300s. The syntax is:

annex *annex_list*

The following example creates an *annex_list* containing one RA 6300 with the Internet address 132.245.6.40:

```
command: annex 132.245.6.40
```

The following example creates an *annex_list* containing two RA 6300s (one specified by its IP address, and the other specified by its name):

```
command: annex 132.245.6.40,frontlobby
```

The following example shows how **na** prompts for missing arguments:

```
command: annex
enter default annex list: 132.245.6.40,frontlobby
```

The following **annex** command displays a message identifying the specified RA 6300, its Internet address, the number of serial lines it has, and its software version:

```
command: annex 132.245.6.1
132.245.6.1: Remote Annex 6300 Rx.x
```

The following **annex** command causes an RA 6300 to prompt for an administrative password, provided that the password has been set and security has been enabled:

```
command: annex frontlobby
Password for 132.245.6.40 <frontlobby>
frontlobby: Remote Annex 6300 Rx.x
```

If you use the **na** command **password** to define a default password for this **na** session, and that password matches the RA 6300 administrative password, no password prompt appears and normal processing continues (see *password* on page 1-16).

The password is not echoed when entered using the **annex** command. If you enter an incorrect password, **na** prompts for the correct one. If the password is incorrect a second time, **na** drops the RA 6300 from the *annex_list*. If an RA 6300 in the list does not respond, **na** ignores that RA 6300 and prints a status message:

```
132.245.6.1: Not responding
Warning:132.245.6.1 has been dropped from the list
```

The **na** utility drops an RA 6300 from the *annex_list* if its name could not be translated to an Internet address, if it does not respond because it is down, or if the wrong Internet address was entered using the **annex** command.

boot

The **boot** command reboots all RA 6300s in the *annex_list* and, optionally, produces a dump of the RA 6300's memory, including the operational code. You can set a time at which the boot is to take place. The **boot** command can send a warning message to users attached to the RA 6300. Table C-3 lists the supported arguments for the **boot** command. The syntax is:

boot [–**adhlq**] [[+] [*HH*:] [*MM*]] [*annex_list*] [*filename*] [*warning*]

When the RA 6300 reboots, it terminates all active connections.

If you try to boot software Release 8.0 and above on a unit with less than 2 MB of system RAM, the **Net**, **Load**, and Status 8 indicators will flash until the **Reset** button is pressed.

If you try to boot with a non-existent image file name, the RA 6300 will hang as it searches for the image. You must press the **Reset** button to recover.

Table C-3. Supported Arguments for the boot Command

Argument	Description
–a	Aborts any delayed boots that are pending.
–d	Causes a dump before rebooting.
–h	Returns to ROM monitor prompt if the RA 6300 is in Test mode.
–l	Boots the operational image and stores it on local media; for use with the stand-alone file system. Only ROM revisions 0600 and greater with the self-boot option loaded support **–l**. After a **boot –l** is executed, the **ls** command may not show the newly-loaded image.
–q	Causes a boot without sending a warning message.
HH:MM	The exact clock time for the boot, e.g., 15:15 indicates 3:15 p.m.
+HH:MM	The number of hours and minutes before the boot takes place, e.g., +2:15 indicates a boot will occur in two hours and fifteen minutes.
annex_list	Specifies the RA 6300s to be booted. If you do not include an *annex_list*, the command prompts for it. Pressing the **Return** key accepts the default *annex_list*.
filename	Identifies the name of the file in which the RA 6300's image is maintained. If you do not enter a *filename*, the RA 6300 prompts for one. Pressing the **Return** key at the prompt directs the RA 6300 to boot the default *filename*. This filename is the value of the RA 6300 *image_name* parameter, or, if that is not set, the name **oper.63.enet**.
warning	Allows an additional 249-character message. Warning messages are sent out to users periodically. If you do not specify a time delay or message, the **boot** command generates an automatic warning message.

The following sample **boot** command requests a boot in one hour and fifteen minutes:

```
command: boot +1:15
annex list (return for default): thirdfloor, 132.245.6.40
filename (return for default): <cr>
warning: Shutting down for PM
```

The RA 6300 can request its boot file from a defined preferred load host. If that host is not defined, or does not respond, the RA 6300 broadcasts its request and boots from the first load host to respond.

broadcast

The **broadcast** command sends a message to modem users on specified internal asynchronous ports at the identified Remote Annex 6300s. The syntax is:

broadcast [=*async_port_set* | =*keyword* [**@***annex_identifier*]] *message*

The *async_port_set* argument indicates the numbers of the internal asynchronous port(s) to which the message is to be broadcast. For example, a port set of 1, 2, 3 specifies internal ports asy1, asy2, and asy3. If the *message* requires more than one line, using the \ character at the end of each line inserts a new line. Table C-4 lists the available keywords.

Table C-4. Supported Keywords for the broadcast Command

Keyword	Description
all	Broadcasts to all asynchronous modem ports and all virtual connections.
virtual	Broadcasts to all virtual CLI connections (you cannot broadcast to a single virtual CLI connection).

copy

The **copy** command requires superuser privileges.

The **copy** command copies a given set of parameters from one RA 6300 (or global port) to another RA 6300 (or global port). Table C-5 defines each copy command. The syntax is:

copy annex *annex_identifier annex_list*

copy interface *interface_name***@***annex_identifier interface_set*

copy printer *printer_number***@***annex_identifier printer_set*

copy port@*annex_identifier*

Table C-5. Descriptions of the copy Command

Command	Description
copy annex	Copies all RA 6300-wide parameters except the IP address, the administrative password, the access control protocol key, LAT key, option key, and the virtual CLI password from the specified RA 6300 to the *annex_list*.
copy interface	Copies all interface parameters from the specified interface to the *interface_set*.
copy port	Copies all global port parameters except the port password from the current RA 6300 to the RA 6300 specified by *annex_identifier*.

To copy global port parameters from the RA 6300 named frontlobby to the RA 6300 at 132.245.6.55:

```
command: copy 1@frontlobby 1@132.245.6.55
```

dumpboot

The **dumpboot** command requires superuser privileges. When the RA 6300 dumpboots, it terminates all active connections.

The **dumpboot** command performs a dump of every RA 6300 specified in the *annex_list* and then reboots the RA 6300. You can set the boot time, and the **dumpboot** command sends a warning message to users attached to the RA 6300. Table C-6 describes the arguments for **dumpboot**. The syntax is:

dumpboot [**–aq**] [[+] [*HH*:] [*MM*]] [*annex_list*] [*filename*] [*warning*]

The following is an example of the **dumpboot** command:

```
command: dumpboot
annex list (return for default): backhall
filename (return for default): <cr>
warning: Diagnostic testing
```

The RA 6300 sends the dump to a defined preferred dump host. If that host is not defined or does not respond, the RA 6300 broadcasts its dump request and dumps to the first host that responds.

Table C-6. Arguments for the dumpboot Command

Argument	Description
–a	Aborts any delayed dump boots that are pending.
–q	Performs a boot without sending a warning message.
HH:MM	The exact clock time for the boot, i.e., 15:15 indicates 3:15 p.m.
+HH:MM	The number of hours and minutes before the boot takes place, e.g., +2:15 indicates a boot will occur in two hours and fifteen minutes.
annex_list	Specifies the RA 6300s for which dumps and boots are to be performed. If you do not include *annex_list*, the command prompts for it. Pressing the **Return** key accepts the default *annex_list.*
filename	Identifies the name of the file in which the RA 6300's image is maintained. If you do not enter a filename, the RA 6300 prompts for one. Pressing the **Return** key at the prompt directs the RA 6300 to boot the default filename. The RA 6300 requests the boot file from a preferred load host if it is defined and available; otherwise, it broadcasts a boot request.
warning	Allows you to enter an additional 250-character message. Warning messages are sent out to users periodically. If you do not specify a time delay or message, the **dumpboot** command generates an automatic warning message.

echo

The **echo** command writes its argument to the standard output. This command is intended for use in script files. The **write** command automatically puts **echo** commands in the script file it writes. The *write* command section of this chapter includes an example of **echo** commands included in the script file created by the **write** command. The syntax is:

echo *message*

help

The **help** (or **?**) command displays on-line help information about **na**. Entering **help** without arguments displays a list of **na** commands. Table C-7 defines the arguments for **help**. The syntax is:

help [*command_name* | *parameter_name* | ***** | **syntax**]

Table C-7. Arguments for the help Command

command_name	Displays the command syntax, along with a description of the command and its arguments.
parameter_name	Displays the legal values for that parameter.
*	Displays available information for all commands and parameters.
syntax	Displays the syntax for all commands.

The **help** *command_name* display looks like this:

```
command: help boot
command: boot Syntax: boot [-adlq][[+][HH:][MM]]\
                      [<filename>] [<warning>
```

The **help** *parameter_name* display looks like this:

```
command: help timezone_minuteswest
timezone_minuteswest (annex parameter):
Minutes west of GMT: an integer
```

Entering **help** followed by the first letter or first few letters of the command or parameter name displays all entries beginning with the string. The following example represents an abbreviated display:

```
command: help t

telnet_escape (serial port parameter):
escape character to use with the telnet command: a character

term_var (serial port parameter):
Terminal variable: a string, maximum sixteen characters

time_broadcast (annex parameter):
broadcast for time server to use if none found:
Y or y to enable; N or n to disable

timezone_minuteswest (annex parameter):
Minutes west of GMT: an integer

toggle_output (serial port parameter):
character used to toggle output: a character

type (printer parameter):
printer interface style: (dataproducts or centronics)
```

interface

The **interface** command establishes a default *interface_set* used in subsequent commands until another interface is specified. Grouping interfaces using an *interface_set* allows you to issue one **na** command to examine or change the parameter values for multiple interfaces. The syntax is:

interface en0 | **port** | **all**

If you do not identify a specific RA 6300 using the @ symbol and a name or Internet address when entering the *interface_set*, all RA 6300s in the current *annex_list* are used. An *interface_set* referring to the default *annex_list* is updated if a new **annex** command is issued. Specifying **all** sets the default *interface_ set* to the global port interface plus **en0**.

This example defines the default *interface_set* as the global port on the RA 6300 whose Internet address is 132.254.6.34. Specifying the global port indicates the interface set is for the PRI calls of protocol types SLIP and IP over PPP:

```
command: interface port@132.254.6.34
```

The next example defines the default *interface_set* as interfaces **en0** on the same RA 6300 as above, plus **port** on the RA 6300 whose Internet address is 132.254.35.120:

```
command: interface en0@132.254.6.34;port@132.254.35.120
```

This example defines the default *interface_set* as all but interface **port** on every RA 6300 in the default *annex_ list*:

```
command: interface port
```

password

The **password** command allows you to define a default password for the current **na** session. This command is useful when administering several RA 6300s with the same password. The syntax is:

password [*password*]

If you enter the command without giving the password, the RA 6300 prompts for one, but does not echo it:

```
command: password
password:
```

When accessing an RA 6300 with security enabled using the **annex** command, **na** will try to match the RA 6300’s default password with the administrative password. If they match, access is authorized automatically; if they do not match, **na** prompts for the RA 6300-specific administrative password. Enter a password for a given RA 6300 only once during an **na** session, even if the RA 6300 is dropped or the default *annex_list* is changed.

quit

The **quit** command terminates the **na** program from a script file; **na** quits when it receives an end-of-file character (usually CTRL-D) or when it reaches the end of an input file. The syntax is:

quit

read

The **read** command requires superuser privileges.

The **read** command reads a script file that contains **na** commands. The **na** program executes these commands as if they were entered at a terminal in interactive mode. Use **read** either to restore an RA 6300 configuration that has been lost, or to copy parameter settings from one RA 6300 to another. The syntax is:

read *filename*

You can create script files using a text editor or the **write** command.

If you plan to use LAT or a feature enabled through the **option_key** parameter, set the **lat_key** and **option_key** parameters manually (using **na** or **admin**) and reboot the RA 6300 *before* issuing the **read** command. Also, make sure the script file does not contain a different **option_key** setting; if it does, delete the setting before issuing a **read**.

The **read** command loads parameters even if the subsystem is disabled.

The following sample script file, called **testscript**, modifies RA 6300 parameters:

```
# standard parameters for RA 6300s on our network
set annex pref_load_addr 132.245.6.63
set annex pref_dump_addr 132.245.6.63
set annex load_broadcast Y
set annex name_server_1 dns
set annex pref_name1_addr 132.245.6.9
set annex cli_prompt “%n%s%p%c”
set annex daylight_savings usa
set annex enable_security Y
set annex vcli_security Y
set annex syslog_mask all
set annex syslog_host 132.245.6.9
```

Use this script as follows:

```
command: annex thirdfloor,frontlobby,backhall
command: read testscript
```

reset

The **reset** command requires superuser privileges. A **reset** issued to an internal port terminates the call on that port.

The **reset** command (available from **na** or **admin**) changes some of the current attributes of all the RA 6300s in the default *annex_list* without rebooting them. Unless you use the **reset** command, changes to configuration parameters for an internal port, virtual CLI connection, security, or name server usually become effective only after booting the RA 6300. Table C-8 describes the supported keywords for the **reset** command. The syntax is:

reset annex [=*annex_list] annex_subsystem* |**session**

reset interface [=*interface_list* | *keyword*]

reset int_modem *modem_range*

reset port =*async_port_list* | *keyword*

reset sync=*sync_port_list* | *keyword*

The allowed values for *annex_subsystem* are **security**, **motd**, **nameserver**, **macros**, **lat**, **syslog**, and **all**.

The **reset annex session** command causes the RA 6300 to re-read the Session Parameter Blocks from the configuration file. Existing calls are not reset. No new calls are answered while the reset is in progress.

The **reset int_modem** command (available from **na** or **admin**) performs the hardware reset modem function. This command resets the modem whether or not a call is active or using the specified modem. For modem range, specify individual modem numbers separated by commas or a range of numbers from 1 to the maximum number of installed internal modems. Separate a range with a hyphen, e.g., **1–23**.

Table C-8. Keywords for the reset Command

Keyword	Definition
all	Resets all serial ports and virtual CLI connections.
annex all	Resets the message-of-the-day, the Session Parameter Blocks, the security, name server, LAT, and *syslog* subsystems, and customized user interface macros.
annex lat	Resets the LAT-specific RA 6300 parameters so that any future LAT circuits (connections) will use the new values; existing circuits will continue to use the old values. This keyword will not terminate existing LAT circuits.
annex macros	Re-reads the customized user interface macros.
annex motd	Re-reads the message-of-the-day.
annex nameserver	Resets the name server parameters and flushes the RA 6300's host table.
annex security	Resets the security parameters and reconnects to the security host.
annex syslog	Resets the *syslog* subsystem. The *syslog* subsystem does not use any changes made to the **syslog_port** parameter.
interface	Resets the interface parameters. The syntax is: **reset interface** [*interface_set* | *keyword*.
virtual	Resets all virtual CLI connections.

set

The **set** command requires superuser privileges.

