Chapter 1

Operation

Introduction

This section describes the functions and commands available on the router to support day-to-day operational and network management activities.

The commands described in this section fall into the following functional groups:

- The command processor and router configuration.
- The User Authentication Facility.
- Monitoring and fault diagnosis of the router and the network.
- Managing flash memory and the Flash File System (FFS).
- Downloading software releases and enhancements.

The Command Processor

You can set up, manage, monitor, and troubleshoot the router using one of these *user interfaces*:

- the CLI (command line interface), which provides an extensive set of commands. You can enter commands from:
  - a terminal, or a terminal emulator on a PC, connected to one of the asynchronous ports
  - a telnet session, for remote access
  - a secure shell session, for secure remote access
- the web-based GUI (graphical user interface), which includes the commonly-required functions for a number of protocols. You can access the GUI using
  - HTTP, for local or remote access
  - HTTPS, for secure remote access

This section describes the basic functionality of the CLI.

Logging In

When the router is first shipped, it has a *manager* account that includes an initial login name and password. If you are just starting to use the router:

Enter the following name at the login prompt:

```
login: manager
```

Enter the following password at the password prompt:

```
password: friend
```

Users with Manager privilege can set up other accounts with the **add user** command on page 1-65

All users must enter a login name and password to access the command prompt. Users access the router from a terminal or PC connected to the RS-232 terminal port (asyn0), or through a Telnet or HTTP connection.

Before users get the prompt that lets them log in, contents from a file named *login.txt* is displayed if it exists in flash memory. The login.txt file lets various kinds of messages be sent to users. The following diagram is an example of output from the login.txt file.

```
INFO: Self tests beginning.
INFO: RAM test beginning.
PASS: RAM test, 65536k bytes found.
INFO: BBR tests beginning.
...........
...........
INFO: Switch startup complete

Warning: This equipment is for authorised persons only. If you
do not have proper clearance, please logout now.

Login:
```

Users with Manager level privilege or higher create the file named login.txt by using the **edit** command or by loading an existing text file. The contents of the file must be in printable ASCII characters but with no control characters. When no login.txt file exists, the login prompt is displayed without a message.

For more information to help create a login.txt file, see the **edit** command on page 1-85 and the **load** command on page 1-96.

After someone with User level privilege successfully logs in, the router activates an auto-executing file, autoexec.scp, if one is in flash memory. Users with Manager level privilege or higher also create these script files. For more information about scripts, see Chapter 34, Scripting.

Normal Mode and Security Mode

The commands that a user is allowed to execute depend on the mode in which the router is operating and the user's privilege level. The router operates in two modes: normal (default) and security.

Security mode provides additional protection to routers with encryption hardware or configured to provide sensitive security functions such as IP authentication, IPsec (Chapter 45, IP Security (IPsec)), Secure Shell (see Chapter 43, Secure Shell) and encryption (Chapter 25, Compression and Encryption Services). Security mode is enabled by using the command:

```
enable system security_mode
```

This command also creates a security mode enabler file in the router's file subsystem. This file cannot be manually modified, displayed, deleted, copied, or renamed. If the router is restarted, the startup process checks for the presence of the enabler file. If the enabler file is present, the router boots up in security mode; otherwise, the it boots up in normal mode. The router is restored to normal operating mode by using the command:

```
disable system security_mode
```

This command also deletes the security mode enabler file in the router’s file subsystem. Sensitive data files such as encryption keys can be stored in the file subsystem when the router is operating in security mode.

Disabling security mode deletes sensitive data files, such as encryption keys, from the router's file subsystem.

To display the current operating mode, use the command:

```
show system
```

When the router operates in security mode, only users with Security Officer privilege (see “Privilege Levels” on page 1-9) can execute commands that could affect the security of the router and its keys (Table 1-1 on page 1-7).

Table 1-1: Commands requiring Security Officer privilege when the router is in security mode

Command	Specific Parameters
ACTIVATE IPSEC	
ACTIVATE SCR	
ADD FR DLC (PIC bay only)	ENCRYPTION
ADD IP INT	
ADD IP SA	
ADD PKI	
ADD SA	
ADD SCR	
ADD SSH	
ADD USER	
CREATE CONFIG	
CREATE ENCO KEY	
CREATE FR (PIC bay only)	DEFENCRYPTION
CREATE IPSEC	
CREATE ISAKMP	
CREATE PKI	
CREATE PPP	
CREATE PPP TEMPLATE	
CREATE SA	
CREATE SNMP COMMUNITY	
CREATE STAR	
DEACTIVATE SCR	
DELETE FILE	
DELETE IP SA	
DELETE PKI	
DELETE SA	
DELETE SCR	
DELETE SSH	