The **set** command modifies RA 6300 configuration parameters:

set annex	Modifies RA 6300 parameters.
set interface	Modifies interface parameters.
set port	Modifies global port parameters.
set pri	Modifies PRI parameters.
set pri b	Modifies the PRI B channel parameters.

The syntax is:

set annex [=*annex_list*] *annex_parameters*

set interface [=*interface_list*] *interface_parameters*

set port *port_parameters*

set pri *pri_parameters* [*pri_line_parameters*]

set pri b [=*channel_range*] *pri_b_parameters*

The *annex_parameters*, *interface_parameters*, *port_parameters*, and *pri_parameters* and *pri_b_parameters*arguments require a name and a value separated by a space. A space is required between each parameter argument. You can enter more than one parameter argument with each command. If you are entering multiple parameter arguments that require a new line, precede the new line with the ‘‘\’’ character. Changes made to parameters take effect after booting or resetting the RA 6300 or the port(s).

Sample command lines for setting port parameters are:

```
command: set port data_bits 7
command: set port stop_bits 1
command: set port parity odd
command: set port control_lines none
command: set port mode cli
command: set port inactivity_timer 120
```

show

The **show** command displays current RA 6300, interface, global port, or pri parameters:

show annex	Displays RA 6300 parameters.
show interface	Displays interface parameters.
show port	Displays the global port parameters.
show pri	Displays PRI parameters.
show pri b	Displays PRI b-channel parameters.

The syntax is:

show annex [=*annex_list*] [*keyword* | *annex_parameters*]

show interface [=*interface_list*] [*keyword* | *interface_parameters*]

show port [*keyword* | *port_parameters*]

show pri [*keyword* | *pri_line parameters*]

show pri b [=*range*] *b_channel_parameters*

Each keyword displays a subset of parameters:

- Table C-9 lists the keywords and associated parameters for the **show annex** command.
- Table C-10 lists the keywords and associated parameters for the **show interface** command.
- Table C-11 lists the keywords and associated parameters for the **show port** command.
- Table C-12 lists the keywords and associated parameters for the **show pri** command
- Table C-13 lists the keywords and associated parameters for the **show pri b** command.

Table C-9. Keywords for the show annex Command

Keyword	Parameters
all	Displays all RA 6300 parameters.
appletalk	a_router, default_zone_list, node_id, zone
generic	inet_addr, subnet_mask, pref_load_addr, pref_dump_addr, load_broadcast, broadcast_addr, load_dump_gateway, load_dump_sequence, image_name, motd_file, config_file, authoritative_agent, routed, server_capability, disabled_modules, tftp_load_dir, tftp_dump_name, ipencap_type, ip_forward_broadcast, tcp_keepalive, option_key, session_limit, output_ttl
ipx	ipx_frame_type, ipx_file_server, ipx_dump_username, ipx_dump_passwd, ipx_dump_path, ipx_do_checksum
lat	lat_key, facility_num, server_name, sys_location, lat_queue_max, service_limit, keep_alive_timer, circuit_timer, retrans_limit, group_value, vcli_groups, multicast_timer, multisessions_enable

(continued on next page)

Table C-9. Keywords for the show annex Command (continued)

Keyword	Parameters
mop	pref_mop_host, mop_password, login_prompt, login_password, login_timer
nameserver	nameserver_broadcast, rwhod, pref_name1_addr, pref_name2_addr, name_server_1, name_server_2, host_table_size, min_unique_hostnames
router	rip_routers, rip_auth
security	enable_security, security_broadcast, pref_secur1_host, pref_secure2_host, network_turnaround, acp_key, password, allow_snmp_sets, loose_source_route, lock_enable, passwd_limit, chap_auth_name
syslog	syslog_mask, syslog_facility, syslog_host, syslog_port
time	time_broadcast, daylight_savings, timezone_minuteswest, time_server
tmux	tmux_enable, tmux_max_host, tmux_max_mpx, tmux_delay
vcli	max_vcli, cli_prompt, vcli_security, vcli_password, vcli_inactivity

Table C-10. Keywords for the show interface Command

Keyword	Description
all	Displays all interface routing parameters (rip_send_version, rip_horizon, rip_next_hop, rip_sub_accept, rip_accept, rep_recv_version, rip_default_route, rip_sub_advertise, rip_advertise).
en0	Displays the interface parameters for the **en0** interface.
port	Displays the interface parameters for the global port.

Table C-11. Keywords for the show port Command

Keyword	Parameters
all	Displays all asynchronous port parameters.
appletalk	at_guest, at_nodeid, at_security, arap_v42bis
editing	attn_string, echo, telnet_escape, telnet_crlf, map_to_lower, map_to_upper, char_erase, line_erase, hardware_tabs, erase_char, erase_word, erase_line, redisplay_line, toggle_output, newline_terminal, forward_key, backward_key
flow	input_flow_control, input_start_char, input_stop_char, output_flow_control, output_start_char, output_stop_char, ixany_flow_control, need_dsr, v120_mru
generic	mode, location, term_var, prompt, cli_interface, data_bits, stop_bits, parity, max_session_count, allow_broadcast, broadcast_direction, imask_7bits, cli_imask7, banner, tcp_keepalive, default_session_mode, dedicated_arguments, resolve_protocol
ipx	ipx_security
lat	authorized_groups, latb_enable, multisessions_enable
ppp	local_address, address_origin, metric, slip_ppp_security, net_inactivity, do_compression, allow_compression, net_inactivity_units, ppp_mru, ppp_acm, ppp_security_protocol, ppp_username_remote, ppp_password_remote, ppp_ncp, ppp_sec_auto
security	user_name, cli_security, connect_security, port_server_security, port_password, ipx_security, ipso_class
serial	local_address, address_origin, metric, slip_ppp_security, net_inactivity, do_compression, allow_compression, net_inactivity_units

(continued on next page)

Table C-11. Keywords for the show port Command (continued)

Keyword	Parameters
slip	local_address, address_origin, metric, slip_ppp_security, net_inactivity, do_compression, allow_compression, net_inactivity units, subnet_mask, slip_mtu_size, slip_no_icmp, slip_tos
syn	user_name, port_password, ppp_username_remote, ppp_password_remote, slip_ppp_security, address_origin, ppp_security_protocol, ppp_ncp, metric, subnet_mask, ppp_mru, inactivity_timer, input_is_activity, output_is_activity, reset_idle_time_on, net_inactivity, net_inactivity_units
timers	forwarding_timer, forwarding_count, cli_inactivity,: inactivity_timer, input_is_activity, output_is_activity, reset_idle_time_on, long_break, short_break, autodetect_timeout
tn3270	printer_host, printer_name
vci	login_port_password, login_timeout

Table C-12. Keywords for the show pri Command

Keyword	Parameters
all	Displays all of the PRI line parameters: switch_type,num_b_channels, dsx1_line_length, buildout, fdl_type, analog_encoding

Table C-13. Keywords for the show pri b Command

Keyword	Parameters
all	Displays all of the pri B channel parameters: remote_address, ipx_network, ipx_node. (Entering **show pri b** does the same thing.)

write

The **write** command creates a script file from the configuration data for a specific RA 6300. You can modify this script file using any text editor. Use the **write** command either to back up the current RA 6300's configuration or copy it to multiple RA 6300s. Once you write a script file, issuing the **read** command activates the RA 6300 parameter settings contained in the file (for more details, see *read* on page 1-17). The syntax for **write** is:

write *annex_identifier filename*

For security reasons, the following basic RA 6300 and port parameters are written to the script file as comments: **acp_key**, **lat_key**, **option_key**, **password**, **port_password**, **ppp_password_remote**, **rip_auth**, and **vcli_password**.

Since the **inet_addr** parameter uniquely identifies the RA 6300's location in the network, it is not written to the script file and it is not restored during a **read**. You must set this parameter manually.

You can remove the pound sign (#) from the parameters written as comments in the script file, enter valid data for their settings, and issue a **read** command to copy or restore these parameters to another RA 6300.

Entering passwords as plain text in the script file poses a possible security breach for your system. Take appropriate precautions against unauthorized access of this file.

The **write** command writes **set annex**, **set port**, and **set interface** commands into the script file for each basic RA 6300, printer, port, and interface parameter. The **write** command also includes **echo** commands in the script file. When the script is executed using the **read** command, the arguments to the **echo** command are written to the standard output, indicating the progress of the **read**.

The following is an example of the **write** command:

```
command: write 132.245.6.101 fronthall.script
```

The following example uses the **write** and **read** commands to install a new RA 6300 and to create a back-up copy of an RA 6300. The first line writes configuration data for the RA 6300 *thirdfloor* to a file named **thirdfloor.prm**. The data from *thirdfloor* is copied to the new RA 6300 specified in the *annex_list* defined using the **annex** command.

```
command: write thirdfloor.prm
command: annex 132.245.6.40
command: read thirdfloor.prm
```

Following is an excerpt from the script file **fronthall.script**:

```
# annex 132.245.6.101

echo setting annex parameters
set annex pref_load_addr 132.245.6.75
set annex pref_dump_addr 132.245.6.75
set annex load_broadcast Y
set annex image_name “”
set annex subnet_mask 255.255.255.0
set annex authoritative_agent Y
         :
echo setting serial port parameters global port
set port data_bits 8
set port stop_bits 1
set port parity none
set port do_compression N
set port slip_allow_compression Y
set port slip_no_icmp Y
set port slip_tos Y
```

Chapter 2
Configuration Parameters

This chapter replaces Chapter 2 in Book C of the *Remote Annex Administrator's Guide for UNIX*.

The configuration parameters define the operating characteristics for a given RA 6300. There are three ways in which you can configure and manage these parameters:

- The host-based **na** utility sends requests to the RA 6300 to read, set, reset, show, or copy configuration parameters (for more details, see *na Commands* on page C-1).
- The CLI **admin** command is a counterpart of **na** that runs locally on the RA 6300.
- The SNMP agent included in the RA 6300 software provides MIB objects that can be read and set by a standard SNMP management station; these MIB objects correspond to the configuration parameters (for more details, see *Simple Network Management Protocol (SNMP)* on page B-57).

This chapter includes the following sections:

- *Parameter Conventions*
- *Parameter Descriptions*

Parameter Conventions

This section describes the conventions for entering parameter values and returning those values to the supplied defaults.

Entering Parameter Values

The conventions for entering parameter values depend on the type of information the parameter defines.

- For parameters requiring an IP address, specify the address in dot notation as a decimal number (from **0** to **255**), a hexadecimal number, or a combination of both: 192.9.200.100, 0xC0.0x9.0xC8.0x64, or 192.9.200.0x64.
- For parameters requiring a yes/no input, use either **Y** or **N**. These parameters are not case sensitive.
- For parameters that define passwords, the **na**/**admin** command **show** displays only **"<set>"** or **"<unset>"**; it never displays the values entered for the parameters. If you forget a password after setting it, you can reset it only by using the ROM monitor **erase** command to erase all of the RA 6300's non-volatile memory.

  Saving the configuration to a file (using the **write** command) prevents having to reconfigure the RA 6300 if non-volatile memory is erased.

- Parameters that require a string for input allow a maximum of 16 characters, unless otherwise specified.

Setting Parameters to Supplied Defaults

Each configuration parameter, except the RA 6300’s IP address, has a default value. Using the **na** command **set**, you can return any parameter to its default setting (for more details, see *set* on page C-20). Depending on the parameter type, the syntax options are:

Setting Annex Parameters

1. **set annex *annex_parameter* 0**

   The **set annex** *annex_parameter* **0** command sets parameters that require a numeric value. For example, to set **pref_dump_addr** to its default, 0.0.0.0, enter:

   ```
   command: set annex pref_dump_addr 0
   ```

2. **set annex *annex_parameter* ""**

   The **set annex** *annex_parameter* "" command sets all parameters that require a string value; these parameters default to either a null string ("") or **“<unset>”**. For example, to set **image_name** to its default, a null string (""), enter:

   ```
   command: set annex image_name ""
   ```

3. **set annex *annex_parameter* default**

   The **set annex** *annex_parameter* **default** command sets all other parameters. These parameters are set by choosing either an option from a known list or a yes/no response. For example, to set **enable_security** to its default, **N**, enter:

   ```
   command: set annex enable_security default
   ```

Setting interface Parameters

To set an interface parameter to its default value, use the **set interface** command:

1. **set interface *interface_parameter* ^@**
2. **set interface *interface_parameter* " "**
3. **set interface *interface_parameter* default**

Setting Port Parameters

Unlike port parameters used with other Annexes, the Remote Annex 6300 port parameters are global. This means that each parameter applies to all internal ports of the type listed in the parameter description: asynchronous (*asy* and *ta*) and synchronous (*syn*).

To set a global port parameter to its default value, use the **set port** command:

1. **set port *port_parameter* ^@**
2. **set port *port_parameter* ""**
3. **set port *port_parameter* default**

The **set port** *port_parameter* **^@** command sets parameters that have single-character default values. Enter the default value as a two-character sequence consisting of the circumflex character (^) followed by the at sign (**@**). For example, to set **erase_word** to its default value, ^W, enter:

```
command: set port erase_word ^@
```

The **set port** *port_parameter* "" command sets parameters that require a string; these parameters default to either a null string ("") or **"<unset>"**. For example, to set **user_name** to its default, the null string, enter:

```
command: set port user_name ""
```

The **set port** *port_parameter* **default** command sets all other parameters. These parameters are set by choosing either an option from a known list or a yes/no response. The *keyword* **default** sets these parameters to their default values.

Setting All Parameters

To set all of the RA 6300’s parameters to the supplied defaults, use the ROM monitor **erase** command (see the appropriate *Annex Hardware Installation Guide*). This command erases all parameters, including the RA 6300’s IP address. After issuing **erase**, you must re-enter the RA 6300’s IP address and re-configure the RA 6300.

Parameter Descriptions

The RA 6300 configuration parameters are grouped by type, e.g., Annex, interface, etc. Parameters within these groups are further divided by relative function. Each function has an associated keyword (e.g., nameserver, security, time, etc.). The **set** and **show** commands accept these keywords as arguments.

- Table C-14 lists the keywords and the associated parameters that display with the **show annex** command.
- Table C-15 lists the keywords and the associated parameters that display with the **show interface** command.
- Table C-16 lists the keywords and the associated parameters that display with the **show port** command.
- Table C-17 lists the keywords and associated parameters that are displayed by the **show pri** command.
- Table C-18 lists the keywords and associated parameters listed with the **pri show b** command.

In this chapter, Annex parameters are those parameters that you set or show via the **set annex** or **show annex** command. RA 6300 parameters are all of the configuration parameters, including Annex, interface, port, and pri parameters.

The LAT-related parameters are visible only when the **lat_key** parameter contains the correct key value (for more details, see *lat_key* on page 2-66).

The AppleTalk, tn3270, and IPX parameters are visible only when the **option_key** parameter is set to the correct key value.

Table C-14. Keywords for the show annex Command

Keyword	Parameters
all	Displays all Annex parameters.
generic	inet_addr, subnet_mask, pref_load_addr, pref_dump_addr, load_broadcast, broadcast_addr, load_dump_gateway, load_dump_sequence, image_name, motd_file, config_file, authoritative_agent, routed, server_capability, disabled_modules, tftp_load_dir, tftp_dump_name, ipencap_type, ip_forward_broadcast, tcp_keepalive, option_key, session_limit, output_ttl
vcli	max_vcli, cli_prompt, vcli_security, vcli_password, vcli_inactivity
nameserver	nameserver_broadcast, max_chap_chall_int, rwhod, pref_name1_addr, pref_name2_addr, name_server_1, name_server_2, host_table_size, min_unique_hostnames
security	enable_security, security_broadcast, pref_secure1_host, pref_secure2_host, network_turnaround, loose_source_route, acp_key, password, allow_snmp_sets, lock_enable, passwd_limit, chap_auth_name
time	time_broadcast, daylight_savings, timezone_minuteswest, time_server
syslog	syslog_mask, syslog_host, syslog_facility, syslog_port

(continued on next page)

Table C-14. Keywords for the show annex Command (continued)

Keyword	Parameters
mop	pref_mop_host, mop_password, login_prompt, login_password, login_timer
lat	lat_key, facility_num, server_name, sys_location, lat_queue_max, service_limit, keep_alive_timer, circuit_timer, retrans_limit, group_value, vcli_groups, multicast_timer, multisessions_enable
appletalk	a_router, default_zone_list, node_id, zone
router	rip_routers, rip_auth
ipx	ipx_file_server, ipx_frame_type, ipx_dump_username, ipx_dump_password, ipx_dump_path, ipx_do_checksum
tmux	tmux_enable, tmux_max_host, tmux_delay, tmux_max_mpx

Table C-15. Keywords for the show interface Command

Keyword	Description
all	Displays all global port and **en0** interface routing parameters (rip_send_version, rip_horizon, rip_next_hop, rip_sub_accept, rip_accept, rep_recv_version, rip_default_route, rip_sub_advertise, rip_advertise).
port	Displays the interface parameters for the global port.
en0	Displays the interface parameters for the **en0** interface.

Table C-16. Keywords for the show port Command

Keyword	Parameters
all	Displays all asynchronous port parameters
generic	mode, location, type, term_var, prompt, cli_interface, speed, autobaud, data_bits, stop_bits, parity, max_session_count, allow_broadcast, broadcast_direction, imask_7bits, cli_imask7, banner, tcp_keepalive, default_session_mode, dedicated_arguments, resolve_protocol
flow	control_lines, input_flow_control, input_start_char, input_stop_char, output_flow_control, output_start_char, output_stop_char, ixany_flow_control, need_dsr, v120_mru
timers	forwarding_timer, forwarding_count, cli_inactivity, inactivity_timer, input_is_activity, output_is_activity, reset_idle_time_on, long_break, short_break
security	user_name, cli_security, connect_security, port_server_security, port_password, ipso_class, ipx_security
vci	login_port_password, login_timeout
editing	attn_string, echo, telnet_escape, telnet_crlf, map_to_lower, map_to_upper, char_erase, line_erase, hardware_tabs, erase_char, erase_word, erase_line, redisplay_line, toggle_output, newline_terminal, backward_key, forward_key

(continued on next page)

Table C-16. Keywords for the show port Command (continued)

Keyword	Parameters
serial	local_address, address_origin, metric, slip_ppp_security, net_inactivity, do_compression, allow_compression, net_inactivity_units
slip	subnet_mask, slip_mtu_size, slip_no_icmp, slip_tos, address_origin, net_inactivity_unit, do_compression, net_inactivity, slip_ppp_security, metric, local_address
ppp	ppp_acm, ppp_mru, ppp_security_protocol, ppp_username_remote, ppp_password_remote, ppp_ncp, ppp_sec_auto, address_origin, mp_endpoint_address, mp_endpoint_class
appletalk	at_guest, at_nodeid, at_security, arap_v42bis
tn3270	printer_host, printer_name
lat	authorized_groups, latb_enable, multisessions_enable

Table C-17. Keywords for the show pri Command

Keyword	Parameters
all	switch_type, num_b_channels, dsx1_line_length, buildout, fdl_type, analog_encoding, remote_address, ipx_network, ipx_node

Table C-18. Keywords for the show pri b Command

Keyword	Parameters
all	remote_address ipx_network, ipx_node. (Entering **show pri b** displays the same parameters.)

The parameter descriptions that follow are in alphabetical order.

The port parameters in this section refer to the internal ports (*asy*, *syn*, and *ta*) supported by the Remote Annex 6300 and are also referred to as global port parameters. In addition, these parameters can be included in Session Parameter Blocks within the RA 6300 configuration file. For information on internal ports and Session Parameter Blocks, see *Configuring the PRI Interface, Global Ports, and Sessions* on page A-51.

All parameters that can be used for asynchronous (*asy*)ports can also be used for *ta* ports.

a_router

The Ethernet address of the network's A_Router. The RA 6300 uses this value as a hint at start-up. When a Routing Table Maintenance Protocol (RTMP) message arrives from this Ethernet address, the RA 6300 gleans the AppleTalk DDP address from the packet and tries to talk to the AppleTalk router. The address is a hexadecimal Ethernet address, e.g., 00-7F-12-33-44-55. The default is **00-00-00-00-00-00**.

acp_key

This Annex parameter defines the encryption key used to exchange messages between the Annex and the security server. This parameter works only when the **enable_security** parameter is set to **Y** and a security server is defined. The security server maintains the encryption key for each Annex in the **acp_keys** file. The default for this string is **"<unset>."**

The RA 6300 and the security server can communicate only when this parameter's value matches the RA 6300's value in the security server's **acp_keys** file.

address_origin

This asynchronous port parameter determines where the RA 6300 looks to find the local and remote IP addresses to use for the endpoints of a PPP/IPCP link. Table C-19 describes the options. The default is **local**.

Table C-19. Valid Options for address_origin Parameter

Option	Description
acp	The RA 6300 passes its own address, and the user name and port, to the ACP host. The host then determines the local and remote addresses for the link by searching for entries in the **acp_dialup** file.
local	The RA 6300 uses values set by the **local_address** and **remote_address** parameters. It does not search **acp_dialup**.
dhcp	The RA 6300 contacts a DHCP server to obtain a remote address dynamically on behalf of the remote client. See *Dynamic Allocation of Network Addresses* on page A-274 for a complete explanation. This value is valid only when the port is effectively in PPP mode (e.g., the mode was set to **ppp** or the **ppp** command was issued when the port was in CLI mode). DHCP is not supported with SLIP.

When **address_origin** is set to **acp**, the *local* and *remote* field settings in the **acp_dialup** file supersede the values set in the **local_address** and **remote_address** port parameters (for more details, see *Determining Dial-up Addresses using the acp_dialup File* on page A-278).

This parameter replaces the **dialup_addresses** parameter for Release 13.2 and later. Earlier releases do not recognize **address_origin** and require the use of **dialup_addresses** instead.

allow_broadcast

This asynchronous port parameter allows an asynchronous port to receive administrative broadcast messages generated by the **boot** and **broadcast** commands. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

allow_compression

This asynchronous/synchronous port parameter allows the RA 6300 to use TCP header compression on a SLIP or PPP line. Header compression occurs only if the other side of the serial link initiates compression. A **Y** enables this parameter, an **N** disables it. The default is **N**.

analog_encoding

This PRI line parameter specifies the encoding type used for modem calls. Valid values are:

- **a_law** (used for E1/PRI lines in Europe)
- **mu_law** (used for T1/PRI in the U.S.)
- **auto** (the default, which uses **a_law** or **mu_law** as appropriate)

Typically, you do not need to change this parameter. To check that the correct value is being used, issue the **pri** command from the superuser CLI.

allow_snmp_sets

This Annex parameter enables the RA 6300 to accept and process SNMP **set** commands. When disabled, the RA 6300 rejects all SNMP **set** commands; the RA 6300 SNMP agent returns the error *no such name* for the first object in the **set** command. A **Y** enables this parameter, an **N** disables it. The default is **N**.

arap_v42bis

This asynchronous port parameter enables V.42bis compression during an AppleTalk session. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

at_guest

This asynchronous port parameter allows guests to log into an AppleTalk session. When **at_guest** is enabled, if a client requests guest access, the RA 6300 asks ACP for username guest privileges. A **Y** enables this parameter, an **N** disables it. The default is **N**.

at_nodeid

This asynchronous port parameter defines the node ID hint used for an ARA client during connection establishment. This parameter value is an AppleTalk address in the form *net.node*. The valid *net* values are **0** to **65534**. The valid *node* values are **0** to **254**. The default is **0.0**.

at_security

This asynchronous port parameter turns on ACP service for an AppleTalk session on this port. When both **at_security** and **enable_security** are enabled, the RA 6300 uses ACP to get security information about the client, including authentication, logging, and zone access. If **at_security** is not enabled, the RA 6300 uses only local security. A **Y** enables this parameter, an **N** disables it. The default is **N**.

attn_string

This asynchronous port parameter defines a control character sequence that returns users to the CLI prompt. Users can define a temporary control character sequence using the CLI **stty attn** command; **stty attn** "" disables the sequence. The default is no control character sequence, displayed as ""; the default for virtual CLI connections is **CTRL-A** (^A).

If you are running a **stats** [**–sm** [*ports*] | [*time*]] command with a defined time interval, the RA 6300 ignores an attention string with multiple characters.

autodetect_timeout

This asynchronous/synchronous port parameter specifies the number of seconds that the RA 6300 waits for **auto_detect** mode to identify an incoming call as PPP. If the number of seconds is exceeded or the user enters a carriage return before the call is detected as PPP, the RA 6300 places the user in CLI mode. Valid values are **1–60**. The default is **30**.

authoritative_agent

This Annex parameter enables the RA 6300 to send an ICMP Address Mask reply to a host that broadcasts a subnet mask request. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

authorized_groups

This asynchronous port parameter specifies the LAT protocol remote group codes that are accessible to users on a given RA 6300 port. You can enter *all*, *none*, a series of numbers between *0* and *255* separated by commas (e.g., *1, 5,7*) or a range of numbers between *0* and *255* separated by dashes (e.g., *1–5,200–255*) followed by **enabled** or **disabled**. The default is *all* **disabled**.

backward_key

This asynchronous port parameter specifies a character or string that reopens the next lower numbered session (already established at your port) from within the current session without returning to local mode. When defining this value, use a unique, unused character (such as Control B) or a string of characters. To clear an existing setting, enter a null string (""). The default is no control character sequence.

On virtual (*telnet*) ports, the **backward_key** value is limited to one printable or Control character. If the user tries to set this value to more than one character, i.e., a string, the setting is ignored and the previous value is restored.

On non-virtual ports, a **backward_key** string can range from 1 to 16 characters.

banner

This asynchronous port parameter controls whether or not the RA 6300 banner and message-of-the-day display on CLI ports. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

This parameter has effect only at the CLI level.

broadcast_addr

This Annex parameter defines the IP address for RA 6300 broadcasts. It is recommended that you set a subnet broadcast address, if possible. In this case, you set the subnet portion of the broadcast address to match the RA 6300 subnet address, as determined by the RA 6300 subnet mask, and you set the host portion of the broadcast address to all one-bits. For example, if the RA 6300 subnet address is 132.254.9.0, and the RA 6300 subnet mask is 255.255.255.252, you should set the broadcast address to 132.254.9.3. To calculate this, subtract the subnet mask from 255.255.255.255. Thus, in the previous example, you subtract 255.255.255.252 from 255.255.255.255 to arrive at 0.0.0.3.

If your network is not subnetted, you can specify a network broadcast address. In this case, you set the network portion of the broadcast address to match the RA 6300 network address, as determined by the intrinsic mask for the network class. And you set the host portion of the broadcast address to all 1-bits.

Finally, you can set a limited broadcast address of 255.255.255.255 that reaches all nodes on the subnet. However, if you have more than one subnet on the same physical cable, the RA 6300 will broadcast to all nodes on all of the subnets. This can be troublesome if some of the subnets or nodes do not recognize the broadcast.

The default for **broadcast _addr** is **0.0.0.0**, which RA 6300 RIP routing does not support (because most hosts do not recognize it).

broadcast_direction

This asynchronous port parameter defines the direction in which an administrative broadcast message is sent on a port. The options are **network** or **port**; the default is **port**. This parameter is valid only for a **slave** port (defined by the **mode** parameter).

If you specify **network**, the RA 6300 sends administrative broadcast messages out the network side of the connection to the initiator. If you specify **port**, the RA 6300 sends broadcast messages out the port side of the connection.

buildout

This PRI line parameter is applicable only to RA 6300s with internal CSUs. It is a string defining the CSU transceiver line provided by the telephone company. Valid values are: **0db**, **7.5db**, **15db**, **22.5db**. The default is **0db**.

chap_auth_name

This Annex parameter defines the character string that is used as the *Name* field entry when issuing a CHAP *challenge* over a PPP link. The minimum string length is one character and the maximum string length is 16 characters; the default is **chap**.

Change the default value:

- If you want the remote end of the PPP link to pick a secret token that depends on the received name.
- If you want to achieve the highest level of security.

char_erase

When this asynchronous port parameter is enabled, the RA 6300 echoes both the character erase and the word erase characters for a video terminal; i.e., the previous character (or word) looks as if it has been erased. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

When **char_erase** is disabled, the RA 6300 echoes the erase characters for a hard-copy terminal. It echoes the first erase character as a “\” followed by the deleted character. Each additional use of the erase character deletes and displays another character. The first character typed (other than the erase character) echoes a “/” and the character, e.g., typing “**asdf**<Delete><Delete>**g**” echoes as “**asdf\fd/g**.”