Table 1-1: Commands requiring Security Officer privilege when the router is in security mode (continued)

Command	**Specific Parameters**
DELETE USER	
DESTROY ENCO KEY	
DESTROY IPSEC	
DESTROY ISAKMP	
DESTROY PKI	
DESTROY SA	
DESTROY STAR	
DISABLE FEATURE	
DISABLE IPSEC	
DISABLE ISAKMP	
DISABLE PKI DEBUG	
DISABLE SA	
DISABLE SSH	
DISABLE USER	
DUMP	
EDIT	
ENABLE FEATURE	
ENABLE IPSEC	
ENABLE ISAKMP	
ENABLE PKI DEBUG	
ENABLE PPP DEBUG	
ENABLE PPP TEMPLATE DEBUG	
ENABLE SA	
ENABLE SNMP	
ENABLE SSH	
ENABLE STAR	MKTTRANSFER
ENABLE USER	
LOAD	
MAIL	
MODIFY	
PURGE IPSEC	
PURGE PKI	
PURGE USER	
RENAME FILE	
RESET ENCO	
RESET IPSEC	
RESET USER	
SET CONFIG	
SET ENCO KEY	
SET FR (PIC bay only)	ENCRYPTION, DEFENCRYPTION

Table 1-1: Commands requiring Security Officer privilege when the router is in security mode (continued)

Command	Specific Parameters
SET INSTALL	
SET IP INT	
SET IPSEC	
SET PKI	
SET PPP	
SET PPP TEMPLATE	
SET SA	
SET SCR	
SET SNMP COMMUNITY	
SET SSH	
SET STAR	
SET USER	
SHOW CONFIG	
SHOW ENCO KEY	
SHOW FEATURE	
SHOW FILE	
SHOW PPP	CONFIG
SHOW STAR	[=id], MKTTRANSFER, NETKEY
UPLOAD	

Privilege Levels

The commands that a user is allowed to execute depend on the person’s privilege level as well as the mode in which the router is operating. The router supports three levels of privilege for users:

- User (lowest)
- Manager
- Security Officer (highest)

User and Manager privileges can be distinguished by the different prompts the command processor displays when it is ready to receive commands. A User level prompt is a single angle bracket (>), whereas a Manager prompt looks like the following:

```
Manager >
```

User level

The commands that can be executed by a user depend on the user’s privilege level and whether the router is operating in normal or security mode:

The User level has access to a very limited subset of commands, regardless of whether the router is operating in normal or security mode. User level commands affect the user’s own session or asynchronous port. User privilege applies to a user who has not logged in—someone using a terminal connected to an asynchronous port that is **not** in secure mode—or a user who has logged in with a username with User privilege.

Manager level

The Manager level has access to the set of commands for configuring and viewing all aspects of the router that are not security critical. When the router operates in security mode, users with Manager privilege cannot execute a subset of the commands known as the security commands. Manager privilege can be gained in one of the following ways:

- Using the following command from a port or Telnet session to login under a name that has Manager privilege:

  ```
  login
  ```

  The command prompts for a login name and password. The password is case-sensitive and must be entered exactly as defined. If the password is entered correctly, the port or Telnet connection gains Manager privilege and the prompt changes to the Manager level prompt. This is the usual method of gaining Manager privilege, especially when managing remote routers.

- Using the following command to set a port as a semipermanent Manager port:

  ```
  set manager asyn
  ```

  Any terminal connected to the specified port has Manager privilege. The **set manager asyn** command on page 1-119 is a Manager level command and can only be entered from a port or a Telnet session that already has Manager privilege. Only one port at a time can be defined as manager port.

To return to User level, use the command:

```
logoff
```

Normally, the prompt changes when the user's privilege level changes from User to Manager or vice versa. The prompt does not change when commands are entered from a terminal connected to a physical port and the port's **prompt** parameter has been changed to a user-defined string with the **set asyn** command on page 7-37 of Chapter 7, Interfaces.

Security Officer level

The Security Officer level has access to the full set of commands regardless of whether the router is operating in normal mode or security mode. When the router is in security mode, only users with Security Officer privilege can execute security commands (see Table 1-1 on page 1-7). When the router is operating in normal mode, Manager privilege is equivalent to Security Officer privilege. A user must login under a login name that has Security Officer privilege from a terminal directly connected to an asynchronous port on the router or a Telnet session originating from an authorised IP address (see "Remote Security Officer" on page 1-11).