This parameter has effect only at the CLI level.

circuit_timer

This Annex parameter defines the time interval in tens of milliseconds between the transmission of LAT packets (e.g., if you enter *9*, the time interval will be 90 milliseconds). Allowable values range from **1** to **25**. The default value is **8** (80 milliseconds).

cli_imask7

When this asynchronous port parameter is enabled, the RA 6300 masks CLI input to seven bits. The RA 6300 masks input only at the CLI. When **cli_imask7** is disabled, the RA 6300 expects eight-bit ASCII input. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

cli_inactivity

This asynchronous port parameter specifies the amount of time in minutes that the RA 6300 remains idle before disconnecting a CLI session from a port. Unlike the **inactivity_timer**, this timer does not disconnect a CLI session with active jobs. Allowable values range from **0** (or **off**) to **255** minutes or **immediate**. The default is **0**.

Entering **0** disables the timer; entering **255** causes the RA 6300 to disconnect as soon as it exits from its last job. Entering **immediate** causes the RA 6300 to hang up the port immediately after exiting the last job.

cli_interface

This asynchronous port parameter allows you to control the prompt that appears for VMS or UNIX environments. Allowable values are **vci** and **uci**. The default is **uci**.

When set to **vci**, the *Local*> prompt is displayed followed by the *Username*> prompt; the **uci** setting provides a standard UNIX interface (with prompts defined by the **cli_prompt** and **prompt** parameters).

cli_prompt

This Annex parameter defines the RA 6300 prompt for all CLI users. This parameter uses formatting codes consisting of the percent character (%) and a single lowercase letter. You can combine up to 16 of these codes (e.g., *%a%c*). You can also enter text that will appear in the prompt as long as the entry as a whole does not exceed 32 characters. The default prompt is %a%c (the string *annex:*). Table C-20 lists the formatting codes.

Table C-20. Formatting Codes for RA 6300 Prompts

Code	Expansion
%a	The string *annex*.
%c	A colon followed by a space.
%d	The current date and time in the following format: Mon Mar 14 13:59:42 1991.
%i	The RA 6300's IP address.
%j	A new line character, skip to the beginning of the next line.
%n	The RA 6300's name, if known, or the IP address.
%r	The string *port*.
%s	A space.
%t	The current time in 24-hour format.
%u	The user name defined for the port; if none, a null string.

cli_security

This asynchronous port parameter enables user authentication by the host-based ACP server for all CLI connections. When disabled, you cannot use any RA 6300 security mechanism other than the administrative password for CLI ports. A **Y** enables this parameter, an **N** disables it. The default is **N**.

When **cli_security** is enabled, the RA 6300 logs PPP/SLIP logins/logouts to the ACP log file.

config_file

This Annex parameter defines the file name for the configuration file maintained on the load host. This file contains information about gateways, macros, and services; it must reside in the directory **/usr/spool/erpcd/bfs**. The default file name is **config.annex**.

connect_security

This asynchronous port parameter enables the host-based security policy for access from the CLI to the network (using **telnet** and **rlogin** only). If **connect_security** is enabled, the user must receive authorization to connect to a host on the network. The supplied security policy scans the file **/install-directory/acp_restrict** to authorize a connection to a host from the RA 6300. If authorization is not granted, the connection is not made. A **Y** enables this parameter, an **N** disables it. The default is **N**.

data_bits

This asynchronous port parameter defines the number of data bits in a character. This value does not include the start, stop, or parity bits. Allowable values are **5** through **8**. The default is **8**.

daylight_savings

This Annex parameter defines the daylight savings time for your geographic location. The RA 6300 uses this parameter to adjust the time display for daylight savings time. Valid options are **us**, **australian**, **british**, **canadian**, **east_european**, **mid_european**, **west_european**, and **none**; the default is **us**.

dedicated_arguments

This asynchronous port parameter defines the command line arguments used for dedicated ports. The parameter accepts a 100-character string. Use this parameter in conjunction with the **mode** (see *mode* on page 2-73). The default is a **null string** ("").

default_zone_list

This Annex parameter contains the zone list that is sent to AppleTalk clients in case of an ACP failure. The string size ranges from 1 to 100 characters. You must use spaces to separate zone names (e.g., *general engineering lab*). To escape embedded spaces within a zone name, use the backslash (\) character. If you do not set this parameter, the RA 6300 provides the network zone list. The default is a **null string** ("").

default_session_mode

This asynchronous port parameter defines the default session mode when the VMS interface is configured (i.e., when **cli_interface** is set to **vci**). Valid options are **interactive**, **passthru**, **passall**, or **transparent**. The default is **interactive**.

dhcp_broadcast

This Annex parameter enables and disables the use of DHCP broadcast messages; a DHCP proxy client will use it to determine if it can generate DHCP broadcast messages as a means of “discovering” a DHCP server. The options for the parameter are **Y** or **N**, **N** being the default (i.e., the broadcast of DHCP messages is disabled). **dhcp_broadcast** functions consistently with all other broadcast parameters defined in the system.

disabled_modules

This Annex parameter allows you to disable individual software modules to free memory space. If you enter more than one module, separate module names using commas. Valid options are **admin**, **atalk**, **edit**, **fingerd**, **ftpd**, **ipx, lat**, **nameserver**, **ppp**, **slip**, **snmp**, **tn3270**, **tstty**, **vci**, **all**, or **none**. The default is **vci** (disables the RA 6300 VMS interface).

You should exercise extreme caution when disabling modules:

- Entering a null string ("") sets this parameter to its default value.
- If **disabled_modules** is set to a value other than **none** and **server_capability** includes the operational image, no modules are disabled; a syslog message announces this override.
- The **vci** option disables the RA 6300 interface for VMS environments along with the following commands: **backwards**, **change**, **clear**, **crash**, **define**, **disconnect**, **forward**, **list**, **logout**, **resume**, **set**, **show**.
- If **lat_key** is invalid and **server_capability** is set to **none**, the LAT code is freed for use by the system.
- Disabling LAT also disables the CLI commands **services**, **connect**, and **queue**.
- Disabling **admin** and **snmp** can cause problems if host-based **na** is not available. To change parameters in this case, return to monitor mode, erase the parameters in non-volatile memory, and reconfigure the RA 6300.

do_compression

This asynchronous port parameter starts TCP/IP header compression on a SLIP link. When enabled, the RA 6300 negotiates for TCP/IP compression for both sides of the connection. A **Y** enables this parameter, an **N** disables it. The default is **N**.

dsx1_line_length

This PRI line parameter specifies the approximate distance, in meters, from the RA 6300 PRI interface to the external CSU. Enter this as a range, e.g., **0–25**, as a single number, e.g., **30**, or as a unique part of a range, e.g., **136–**.

If you enter a single number, the RA 6300 picks the range that the number falls into. (You can then use the **pri show** command to determine the range assigned.) Valid ranges are: **0–25**, **26–65**, **66–100**, **136–185**, **166–185**, and **186–210**. The default is **0-25**.

echo

This asynchronous port parameter directs an RA 6300 to echo all characters as a user types. This echo occurs only at the CLI level. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

enable_security

This Annex parameter activates the security system. To enable any security features, set this parameter to **Y**. A **Y** enables this parameter, an **N** disables it. The default is **N**.

erase_char

This asynchronous port parameter defines a control character sequence for the CLI erase character. The default is the **Delete** key (displayed as **^?**).

erase_line

This asynchronous port parameter defines a control character sequence for the CLI line erase character. The default is **CTRL-U** (^U).

fdl_type

This PRI line parameter specifies the type of Facilities Data Link supported by the telephone company for your PRI line. Valid values are **att** and **ansi**. This parameter is applicable only to internal CSUs used with T1/PRI lines.

erase_word

This asynchronous port parameter defines a control character sequence for the CLI word erase character. The default is **CTRL-W** (^W).

facility_num

This Annex parameter identifies a LAT host by number. Allowable values range from **0** to **32767**. The default value is **0**.

forward_key

This asynchronous port parameter specifies a character or string that reopens the next available, higher numbered session already established at your port. When defining this value, use a unique, unused character (such as Control F) or a string of characters. To clear an existing setting, enter a null string (""). The default is no control character sequence.

On virtual (*telnet*) ports, the **forward_key** value is limited to one printable or Control character. If the user tries to set this value to more than one character, i.e., a string, the setting is ignored and the previous value is restored.

On non-virtual ports, a **forward_key** string can range from 1 to 16 characters.

forwarding_count

This asynchronous port parameter controls RA 6300 port behavior for received characters. When set to a number other than zero, the port does not forward characters until it receives the specified number of characters. When set to zero, the port uses the value in the **forwarding_timer** parameter. Allowable values range from **0** to **255**. The default is **0**.

If you use both **forwarding_count** and **forwarding_timer**, the RA 6300 uses the value that occurs first.

Setting **forwarding_count** to one or **forwarding_timer** to zero may have a severe effect on the network when heavy serial input occurs.

forwarding_timer

This asynchronous port parameter sets the amount of time in ten millisecond (ms) intervals that can elapse before an RA 6300 forwards received data. If new data arrives before the timer expires, the RA 6300 resets the timer. Allowable values range from **0** to **255** or **off**. The default is **5** (**50 ms**); if you set the value to **0**, the RA 6300 uses **5**.

If you use both **forwarding_count** and **forwarding_timer**, the RA 6300 uses the value that occurs first.

Setting **forwarding_count** to one or **forwarding_timer** to zero may have a severe effect on the network when heavy serial input occurs.

group_value

This Annex parameter specifies the LAT protocol remote group codes that can access local services offered by an RA 6300. To access these services, the RA 6300 must have at least one enabled group code that matches the service's group codes. Valid options are *all*, *none*, a series of numbers between 0 and 255 separated by commas, or a range of numbers between 0 and 255 separated by dashes, followed by **enabled** or **disabled**. The default is *all* **disabled**.

hardware_tabs

This asynchronous port parameter allows the RA 6300 to convert ASCII tab characters to the correct number of spaces when a terminal does not support hardware tabs. This occurs only at the CLI level. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

host_table_size

This Annex parameter defines the number of entries allowed in the host table. Allowable values range from **0** to **255**. Entering **255** allows an unlimited number of entries; entering **254** indicates that there is no host table. In this case, the RA 6300 requires a name server to resolve every host name. The default is **64**.

image_name

This Annex parameter defines the file name containing the operational code that loads by default when you boot an RA 6300. The string size ranges from 0 to 100 characters. The default is a **null string** ("").

imask_7bits

This asynchronous port parameter enables an RA 6300 to mask input to seven bits. When disabled, the RA 6300 expects eight-bit ASCII input. This parameter has no effect on transmitted characters. A **Y** enables this parameter, an **N** disables it. The default is **N**.

SLIP and PPP will not work if this parameter is enabled.

inactivity_timer

This asynchronous port parameter specifies the number of minutes that a port can remain inactive. If the timer expires, the RA 6300 terminates all sessions and resets the port.

You can use the **input_is_activity** and **output_is_activity** parameters to define activity as input to the port or output from the port. Setting these parameters to **N** causes the timer to run independent of activity. Allowable values range from **0** to **255**. The default is **0** (timer disabled).

If you want a port to reset after a given number of minutes, regardless of any activity, you must also set the following parameters:

inactivity_timer	***x***
input_is_activity	**N**
output_is_activity	**N**
type	**dial_in**

These setting are required because the timer does not start until one of the following events occur:

1. **Input occurs and** input_is_activity **is set to** Y.
2. **Output occurs and** output_is_activity **is set to** Y.
3. **A port with** type **set to** dial_in **starts (regardless of the value of the** control_lines **parameter).**

inet_addr

This Annex parameter defines the RA 6300's IP address. This 32-bit address contains four 8-bit fields separated by periods. Each field contains a number ranging from 0 to 255 or a hexadecimal number. The IP address always displays in decimal notation. This parameter has no default.

input_flow_control

This asynchronous port parameter specifies the method of flow control for input received from a device connected to an asynchronous port. Table C-21 describes the valid options; the default is **bell**.

Table C-21. Valid Options for the input_flow_control Parameter

Option	Description
bell	The RA 6300 rings the terminal bell (sends ^G) when its input buffer is full.
eia	Flow control is delegated to a lower level (e.g., a parallel port). The **control_lines** parameter must be set to **flow control** or **both**, and the device must be wired appropriately.
start/stop	Designates flow control by recognizing xon and xoff characters. This action is independent of the **control_lines** parameter setting.
none	Disables flow control; characters are lost if the buffers overflow.

input_is_activity

This asynchronous port parameter defines activity as input. When enabled, the RA 6300 sets the inactivity timer when it receives input at the port. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

input_start_char

This asynchronous port parameter defines the control character sequence that restarts input if the **input_flow_control** parameter is set to **start/stop**. The default is **CTRL-Q** (^Q).

input_stop_char

This asynchronous port parameter defines the control character sequence that stops input if the **input_flow_control** parameter is set to **start/stop**. The default is CTRL-S (^S).

ipencap_type

This Annex parameter specifies whether the RA 6300 LAN interface encapsulates IP packets in the Ethernet Version 2 format or the IEEE 802.3 Data Link Layer format. The values for this parameter are **ethernet** or **ieee802**. The default is **ethernet**.

ip_forward_broadcast

This Annex parameter allows an RA 6300 to broadcast a packet to the SLIP or PPP interfaces. When the RA 6300 receives a packet sent to a broadcast address (except 0.0.0.0 and 255.255.255.255), it scans the list of installed interfaces and matches the broadcast address against the interface's remote address using a subnet or net mask. If these addresses match, the RA 6300 copies the packet to that interface. When disabled, the RA 6300 does not scan the interface list and does not copy broadcast packets. A **Y** enables this parameter, an **N** disables it. The default is **N**.

ipso_class

This asynchronous port parameter specifies the U.S. Department of Defense basic IP Security Option (IPSO) classification level included in TCP packets generated locally on RA 6300 CLI, dedicated, or adaptive asynchronous ports.

The option is not added to locally generated ICMP messages, RIP updates, or other system packets. The RA 6300 does not check incoming packets for the presence of the IPSO. Valid options for the classification level are **topsecret**, **secret**, **confidential**, **unclassified**, and **none**. If you specify **none,** no IPSO classification is added. The default is **none**.

ipx_do_checksum

This Annex parameter controls whether or not the RA 6300 enables an IPX checksum. A **Y** enables this parameter, an **N** disables it. The default is **N**.

ipx_dump_password

This Annex parameter contains a user password for logging on to the Novell file server before the RA 6300 sends a dump file to the server. The string size ranges from 0 to 16 characters. The default is **“<unset>”**.

ipx_dump_path

This Annex parameter specifies the full pathname that stores the uploaded RA 6300 dump image on the Novell file server. The string size ranges from 0 to 100 characters. This parameter has no default value.

show_dump_username

This Annex parameter provides a user name for logging on to the Novell file server before the RA 6300 sends a dump file to the server. The string size ranges from 0 to 48 characters. This parameter has no default value.

ipx_file_server

This Annex parameter contains the name of the Novell file server from which the RA 6300 boots. The string size ranges from 0 to 48 characters. This parameter has no default value.

ipx_frame_type

This Annex parameter defines the framing used for IPX packets on the Ethernet interface. Valid options are **ethernetII**, **raw802_3**, **802_2**, or **802_2snap**. The default is **raw802_3**.

ipx_network

This PRI B channel parameter associates IPX network address(es) with B channel(s). This parameter works only when the **mode** parameter is set to **ppp**. The syntax is:

set pri b[=*channel_range* | **all**] **ipx_network** *net_number* [*increment*]

net_number is a 4-byte, Novell network number that the RA 6300 suggests for the remote PC client on an IPXCP (IPX over PPP) link. Valid values are **00000001** to **FFFFFFFF**, or **0**. Leading zeroes, if any, should be included. The network number must be unique on the network and on the RA 6300 itself.

When the IPXCP connection is established, the RA 6300 and the client negotiate the network number, each suggesting a value. The peer suggesting the highest number wins the negotiation, and the network number is set to that value. If both ends of the link set the network number to 0, a unique, randomly-generated number is used as the default.

This parameter is overridden by the value in the remote address field of the **acp_dialup** file, if that field is configured correctly.

The arguments that can be used with the **ipx_network** parameter are described in Table C-27.

Table C-22. Arguments used with ipx_network parameter

Argument	Description
channel_range	An integer specifying the number of a single B channel, a list of B channel numbers separated by commas, a range of B channel numbers separated by a hyphen, or the keyword **all**, which specifies all B channels. If not specified, 23 (for PRI/T1 lines) or 30 (PRI/E1 lines) IPX network numbers are assigned based on the value of increment.
net_number	The IPX network number to be assigned to the B channel if only one channel is specified in *channel_range*, or the IPX network number to be assigned to the first channel in *channel_range*.
increment	An integer specifying how *net_number* is to be incremented to automatically generate IPX network addresses for the B channels specified in *channel_range*. Use this for automatic IPX net address assignment. For example, For example, if you set *channel_range* to **1, 2, 3,** *net_number* to **00000001**, and increment to **2**, the RA 6300 assigns IPX node numbers **00000003**, **00000005**, and **00000007** to the B channels. The default increment is **0**.

ipx_node

This PRI B channel parameter is a string of 12 hexadecimal digits representing the 6-byte, non-zero node number the RA 6300 suggests for the node number of the remote PC client on an IPXCP (IPX over PPP) link. Valid values are **00-00-00-00-00-00** to **FF-FF-FF-FF-FF-FE**, except for multicast addresses. A multicast address is any address that has a 1 in the last bit of the first octet. For example, the Appletalk multicast address is 090007000000, of which the first octet (09) is 0000 1001 in binary; the rightmost 1 is the multicast indicator. The syntax for specifying this parameter is:

set pri b[=*channel_range* | **all**] **ipx_node** *node _number* [*increment*]

Table C-23 describes the arguments used in the previous command.

If the client suggests any valid value for the node number, that number will be used instead of the **ipx_node** value.

This parameter is overridden by the network number in the remote address field of the **acp_dialup** file, if that field is configured correctly. If the node number is not set in **acp_dialup** or through the **ipx_node** parameter, and no value is suggested by the client, the RA 6300 uses its own Ethernet address plus 1.

Table C-23. Arguments used with ipx_node parameter

Argument	Description
channel_range	An integer specifying the number of a single B channel, a list of B channel numbers separated by commas, a range of B channel numbers separated by a hyphen, or the keyword **all**, which specifies all B channels. If not specified, 23 (for PRI/T1 lines) or 30 (PRI/E1 lines) IPX node numbers are assigned based on the value of increment.
node_number	The IPX node number to be assigned to the B channel if only one channel is specified in *channel_range*, or the IPX node number to be assigned to the first channel in *channel_range*.
increment	An integer specifying how *node _number* is to be incremented to automatically generate IPX network addresses for the B channels specified in *channel_range*. For example, if you set *channel_range* to **1, 2, 3,** *node _number* to **00-00-00-00-00-01**, and *increment* to **4**, the RA 6300 assigns IPX node numbers **00-00-00-00-00-05**, **00-00-00-00-00-09**, and **00-00-00-00-00-0D** to the B channels. The default increment is **0**.

ipx_security

This asynchronous port parameter controls whether or not IPX security is enabled on the port. A **Y** enables this parameter, an **N** disables it. The default is **N**.

ixany_flow_control

This asynchronous port parameter treats any input character as a start (XON) character, if output has been suspended by a stop (XOFF) character. A **Y** enables this parameter, an **N** disables it. The default is **N**.

keep_alive_timer

This Annex parameter defines the number of seconds between the transmission of identification packets during times of network inactivity. This parameter works only for the LAT protocol. The packets serve only as notices to remote nodes that the host's services are available. Allowable values range from **10** to **255** (seconds). The default is **20** (seconds).

lat_key

This Annex parameter restricts access to LAT-related RA 6300 commands, parameters, functions, and the LAT protocol within the RA 6300. Each RA 6300 requires a unique key value (contact your supplier to obtain a LAT key). After setting the key, your system administrator must reboot the RA 6300.

lat_queue_max

This Annex parameter limits the number of HIC requests that the RA 6300 can queue. This parameter affects only the operation of HIC requests received after changing the parameter's value and setting LAT. Allowable values range from **1** to **255** or **none** (entering **none** sets the value to **255**). The default value is **4**.

latb_enable

This asynchronous port parameter enables the RA 6300 to decode a LAT hosts's data-b packet. Data-b packets change certain asynchronous port parameters (see your LAT host's documentation for more details). A **Y** enables this parameter, an **N** disables it. The default is **N**.

If **latb_enable** is set to **Y** and the LAT host sends a data-b slot message requesting that flow control (XON/XOFF) be turned off, the RA 6300 turns off flow control and passes XON/XOFF characters to the host. This scenario can adversely affect both XON/XOFF and the cursor keys on the terminal.

line_erase

This asynchronous port parameter allows an RA 6300 to echo line erase for a video terminal. When enabled, the RA 6300 erases all characters on the line and moves the cursor back to the beginning of the line. When disabled, the RA 6300 echoes the line erase character for hard-copy terminals, making the deleted line visible and positioning the print head at the beginning of the next line. The line erase occurs only at the CLI level. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

load_broadcast

This Annex parameter defines, during a boot, whether or not the RA 6300 requests the configuration or message-of-the-day files from other hosts on the network if any or all of the files are not available on the preferred load host. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

load_dump_gateway

This Annex parameter specifies the gateway's IP address. A gateway is required if the preferred load or dump host is on a different network or subnet than the RA 6300. The default is **0.0.0.0** (no gateway).

load_dump_sequence

This Annex parameter specifies available network interfaces (Ethernet or self) and the order in which they are used for a down-line load or an up-line dump. You can list more than one interface by using commas to separate interface names. Table C-24 describes the valid options.

Table C-24. Valid Options for the load_dump_sequence Parameter

Option	Description
net	For use with a local area network. This is the default value.
self	Instructs the RA 6300 to boot its image from the Flash ROMs. Since the RA 6300 cannot dump back to itself, when booting via **self**, always have a secondary load/dump interface by setting **load_dump_sequence** to **self,net**.

local_address

This asynchronous/synchronous global port parameter defines the IP address for the **asy**, **ta**, or **syn** port on the RA 6300 side of a link. This IP address is used only when the **mode** parameter is set to **slip** or **ppp**. The default is **0.0.0.0**.

location

This asynchronous port parameter defines an asynchronous port location or description that displays with the CLI **who** command. The string size ranges from 0 to 16 characters. The default is a **null string** ("").

lock_enable

This Annex parameter enables any port to use the RA 6300 interface for VMS Environment's **lock** command. A **Y** enables this parameter, an **N** disables it. The default is **N**.

login_password

This Annex parameter specifies the password for all ports using a VMS interface. The string size ranges from 0 to 16 characters. For security reasons, the RA 6300 displays this value as **"<set>"** or **"<unset>."** The default is **"<unset>."**

This parameter works only when **cli_interface** is set to **vci** and **login_port_password** is set to **Y**.

login_port_password

This asynchronous port parameter enables the port password when the VMS command interface is configured (i.e., when **cli_interface** is set to **vci**). A **Y** enables this parameter, an **N** disables it. The default is **N**.

login_prompt

This Annex parameter defines the prompt that appears for all ports using a VMS interface. The string size ranges from 0 to 16 characters. The default is the # symbol.

This parameter works only when **cli_interface** is set to **vci**.

login_timer

This Annex parameter specifies the number of minutes a port using a VMS interface can remain inactive. Valid values are **0** through **60** (minutes). Entering **0** sets the timer to 30 minutes. The default is **30**.

This parameter works only when **cli_interface** is set to **vci**.

login_timeout

This asynchronous port parameter enables a login timer when the VMS command interface is configured (i.e., when **cli_interface** is set to **vci**). A **Y** enables this parameter, an **N** disables it. The default is **N**.

long_break

This asynchronous port parameter enables an RA 6300 to return a user to the CLI prompt after receiving a break signal of more than two seconds. When disabled, the RA 6300 passes the break to the local application. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

loose_source_route

This Annex parameter controls the Loose Source Routing protocol which defines a sequence of IP addresses that a datagram must follow. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

When **loose_source_route** is enabled, the RA 6300 forwards all IP packets that have the *Loose Source Routing and Record* option set. The RA 6300 forwards all IP packets that have the *Strict Source Routing and Record* option set only if the next routing address is directly reachable by the RA 6300. Otherwise, the RA 6300 drops these packets and sends an ICMP type *Destination Unreachable* message with a code of *Source Route Failed*.

When **loose_source_route** is disabled, the RA 6300 will not forward any IP packets that have the *Strict Source Routing and Record* or *Loose Source Routing and Record* options set. The RA 6300 accepts these packets only if the RA 6300 itself is the ultimate destination. If the packets are not addressed to the RA 6300, they are dropped and the RA 6300 sends an ICMP type *Destination Unreachable* message with a code of *Source Route Failed* will to the originator.

Loose source routing can pose a security risk if you use filters on your network router. If you are concerned with security, set this parameter to **N**.

map_to_lower

This asynchronous port parameter enables an RA 6300 to convert uppercase characters sent from a terminal into lowercase characters. This conversion occurs only at the CLI level. Enable this parameter for older terminals that do not support lower case characters. A **Y** enables this parameter, an **N** disables it. The default is **N**.

map_to_upper

This asynchronous port parameter enables an RA 6300 to convert lowercase characters sent to a terminal into uppercase characters. This conversion occurs only at the CLI level. Enable this parameter for older terminals that do not support upper case characters. A **Y** enables this parameter, an **N** disables it. The default is **N**.

max_session_count

This asynchronous port parameter specifies the number of active sessions (jobs) allowed per port. The allowable values range from **1** to **16**. The default is **3**.

max_chap_chall_int

This Annex security parameter enables the RA 6300 to re-issue a CHAP challenge to a remote node at random times during the course of a PPP connection. The parameter itself specifies the maximum number of seconds in the interval from which the RA 6300 randomly chooses the times to re-issue the challenge. Valid values are **0** to **65535** (approximately 18.2 hours). For example, a value of 60 specifies an interval ranging from one to 60 seconds. The default is **0**, which disables the re-issuing of challenges.

max_vcli

This Annex parameter determines the maximum number of virtual CLI connections the RA 6300 can create at a time. Allowable values are the string **unlimited** or a decimal number from **0** to **254**. A value of **0** prevents any virtual CLI connections. The default is **unlimited**.

metric

This asynchronous/synchronous port parameter defines the hop count to the remote end of the asynchronous or synchronous line when the **mode** parameter is set to **slip** or **ppp**. Modify this parameter only if you want the RA 6300 to use a route other than the SLIP or PPP interfaces to the remote end. Allowable values are **1** to **15**. The default is **1**.

min_unique_hostnames

This Annex parameter determines whether or not you can identify a host in the host table by entering a minimal string rather than the full host name. A **Y** enables minimum uniqueness, an **N** disables it. The default is **Y**.

mode

This asynchronous port parameter sets the mode for access to an asynchronous port. Table C-25 describes the valid options; the default is **cli**.

Table C-25. Valid Options for the mode Parameter

Option	Description
arap	Allows a port to perform as a network interface using ARAP.
auto_detect	Allows a port to identify an incoming packet's protocol and to convert to IPXCP, PPP, ARAP or CLI.
cli	Allows a port connected to a terminal or incoming modem access to the CLI. The CLI provides access to the network and connections to other hosts via the **telnet**, **connect**, **rlogin** and **tn3270** commands. The **tn3270** command is available only if the **option_key** parameter is set to the correct value for the RA 6300.

(continued on next page)

Table C-25. Valid Options for the mode Parameter (continued)

Option	Description
connect	Allows a port to communicate with a LAT host via the **connect** command. This option works with the **dedicated_arguments** parameter.
ipx	Allows dial-in Novell access.
ppp	Allows a port to perform as a network interface using PPP. IP packets are encapsulated by PPP.
rlogin	Allows a port to communicate via the **rlogin** command.Use this option in conjunction with the asynchronous port parameter **dedicated_arguments**.
slip	Allows a port to perform as a network interface using SLIP. IP packets are encapsulated by SLIP.
telnet	Allows a port to communicate via the **telnet** command.Use this option in conjunction with the asynchronous port parameter **dedicated_arguments**.
tn3270	Allows a port to communicate via the **tn3270** command. Use this option in conjunction with the asynchronous port parameter **dedicated_arguments**.
unused	The port has no connection; the RA 6300 ignores it.

You should mark a port as unused if it is not connected to a device.

mop_password

This Annex parameter contains the MOP maintenance password. In this 8-byte password, each byte consists of two hexadecimal digits. The string size ranges from 0 to 16 characters. For security reasons, the RA 6300 displays values as “<set>” or “<unset>.” The default is **“<unset>.”**

motd_file

This Annex parameter defines the file name for the message-of-the-day file maintained on the load host. The string size ranges from **0** to **16** characters. The default file name is **motd**; the file resides in the directory chosen during the host installation process (typically **/usr/spool/erpcd/bfs**).

multicast_timer

This Annex parameter defines the number of seconds that can elapse between service announcement transmissions for the LAT protocol. Allowable values range from **10** to **180** (seconds). The default is **30**.

multisessions_enable

This Annex parameter allows multisessions to be managed on a terminal server basis. When enabled, terminals that support DEC's Terminal Device/Session Management Protocol (TD/SMP) can display two active windows simultaneously over one communication line. A **Y** enables this parameter, an **N** disables it. The default is **N**.

name_server_1

This Annex parameter defines the type of name service used with the primary name server. When using this parameter, you must specify a host using the **pref_name1_addr**. The options are **dns, ien_116**, or **none**. The default is **none**.

name_server_2

This Annex parameter defines the type of name service used with the secondary name server. The service type specified with this parameter is queried if the type specified with **name_server_1** is not available. When using this parameter, you must specify a host using **pref_name2_addr**. The options are **dns**, **ien_116**, or **none**. The default is **none**.

nameserver_broadcast

This Annex parameter defines whether or not the RA 6300 broadcasts a name server request if the preferred name servers do not respond. A **Y** enables this parameter, an **N** disables it. The default is **N**.

need_dsr

This asynchronous port parameter allows an RA 6300 to use the DSR (Data Set Ready) signal to determine whether a device is attached to the corresponding asynchronous line. The RA 6300 will not allow connection to a slave port and will not activate the CLI until the DSR signal is active. If DSR is deactivated, the connection to a slave line is terminated and the CLI is deactivated.

When disabled, the DSR signal is not required to connect to a slave line, and DSR is not required to activate a CLI line (see *Modem Assignments* on page A-81 for details on using **need_dsr** in conjunction with modem control). DSR is always considered active on ports that do not have a DSR signal (ports with partial modem control lines). A **Y** enables this parameter, an **N** disables it. The default is **N**.

net_inactivity

This asynchronous/synchronous port parameter defines the amount of time that network protocols (e.g., SLIP, PPP, ARAP) running on the serial line can remain inactive before the port is reset. Valid options are **0** to **255**; a zero value indicates *off*. The default is **0** (*off)*.

Use this parameter in conjunction with **net_inactivity_units**.

Since the accuracy of the inactivity timer is within five seconds, if **net_inactivity_units** is set to seconds, we recommend using a value for **net_inactivity** that is a multiple of five.

net_inactivity_units

This asynchronous/synchronous port parameter defines the units of time used for the port's inactivity timer. Valid options are **minutes** or **seconds**. The default is **minutes**.

Use this parameter in conjunction with **net_inactivity**.

If you are using an ISDN line, it is important to set **net_inactivity** to a small value, such as **30,** and **net_inactivity_units** to **seconds**; otherwise, you may incur high costs from your ISDN service provider.

To set an inactivity timer of two minutes, set:

net_inactivity_units=**minutes**
net_inactivity=**2**

To set an inactivity timer of 30 seconds, set:

net_inactivity_units=**seconds**
net_inactivity=**30**

An inactivity timer can be set to the following combinations:

0 to 4 minutes 15 seconds (= 255 seconds) in 1 second intervals
0 to 4 hours 15 minutes (= 255 minutes) in 1 minute intervals

network_turnaround

This Annex parameter defines the approximate number of seconds that an RA 6300 waits for a response from a security server (an algorithm defines the actual time which typically is longer than the defined value). This parameter works only when the **enable_security** parameter is set to **Y**. Allowable values range from **1** to **255**. The default is **2**.

Setting this parameter to a high number is not recommended unless a large timeout value is required for contacting (for security) slow hosts or waiting for a slow host's response to a security request.

newline_terminal

This asynchronous port parameter interprets carriage returns and line feeds at the CLI level. When enabled, a line feed terminates both the input and the output lines. When disabled, a carriage return or a line feed terminates the input line and a carriage return followed by a line feed terminates output lines. A **Y** enables this parameter, an **N** disables it. The default is **N**.

node_id

This Annex parameter specifies the address the RA 6300 tries to acquire at start-up. If this address is in use, the RA 6300 must acquire a new node ID. The **node_id** is an AppleTalk address in the form *net.node*. Valid *net* values are **0** to **65534**; valid *node* values are **0** to **254**. The default is **0.0**.

num_b_channels

This PRI line parameter specifies the maximum number of B channels that can be used with the RA 6300. Valid values are **1–23** for T1/PRI and **1–30** for E1/PRI. The default is **0**, which the RA 6300 interprets as **23** for T1/PRI connections (in the U.S.) and **30** for E1/PRI connections (in Europe and Australia). The only time you might want to change the default is in Europe, where telephone company providers support PRI connections with fewer than 30 channels (which can be cheaper.)

option_key

This Annex parameter enables the following features:

- The AppleTalk-specific functions and parameters.
- The CLI **tn3270** command.
- Filtering and dial-out
- IPX.

These features are available separately or in any combination. Each RA 6300 requires a unique key value for the feature(s) you choose; contact your supplier to obtain an **option_key** value.

output_flow_control

This asynchronous port parameter defines the method that a device uses to stop output from an RA 6300. Table C-26 describes the valid options; the default is **start/stop**.

Table C-26. Valid Options for the output_flow_control Parameter

Option	Description
bell	Comparable to setting the parameter to **none**.
eia	Selects hardware flow control; **eia** works only if the **control_lines** parameter is set to **flow control** or **both** and the device is wired properly.
start/ stop	Specifies XON/XOFF flow control (independent of the **control_lines** parameter). Upon receiving XOFF (**output_stop_char**), the RA 6300 stops sending output to the device. Upon receiving XON (**output_start_char**), the RA 6300 starts sending output to the device. The RA 6300 removes these characters from the data stream.
both	Specifies both in-band (XON/XOFF) and out-of-band (CTS/RTS) flow control (if **control_lines** is set to **flow_control** or **both**). Both flow controls are independent; data flows out of the port only if CTS is high and the last received character was XON. Receiving XOFF or dropping CTS stops output (RA 6300 to device) flow.
none	Specifies no flow control; characters are lost if the buffers overflow.

output_is_activity

This asynchronous port parameter defines activity as output. When enabled, the RA 6300 resets the inactivity timer when it sends output from the port. If the line **type** is **hardwired**, the RA 6300 also places an entry in the **who** table. A **Y** enables this parameter, an **N** disables it. The default is **N**.

output_start_char

This asynchronous port parameter defines the control character sequence that restarts output if **output_flow_control** is set to **start/stop**. The default is **CTRL-Q** (^Q).

output_stop_char

This asynchronous port parameter defines the control character sequences that stops output if the **output_flow_control** parameter is set to **start/stop**. The default is **CTRL-S** (^S).

output_ttl

This Annex parameter sets the time-to-live (TTL) for packets the RA 6300 generates for RIP updates. TTL is a field in IP packets that limits their lifetime on the network. Each time a packet crosses a router, the router decrements the packet’s TTL by 1. When the value reaches 0, the packet is discarded. Allowable values range from **1** to **255**. The default is **64**.

parity

This asynchronous port parameter defines the type of parity that the asynchronous port uses. The options are **even**, **odd**, or **none**. The default is **none**.

password

This Annex parameter modifies the RA 6300's administrative password. This password is used for access to the superuser CLI commands and for administrative access to an RA 6300. It overrides the CLI **lock** and virtual CLI passwords. The string size ranges from 0 to 15 characters.

If the RA 6300 is configured with an IP address, the default administrative password is the RA 6300's IP address in dotted-decimal notation.

If the RA 6300 is not yet configured with an IP address and the administrative password has not been modified (either via this parameter or via the CLI **passwd** command), the default password is a null string ("").

If the RA 6300 is not configured with an IP address and boots via MOP, IPX, or from FLASH ROM, the default password is a null string ("") and entering a carriage return at the *Password* prompt places you in superuser mode.

Even if **password** is not set, the default administrative password is required to access the superuser CLI commands.

Changes to this parameter take effect immediately.

passwd_limit

This Annex parameter defines the maximum number of times a user can try to enter a password before an RA 6300 resets the port. Entering zero sets the limit to 3. Allowable values range from **0** to **10** (entering **0** sets the value to the default). The default is **3**.

port_password

This asynchronous/synchronous port parameter defines an **asy**, **ta**, or **syn** port password for local password protection. You can use this password as a back-up for host-based security if the security servers do not respond or as an additional line of security after entering a user name password.

When using SecurID, set **port_password** to a null string ("") and do not set a port password in the **acp_passwd** file (see *Using the SecurID Card* on page A-303 for more details).

port_server_security

This asynchronous port parameter enables a host-based security policy for access to the port through the port server. When enabled, only authorized users can access the port. A **Y** enables this parameter, an **N** disables it. The default is **N**.

ppp_acm

This asynchronous port parameter (async control mask) specifies which of the first 32 bytes (0x0 to 0x1F) of the ASCII character can be sent as clear text and which should be protocol-escaped.

The RA 6300 requests the **ppp_acm** parameter as its local mask. If the peer rejects **ppp_acm**, the RA 6300 accepts the hint if it is a superset of the RA 6300's mask; otherwise, it uses the PPP default of 0xFFFFFFFF. The RA 6300 accepts any mask from the peer. Values range from **0x00000000** to **0xffffffff**. The RA 6300 default is **0x00000000**.

The **ppp_acm** parameter is a bit mask that is set as follows:

- **ppp_acm** for ASCII NUL (decimal 0) is 2 to the power of 0 = 0x00000001
- **ppp_acm** for ASCII SOH (decimal 1) is 2 to the power of 1 = 0x00000002
- **ppp_acm** for ASCII DC1 (decimal 17) is 2 to the power of 17 = 0x00020000
- **ppp_acm** for ASCII DC3 (decimal 19) is 2 to the power of 19 = 0x00080000

Thus, the mask for XON/XOFF (DC1 and DC3) equals the OR function of 0x00020000 and 0x00080000, or 0x000a0000.

When the RA 6300 sends an ACCM to the host, it follows this calculation to determine the initial value requested:

- The value set for **ppp_acm** (a 32-bit integer) is read in as the ACCM.

- If **input_flow_control** is set to **start/stop**, the following two additions are made:

  If **input_start_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

  If **input_stop_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

- If **output_flow_control** is set to **start/stop**, the following two additions are made:

  If **output_start_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

  If **output_stop_char** is 0–31 decimal, the bit indexed by this parameter is set in the ACCM.

For example, the initial ACCM sent to the peer is 0x000A0001 if **ppp_acm** is set to 0x00000001 (i.e., the ASCII NUL character will not be sent) and the following parameters are set as indicated:

input_flow_control	**start/stop**
input_start_char	**^S**
input_stop_char	**^Q**
output_flow_control	**start/stop**
output_start_char	**f**
output_stop_char	**h**

Since the output flow control parameters are outside the range 0–31 decimal, they do not affect the ACCM.

The **na/admin** command **show port ppp_acm** still displays the **ppp_acm** setting. The CLI command **netstat –ip***nn*, where *nn* is the port number, displays the true mask (ACCM) value, i.e., the value negotiated between the two PPP processes.

ppp_mru

This asynchronous/synchronous port parameter defines the maximum receive unit (MRU) that the RA 6300 requests as its local MRU. If NAKed and the remote hint is less than this value, the RA 6300 accepts the hint; otherwise, the RA 6300 requests the PPP default (**1500**). Values range from **64** to **1500**. The default is **1500**.

ppp_ncp

This asynchronous/synchronous port parameter specifies the network control protocols that run on the interface. The RA 6300 negotiates for these protocols only. Valid settings are one or more of the following: **ipcp** (Internet Protocol Control Protocol), **atcp** (AppleTalk Control Protocol), **ipxcp** (Internet Packet Exchange Control Protocol), **mp** (Multilink PPP), and **ccp** (Compression Control Protocol). Separate multiple protocols with commas. You can also specify **all** to indicate all of the protocols, which is the default.

For information on Multilink PPP, see the *Multilink PPP Addendum to the Remote Annex Administrator's Guide for UNIX*.

ppp_password_remote

This asynchronous/synchronous port parameter defines a PPP port user's password. The string size ranges from 0 to 16 characters. For security reasons, the RA 6300 displays this parameter's value as "<set>" or "<unset>." The default is **"<unset>."**

ppp_sec_auto

This asynchronous/synchronous parameter, used in conjunction with the **ppp_security_protocol** parameter, allows you to use **auto_detect** mode for PPP clients whether or not the clients support PAP/CHAP. Valid values are **Y** and **N**. If **ppp_sec_auto** is set to **Y** and **enable_security** is set to **Y**, and if the user accesses the RA 6300 in CLI mode (e.g., by entering a carriage return after dialing in) and then switches to **ppp** mode (by issuing the **ppp** command), the RA 6300 treats the **ppp_security_protocol** as if it were set to **none**.

If **ppp_sec_auto** is set to **Y** and **enable_security** is set to **Y**, and if the RA 6300 determines (via a **mode** parameter of **auto_detect**) that a dial-in user is using PPP, the RA 6300 uses the current value of **ppp_security_protocol**.

The default for **ppp_sec_auto** is **N**, which specifies that no matter how the user is placed in **ppp** mode, the RA 6300 interprets **ppp_security_protocol** as described next.

ppp_security_protocol

This asynchronous/synchronous port parameter defines the security check for the peer that the RA 6300 requires before starting the network control protocol. If the RA 6300 wants to use security and the peer refuses, the RA 6300 closes the link. Valid options are **chap** (challenge-handshake protocol), **pap** (password authentication protocol), **chap-pap**, and **none**. The default is **none**.

LCP requests for CHAP received by the RA 6300 are always ACKed, regardless of this parameter’s setting.

ppp_username_remote

This asynchronous/synchronous port parameter defines the user name by which the RA 6300 identifies itself when the remote PPP peer asks for authentication. The string size ranges from 0 to 15 characters. The default is a **null string** ("").

pref_dump_addr

This Annex parameter specifies the IP address for the preferred dump host. This is the host to which the RA 6300 first tries to dump. The default is **0.0.0.0**.

A dump is not sent if the address is set to the default value.

pref_load_addr

This Annex parameter specifies the IP address for the preferred load host. This is the host to which the RA 6300 first requests a load of its operational code. The default is **0.0.0.0**.

Set this address to the boot host’s IP address.

pref_mop_host

This Annex parameter specifies the Ethernet address of the preferred MOP load or dump host. This address consists of six parts separated by dashes. Each part contains a hexadecimal value. The default value is **00-00-00-00-00-00**.

pref_dhcp1_addr

This optional Annex parameter specifies the IP address of the DHCP server that a DHCP client will attempt to discover as the primary source for DHCP services. A DHCP client will broadcast a DHCP message when the **dhcp_broadcast** parameter has been set to Y. The value for **pref_dhcp1_addr** may be set to **0**.

pref_dhcp2_addr

This optional Annex parameter specifies the IP address of the DHCP server that a DHCP client will attempt to discover as a backup source for DHCP services when the primary DHCP server does not respond. A DHCP client will broadcast a DHCP message when the **dhcp_broadcast** parameter has been set to Y. The address specified by **pref_dhcp2_addr** will be used only if **pref_dhcp1_addr** is non-zero and does not respond. The value for **pref_dhcp1_addr** may be set to **0**.

pref_name1_addr

This Annex parameter defines the IP address of the host specified in the **name_server_1** parameter. The default is **0.0.0.0**.

pref_name2_addr

This Annex parameter specifies the IP address of the host specified in the **name_server_2** parameter or a back-up host that serves if **name_server_2** is set to **none**. The default is **0.0.0.0**.

pref_secure1_host

This Annex parameter specifies the IP address of the security server to which the RA 6300 first sends security requests. This parameter works only if the **enable_security** parameter is set to **Y**. The default is **0.0.0.0**.

pref_secure2_host

This Annex parameter specifies the IP address of the host that is the back-up server if the host specified in **pref_secure1_host** is not available. This parameter works only if the **enable_security** parameter is set to **Y**. The default is **0.0.0.0**.

printer_host

This asynchronous port parameter specifies the IP address or fully qualified domain name of a machine running a Berkeley-style *lpd* server. The **tn3270** command uses this server for the print-screen function.

printer_name

This asynchronous port parameter specifies the printer used by the **tn3270** command's print-screen function. You must enter a name listed in the **/etc/printcap** file on the remote host by the **printer_host** parameter .

prompt

This asynchronous port parameter defines a port-specific prompt string. The prompt string consists of displayable characters and embedded formatting codes. Each formatting code, which consists of a percent character (%) followed by a single character, is compressed and stored as a single character in non-volatile memory. The maximum number of characters stored for the prompt string is 16. Since each formatting code consists of two characters, the maximum string size is 32 characters. String sizes smaller than 32 characters are rejected as bad values if they cannot be stored into 16 characters in non-volatile memory after the formatting codes are compressed into single characters. Table C-20 on page C-48 lists and describes these codes. The RA 6300 parameter **cli_prompt** defines the default prompt.

redisplay_line

This asynchronous port parameter defines the reprint line character for CLI users. The allowable value is a control character sequence. The default is **CTRL-R** (^R).

remote_address

This PRI B channel parameter associates IP address(es) with B channel(s). This parameter works only when the **mode** parameter is set to **slip** or **ppp**. The syntax is:

set pri b[=*channel_range* | **all**] **remote_address** *ip_addr* [*increment*]

Table C-27 describes the arguments used with **remote_address**.

If **address_origin** is set to **acp**, the remote address specified in the **acp_dialup** file overrides the remote address described here.

Table C-27. Arguments for the remote_address Parameter

Argument	Description
channel_range	An integer specifying the number of a single B channel, a list of B channel numbers separated by commas, a range of B channel numbers separated by a hyphen, or the keyword **all**, which specifies all B channels. If not specified, 23 (for PRI/T1 lines) or 30 (PRI/E1 lines) IP addresses are assigned based on the value of increment.
ip_addr	The IP address to be assigned to the B channel if only one channel is specified in *channel_range*, or the IP address to be assigned to the first channel in *channel_range*.
increment	An integer specifying how *ip_addr* is to be incremented to automatically generate IP addresses for the B channels specified in *channel_range*. Use this for automatic IP address assignment. If *ip_addr* is set to 132.245.22.2, increment is set to 2, and 3 B channels are specified, the RA 6300 assigns IP addresses 132.245.22.2, 132.245.22.4, and 132.245.22.6 to the B channels. The default increment is **0**.

reset_idle_time_on

This asynchronous port parameter defines whether **input** or **output** resets the idle timer. The idle time is the time lapse between activity and inactivity at the device. This parameter is used with the **who** command. Valid options are **input** and **output**. The default is **input**.

retrans_limit

This Annex parameter defines the number of times an RA 6300 retransmits a packet before notifying the LAT user about a network failure. Allowable values range from **4** to **120**. The default value is **8**.

rip_accept

This interface parameter defines the networks for which the RA 6300 accepts advertised routes. Table C-28 lists the valid options; the default is **all**.

Table C-28. Valid Options for the rip_accept Parameter

Option	Description
access_spec	Uses the form [**include** | **exclude**] *network_list* where **include** means accept RIP updates only for the networks in *network_list*, and **exclude** means accept all RIP updates except for those in *network_list*. You can list up to eight network IP addresses in *network_list*.
none	No RIP updates are accepted over the interface.
all	RIP updates for all networks are accepted.

rip_advertise

This interface parameter defines the networks for which the RA 6300 will advertise routes. Table C-29 lists the valid options; the default is **all**.

Table C-29. Valid Options for the rip_advertise Parameter

Option	Description
access_spec	Uses the form [**include** | **exclude**] *network_list* where **include** means advertise only the networks in *network_list*, and **exclude** means advertise all networks except those in *network_list*. The list can contain up to eight network addresses.
none	Turns off advertising for the interface.
all	Advertises all networks over the interface.

rip_auth

This Annex parameter contains the password that controls authentication for RIP 2 packets. The string size ranges from 0 to 16 characters. The RA 6300 displays this parameter's value as "<set>" if a password is entered or "<unset>" if a null string is entered. When "<unset>, " authentication is turned off and all RIP packets are accepted. The default is a **null string** ("").

rip_default_route

This interface parameter allows an RA 6300 to advertise that it is the default router. Valid values are **0** through **15**, or **off**. A value of 1 through 15 indicates the hop count that will be advertised. A value of 0 or off turns off the advertisement. The default is **off**.

rip_horizon

This interface parameter controls the split horizon algorithm for RIP. Table C-30 describes the valid options for this parameter. The default is **poison**.

Table C-30. Valid Options for the rip_horizon Parameter

Option	Description
off	Disables split horizon.
split	Enables split horizon without poison reverse.
poison	Enables split horizon with poison reverse.

rip_next_hop

This interface parameter specifies whether or not the next hop value is included in RIP version 2 advertisements. Valid options are **never**, **needed**, or **always**. The default is **needed**.

rip_recv_version

This interface parameter controls the RIP version(s) that an RA 6300 accepts. Table C-31 describes the valid options for this parameter. The default is **both**.

Table C-31. Valid Options for the rip_recv_version Parameter

Option	Description
1	Only version 1 packets are accepted.
2	Only version 2 packets are accepted.
both	Both versions are accepted.

rip_routers

This Annex parameter lets you force RIP to direct periodic RIP updates to a router list rather than broadcasting updates. Valid values are the IP addresses of up to eight directly reachable routers. The RA 6300 ignores any address that is not on an attached subnet. Specifying the default, **all**, restores broadcasting.

rip_send_version

This interface parameter controls the RIP version(s) that an RA 6300 sends over the IP interface(s). Table C-32 describes the valid options for this parameter. The default is **compatibility**.

Table C-32. Valid Options for the rip_send_version Parameter

Option	Description
1	Version 1 packets are sent to the broadcast address.
2	Version 2 packets are sent to the RIP multicast address.
compatibility	Version 2 packets are sent to the broadcast address.

rip_sub_accept

This interface parameter controls whether or not subnet routes are accepted over the SLIP, PPP, and Ethernet interfaces. When enabled, subnet routes are accepted; when disabled, subnet routes are rejected. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

rip_sub_advertise

This interface parameter controls whether or not the RA 6300 advertises subnet routes over the SLIP, PPP, and Ethernet interfaces. When enabled, subnet routes are advertised; when disabled, subnet routes are not advertised. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

routed

This Annex parameter determines whether or not the RIP routing daemon is enabled. When enabled, the RA 6300 performs active RIP routing only if the **option_key** parameter is set to the correct value. If **option_key** is not set correctly, the RA 6300 performs only passive RIP routing when the daemon is enabled. When disabled, no RIP routing occurs. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

rwhod

This Annex parameter determines whether or not the RA 6300 listens for RWHO broadcasts when it builds the host table. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

security_broadcast

This Annex parameter determines whether or not the RA 6300 broadcasts for security validation if the preferred security servers are not available. When enabled, the RA 6300 broadcasts for security; when disabled, the RA 6300 does not broadcast for security. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

server_capability

This Annex parameter defines an RA 6300 as a file server host. An RA 6300 can provide operational code only for another RA 6300. Table C-33 describes the valid options; the default is **none**.

Table C-33. Valid Options for the server_capability Parameter

Option	Description
all	The RA 6300 is a file server for the configuration, operational image, and message-of-the-day files.
config	The configuration files.
image	The RA 6300's operational code.
motd	The message-of-the-day file.
none	The RA 6300 is not a file server.

server_name

This Annex parameter names the RA 6300 in the LAT protocol. The name should match the NMS host's node name used in the HIC configuration file. The string size ranges from 1 to 16 characters. The default value is the physical Ethernet address, represented as a hexadecimal value, appended to the string **LAT_** (for example, *LAT_080002BF0020*).

service_limit

This Annex parameter defines the maximum number of LAT services that an RA 6300 can maintain in its local service table. When the table is full, the RA 6300 removes the service that has been idle longest. If all services are busy and the table is full, the RA 6300 discards a new service. Allowable values range from **16** to **2048**. The default is **256**.

session_limit

This Annex parameter specifies the maximum number of active sessions the RA 6300 allows at one time. Allowable values range from **1** to **1152** or **none** (entering **none** sets the value to **1152**). The default is **1152**.

short_break

This asynchronous port parameter allows an RA 6300 to return a user to the CLI prompt after receiving a break of less than two seconds. This occurs only at the CLI level. A **Y** enables this parameter, an **N** disables it. The default is **Y**.

slip_mtu_size

This asynchronous port parameter sets the maximum transmission unit (MTU) size on a SLIP/CSLIP port. This parameter forces the SLIP interface to use **large** (1006) or **small** (256) MTUs. The default is **small**.

slip_no_icmp

This asynchronous port parameter controls whether or not the RA 6300 discards any ICMP packets directed to the SLIP link. When enabled, the RA 6300 reduces unnecessary traffic and messages over the SLIP link. A **Y** enables this parameter, an **N** disables it. The default is **N**.

slip_ppp_security

This asynchronous port parameter controls dial-up SLIP/PPP access. When this parameter and the **enable_security** parameter are enabled, the RA 6300 determines whether or not the user at the CLI is authorized to execute a **slip** or **ppp** command. A **Y** enables this parameter, an **N** disables it. The default is **N**.

slip_tos

This asynchronous port parameter allows an RA 6300 to send interactive traffic (**telnet**, **rlogin**, and **ftp** control sessions) before sending any other traffic. This parameter provides a type-of-service based SLIP queuing. A **Y** enables this parameter, an **N** disables it. The default is **N**.

stop_bits

This asynchronous port parameter specifies the number of stop bits for a port. Allowable values are **1**, **1.5**, or **2**. The default is **1**.

subnet_mask (RA 6300)

This Annex parameter defines the RA 6300's IP subnet mask. It is used to divide a network into subnets. The parameter's default is based on the network portion of the RA 6300's IP address.

Setting this parameter incorrectly can cause routing problems.

subnet_mask (port)

This port parameter defines the subnet mask for an asynchronous SLIP port or a synchronous PPP interface. Typically, you use this parameter to divide a network into subnets. Specifying **0.0.0.0**, which is the default, sets the subnet mask to **255.255.255.255**, which denotes a non-subnetted host address.

Setting this parameter incorrectly can cause routing problems.

sys_location

This Annex parameter supplies LAT host location or identification information. The string size ranges from 0 to 32 characters. The default is a **null string** ("").

switch_type

This PRI line parameter is a string specifying the type of switch provided by the telephone company for the PRI line. Valid values are **AT9** (for the AT&T 5ESS switch), **DMS** (for Nortel’s DMS100 switch), **NI2** (for a switch supporting National ISDN2), and **ETS** (for Euro ISDN, NET5 conformance).

The default is `""`, a blank string. The RA 6300 interprets this as the switch type appropriate for the line:

- For a T1 line, AT9 is used.
- For an E1 line, ETS is used.

The switch type parameter is not case-sensitive.

syslog_facility

This Annex parameter defines the local facility to which the UNIX *syslogd* daemon sends RA 6300 syslog messages. Valid options are **log_local0** through **log_local7**. The default is **log_local7**.

If the host to which messages are logged does not support 4.3BSD logging, this parameter is ignored and messages are logged by priority level (defined by **syslog_mask**).

syslog_host

This Annex parameter defines the IP address of the host that logs RA 6300 messages. The default, **0.0.0.0**, causes the RA 6300 to broadcast its log messages.

syslog_mask

This Annex parameter defines the priority levels that the RA 6300 logs. The options are **all**, **none**, or a combination of levels separated by commas. The default, **none**, disables logging. Table C-34 lists the levels in priority order (see *Event Logging Using syslog* on page B-37 for more details on using *syslog* for event logging).

Table C-34. Priority Levels for the syslog_mask Parameter

Priority Level	Description
emergency	Hardware failures.
alert	All RA 6300 reboots.
critical	Configuration and initialization problems, such as format errors in the configuration file or lack of memory.
error	All line initialization errors, including CLI.
warning	Indications of minor problems.
notice	Time server queries and information about responses.
info	Start and end CLI sessions and RA 6300 jobs created by the **rlogin**, **telnet**, **connect**, **ping**, and **tap** commands.
debug	Activate and exit all RA 6300 processes.

tcp_keepalive (RA 6300)

This Annex parameter specifies the length of time a TCP connection must be idle before an RA 6300 sends keep-alive messages. A keep-alive message contains no data but solicits an acknowledgment from the other end of a connection to determine whether the connection is still active. If the recipient does not acknowledge the message after eight retries, the RA 6300 drops the connection. Valid values are **0** through **255** (minutes). A value of **0** sets the keep-alive time to **120** minutes, which is the default; a value of **255** disables the keep-alive mechanism. The **tcp_keepalive** parameters for serial-line ports and parallel ports override this parameter for those individual ports.

tcp_keepalive (asynchronous)

This asynchronous port parameter specifies the length of time a TCP connection must be idle on one or more specific asynchronous ports. This parameter overrides the RA 6300 **tcp_keepalive** parameter for connections to the host from adaptive, CLI, and dedicated ports, and for connections from the host to slave and adaptive ports. Valid values are **0** to **255** (minutes). The default is **0**. Entering a zero specifies that the keep-alive time is the value set in **tcp_keepalive**; entering 255 disables the keep-alive mechanism for the port.

telnet_crlf

This asynchronous port parameter converts a carriage return in a Telnet session to a carriage return followed by a line feed. When disabled, a carriage return translates to a carriage return followed by a null string. A **Y** enables this parameter, an **N** disables it. The default is **N**.

telnet_escape

This asynchronous port parameter defines the character that returns a CLI user to the *telnet* prompt. Setting this parameter to **U** disables the Telnet escape character. The default is **CTRL-**] (^]).

term_var

This asynchronous port parameter identifies the type of terminal using the CLI connection. You must enter a valid terminal type for the host. The RA 6300 passes the terminal type setting to the host. The string size ranges from 0 to 16 characters. The default is a **null string** ("").

tftp_dump_name

This Annex parameter provides the file name used to dump an RA 6300's core image via **tftp** if the RA 6300 operational image and **erpcd** fail. The parameter must include the entire path of the dump file, including parent directories. The file you enter must have read and write permissions.

tftp_load_dir

This Annex parameter defines the string that precedes all files (e.g., image name, configuration, and **motd** files) when you boot an RA 6300 via **tftp**. This string's value is determined by the system serving the **tftp** requests. This string does not precede the **tftp_dump_name**.

time_broadcast

This Annex parameter defines whether the RA 6300 broadcasts for the time if the preferred load host is not available or does not provide a time server. A **Y** enables this parameter, an **N** disables it. The default is **N**.

time_server

This Annex parameter determines whether or not the RA 6300 queries for time service. Table C-35 lists the IP addresses to which the network administrator can set this parameter. The default is **0.0.0.0**.

Table C-35. IP Addresses for the time_server Parameter

Address	Description
loopback address	Do not query for time service.
0.0.0.0	Query the boot host for time service.
127.0.0.1	Do not send out direct time queries.
host address	Send queries to given host address.
broadcast address	Send queries to given broadcast address — may require *direct broadcast* service on the network routers. This setting does not respect the **time_broadcast** parameter setting.

timezone_minuteswest

This Annex parameter defines the time zone in which the RA 6300 resides. Enter a positive number of minutes for time zones west of GMT, or a negative number for time zones east of GMT. For example, enter *300* for U.S. Eastern Standard Time, which is five hours west of GMT, or *-60* for Paris, which is one hour east of GMT. The default is **300**.

tmux_delay

This Annex parameter defines the maximum number of milliseconds during which small packets can accumulate to form larger packets. When the time expires, the RA 6300 sends the multiplexed packet. Valid values are **0** through **255** (milliseconds). Entering *0* sets this parameter to 20. The default is **20**.

tmux_enable

This Annex parameter controls whether or not an RA 6300 uses TMux to multiplex small TCP packets into a single IP packet. This parameter works only if the host supports TMux. When enabled, and the host does not support TMux, the RA 6300 will not support multiplexing. A **Y** enables this parameter, an **N** disables it. The default is **N**.

tmux_max_host

This Annex parameter specifies the maximum number of host addresses allowed in the TMux address table. If the number of host addresses exceeds the value entered here, the RA 6300 discards the oldest entry. Allowable values are **10** through **255**; the default is **64**.

tmux_max_mpx

This Annex parameter specifies the largest user packet that can be placed in a TMux packet. The RA 6300 does not multiplex larger packets, but passes them directly to the IP layer. Allowable values are **5** through **65535**; the default is **700**.

toggle_output

This asynchronous port parameter defines the character that flushes the output buffer for CLI users. The flush character must be a CTRL-*X* (^*X*), where *X* represents an alphanumeric value (not case sensitive). Pressing this character flushes the output buffer. The default is **CTRL-O** (^O).

user_name

This asynchronous/synchronous port parameter defines an asy or sync port's user name as a string. The CLI **who** command displays this value; the CLI **rlogin** command passes this value to a host. The default is a **null string** ("").

v120_mru

This asynchronous parameter applies exclusively to V.120 (TA) connections. It allows you to change the number of bytes allowed in an incoming V.120 frame if your TA cannot handle the default of **256** bytes. Valid values are **30–260**.

vcli_groups

This Annex parameter specifies which LAT remote group code is assigned to virtual CLI users. All virtual CLI users have the same group code. Values are specified as a series of numbers separated by commas (e.g., 1,5,7) or a range of numbers separated by a dash (e.g., 200-255). Following the range, specify the keyword **enable** or **disable**. Allowable values are **all**, **none**, or numbers between **0** and **255**. Following the range, specify the keyword **enable** or **disable**. The default is **none enable**.

vcli_inactivity

This Annex parameter specifies the number of minutes that a virtual CLI connection can remain inactive. If the number of minutes is exceeded, the RA 6300 terminates the virtual CLI connection. Valid values are **1–255** or **off.** The default is **off**, which specifies that no time limit is imposed on VCLI connections.

vcli_password

This Annex parameter defines a password required for virtual CLI connections to the RA 6300. The string size ranges from 0 to 15 characters. This parameter is useful for local password protection and as a back-up to host-based security. For local password protection, set the **enable_security** parameter to **Y**, set the **vcli_security** parameter to **N**, and define a password for this parameter. As a back-up for host-based security, setting this parameter causes the RA 6300 to request a password on a virtual CLI connection whenever the security server does not respond. The default is **"<unset>"**.

Changes to this parameter take effect immediately.

vcli_security

This Annex parameter enables user validation on virtual CLI connections to and from an RA 6300 for the duration of the connection. When enabled, the RA 6300 enables connection security for all virtual CLI connections and executes the same user validation, including user name and password, that it uses with CLI security on asynchronous ports. This parameter works with host-based security only when the enable_security parameter is set to Y. A Y enables this parameter, an N disables it. The default is N.

zone

This Annex parameter defines the AppleTalk zone name that the RA 6300 uses at start-up. The string size ranges from 0 to 32 characters. You must separate zone names with spaces (e.g., general pubs lab). To escape embedded spaces, use the backslash (\) character. The default is a **null string** ("").

Chapter 3
Using the CLI Commands

This chapter in the *Remote Annex Administrator's Guide for UNIX* describes the Command Line Interpreter (CLI) commands. When using that chapter with an RA 6300, note that:

- The following commands do not apply to this release of the RA 6300: **stats –T** (for the RA 6100), **stats –p**, **t1_loopback**, **queue**, and any command that involves printing.

  There are two **stats –T** commands, one for the RA 6100 and another for the RA 6300. The RA 6100 command described in the *Remote Annex Administrator's Guide for UNIX* does not apply to the RA 6300. For information on the **stats –T** command that *does* apply to the RA 6300, see *Displaying RA 6300 Statistics* on page B-41.

- The warning that the **boot** command can send is seen only by CLI users on modem connections. PPP, SLIP, ARAP, V.120 and sync users do not see it.
- The following statement is meaningless:

  The R6.x ROM Monitor **compact** command is incompatible with R7.0 and above. Once the Annex boots the current operational image, use only the CLI superuser **compact** command to compress non-volatile memory.
- Wherever you would specify a port or set of ports for a non-RA 6300 Annex, you instead specify one of the following for the RA 6300:

  n or **asy***n* (where *n* is an integer specifying the number of an internal asynchronous port.)

  ctl*n* (where *n* is an integer specifying the number of an internal control port). Internal control port #1 (ctl1) is the RA 6300's console port and is the only one available. Other control ports are reserved for future use.

syn*n* (where *n* is an integer specifying the number of an internal synchronous port.)

ta*n* (where *n* is an integer specifying the number of an internal TA (V.120) port.

- The **primgr** process applies only to the RA 6300. This process handles ISDN signaling.
- Ignore the following statement in the descriptions of the **cp**, **more**, **mv**, and **rm** commands:

  Only ROM revisions 0600 and greater with the self boot option installed support this command.
- The **modem** command description is incorrect for the RA 6300 when it states that the Annex supports only modem types defined in the Annex configuration file. RA 6300 modem types are hard-coded.

  Also, the RA 6300 supports two additional arguments to the **modem** command: **–m** and **–u**. For more information, see *Modems* on page A-81.
- The **pri** superuser CLI command is available for use with the RA 6300. This command displays PRI information. Issued with the **b** argument, the **pri** command displays information about the PRI B channel assignments. For a complete description of **pri** and **pri b**, see *Network Administration* on page B-1.
- The superuser CLI **sessions** command is available for use with the RA 6300. This command displays information about active Session Parameter Blocks (SPBs). For a description, see *Displaying Active Session Parameter Blocks* on page B-7.
- The **stty isize** parameter does not apply to the RA 6300 (even though the parameter is still available).

- For the RA 6300, the default value for the **baud** parameter set by the **stty** command is **115200**, not **9600**. Also, the default value for the **oflow** argument is **eia** for the RA 6300.
- The **tap** command has a new argument, **–f**. Specifying this argument forces a tap to occur and allows you to tap across several calls on one internal port.
- When configuring **tn3270** to be run from the RA 6300, you will probably want to set the **term_var** parameter within an SPB, not via **na** or **admin**, since it is unlikely that all the terminals you use will have exactly the same characteristics.
- The output from the **who** command has changed slightly (see *Network Administration* on page B-1).

This chapter in the *Remote Annex Administrator's Guide for UNIX* describes the following utilities:

- **aprint**.
- **erpcd**.
- **rtelnet**.

For the Remote Annex 6300, please note that:

- You should not direct the **aprint** utility to an RA 6300.
- The **rtelnet** utility is not applicable to the RA 6300.
- New arguments are available with **erpcd**; a revised description follows.

erpcd

The **erpcd**, or expedited remote procedure call daemon, responds to all Annex boot, dump, and ACP security requests. This daemon contains two programs:

- **bfs** – the block file server used to access host files and dump Annex images.
- **acp** – the Access Control Protocol program for host-based security requests.

Table C-36 lists the arguments for **erpcd**. The syntax is:

/etc/erpcd [[**–D**[*level*]] [**–c** [*maxnumber*] [**–d** [*udpport*][**–f** [*directory*]\
[**–p**][**–s** [*directory*] [**–u** [*filename*] [**–b** [*max_con*] [**–x** [*max_total*]\
[**–g** [*period*]] [– [**l**, **L**]] [– [**a**, **A**]] [**–n**] [– [**t**, **T**]] [**–v** [version]]

When operational code is downloaded to Annexes, a minimum of one host, accessible to an Annex, must be running **erpcd** with the **bfs** program enabled. A UDP port (121) for **erpcd** must be defined in the services database and the **eservices** file must be configured properly.

For expert *C* coders only: the host **erpcd** daemon can implement macros, extended commands, command logging, and arbitrary security restrictions through a set of interface routines to the Annex CLI. These are documented in the **acp_policy.doc** file; the **acp_policy.c** file contains examples (for more details on implementing code changes, see *Modifying the Supplied Security Application* on page A-325 and *Modifying the Code* on page A-335).

Table C-36. Supported Arguments for erpcd

Argument	Description
–D*level*	Restarts **erpcd** in test mode on the load server host – it does not detach from the *tty* and it prints out extensive debugging information. Entering a debugging *level* increases the amount of debugging information. For each *level*, specify a **D**. For example **–DD** specifies 2 levels. Note that there is no spece between the **D**s.
–c *maxnumber*	Specifies the maximum number of child processes that **erpcd** can create (for handling simultaneous requests).
–d *udpport*	Specifies a UDP port number; the default is 121.

(continued on next page)

Table C-36. Supported Arguments for erpcd (continued)

Argument	Description
–f *directory*	Specifies the location of the **bfs** files (load/dump); defaults to the defined **bfs** directory (usually **/usr/spool/erpcd/bfs**).
–p	Prints the daemon's process ID to standard output so that automatic mechanisms can start/stop this process.
–s *directory*	Specifies the location of the security files (**acp_*)** and **eservices**); defaults to the defined install-annex directory (usually **/etc**).
–u *filename*	Invokes the **acp_userinfo** file syntax checker on the file designated by *filename*. If *filename* is omitted, **stdin** is used. Running this option does not interfere with any other **erpcd** running on the system. Parsing errors are printed on **stderr**. Error messages are in the form: <*filename*>: line <*number*>: <*severity*>: <*description*> where <*filename*> is the name of the file, <*number*> designates the line on which the error occurs, <*severity*> is either an *error* or a *warning* (*error* indicates there is a serious parsing error; *warning* indicates the parser remedied the situation by conversion), and <*description*> describes the error.
–b *max_con*	The number of consecutive login failures a user is permitted before being blacklisted. Valid values are 0-8. A value of 0 enables blacklisting upon any login failure (not recommended). This value can also be set via the MAX_BL_CON variable in **acp_policy.h**. The default, as pre-set via MAX_BL_CON, is 5. If MAX_BL_CON is undefined and you do not specify **–b max_con**, ACP never blacklists based on consecutive login failures. For more details, see *Configuring Blacklisting* on page A-286.

(continued on next page)

Table C-36. Supported Arguments for erpcd (continued)

Argument	Description
–x *max_total*	The number of non-consecutive login failures a user is permitted before being blacklisted. Valid values are 0-20. A value of 0 enables blacklisting upon any login failure (not recommended). This value can also be set via the MAX_BL_NONCON variable in **acp_policy.h**. The default, as pre-set by MAX_BL_NONCON, is 10. If MAX_BL_NONCON is undefined and you do not specify **–x max_total**, ACP never blacklists based on consecutive login failures. For more information, see *Configuring Blacklisting* on page A-286.
–g *period*	The time period, in weeks, over which *max_total* is applied. Login failures that occurred more than this number of weeks ago do not count toward blacklisting. Valid values are 0-52. his value can also be set via the MAX_BL_NONCON variable in **acp_policy.h**. The default, as pre-set via MAX_BL_PERIOD, is 26. If MAX_BL_PERIOD is undefined or is set to 0, MAX_BL_NONCON is effectively disabled. For more information, see *Configuring Blacklisting* on page A-286.
–lL	ACP logfile information directed to syslog. **-l** turns it off; **-L** turns it on.
–aA	ACP logfile. **-a** turns it off; **-A** turns it on.
–n	Use host name instead of IP address in logfile.
–tT	ACP log information stamped with date/second. **-t** use seconds;**-T** use standard time format.
–v	Displays software version number.