A security timer operates while a user is logged in with Security Officer privilege to minimise the risk of unauthorised access to an un-attended terminal or Telnet session. Every time a command is entered, the security timer is restarted. If the timer expires, the user's privilege is reset to Manager level, but the user remains logged in. Any attempt to execute a security command requires the user to re-enter the Security Officer password. Configure the timeout period in seconds by using the command:

```
set user securedelay=10..600
```

Remote Security Officer

The *Remote Security Officer* (RSO) feature lets a remote user connect to a router via Telnet from an authorised IP address, and login using a name with Security Officer privilege as if the user were at a terminal connected directly to the router. By default the Remote Security Officer feature is disabled.

The RSO feature can be enabled or disabled with the commands:

```
enable user rso

disable user rso
```

Authorised IP addresses can be added and deleted with the commands:

```
add user rso ip=ipadd [mask=ipadd]

add user rso ip=ipadd[-ipadd]

add user rso ip=ipv6add[/prefix-length]

add user rso ip=ipv6add[-ipv6add]

delete user rso ip=ipadd[-ipadd]

delete user rso ip=ipv6add/prefix-length

delete user rso ip=ipv6add[-ipv6add]
```

The current state of the RSO feature and the list of authorised IP addresses can be displayed by using the command:

```
show user rso
```

All RSO commands require Security Officer privilege and therefore must be executed from a terminal directly attached to the router or from a Telnet session originating from a previously configured RSO address. RSO must be enabled, and the first address added, from a terminal directly attached to the router. If RSO is disabled (either from a terminal or a Telnet session) it must be re-enabled from a terminal directly attached to the router.

Once RSO has been enabled and configured with one or more IP addresses, a Telnet session from one of the authorised addresses can login as a user with Security Officer privilege.

Entering Commands

The router supports command line editing and recall. The functions available are shown in the following table.

Table 1-2: Command line editing functions and keystrokes

Function	**VT100 Terminal**	**Dumb terminal**
Moves cursor backwards and forwards within the command line by using the cursor keys	←, □→	*Not available*
Moves the cursor to the beginning of the command line	[Ctrl/A]	*Not available*
Moves the cursor to the end of the command line	[Ctrl/E]	*Not available*

Table 1-2: Command line editing functions and keystrokes (continued)

Function	VT100 Terminal	Dumb terminal
Deletes character to left of cursor	[Delete] or [Backspace]	[Delete] or [Backspace]
Toggles between insert/overstrike	[Ctrl/O]	*Not available*
Clears command line	[Ctrl/U]	[Ctrl/U]
Moves backwards through a history of previous commands	↑ or [Ctrl/B]	[Ctrl/B]
Moves forwards through a history of previous commands	↓ or [Ctrl/F]	[Ctrl/F]
Displays command history and select a command from the list	[Ctrl/C] or **show asyn history**	[Ctrl/C] or **show asyn history**
Clears command history	**reset asyn history**	**reset asyn history**
Recalls the most recent command matching a partially entered command	[Tab] or [Ctrl/I]	[Tab] or [Ctrl/I]
Terminates Telnet session before login complete	[Ctrl/D]	[Ctrl/D]
Interrupts (or "breaks") text paging or continuously streaming text, for example, when results from **show** commands are displayed. Buffered text is deleted.	Ctrl-Q	Ctrl-Q

Commands are limited to 1000 characters, excluding the prompt. Pathnames may be up to 256 characters, including file names, and file names up to 28 characters long with extensions of 3 characters, are supported.

The router assumes that the width of the terminal screen is 80 characters, and performs command line wrapping at the 80th column regardless of the setting of the terminal. The cursor does not need to be at the end of the line for the command to be executed. The default editing mode is insert mode. Characters are inserted at the cursor position and any characters to the right of the cursor are pushed to the right to make room. In overstrike mode, characters are inserted at the cursor position and replace any existing characters.

Aliases

The command line interface supports aliases. An *alias* is a short name for an often-used longer character sequence. When the user presses [Enter] to execute the command line, the command processor first checks the command line for aliases and substitutes the replacement text. The command line is then parsed and processed normally. Alias substitution is not recursive—the command line is scanned only once for aliases.

Aliases are created and destroyed by using the commands:

```
add alias=name string=substitution
```

```
delete alias=name
```

A list of all the aliases defined on the router and their replacement strings can be displayed by using the command:

```
show alias
```

Online Help

Online help is available for all router commands by using the command:

```
help [topic]
```

If you do not specify a topic, then a list of available ones is displayed.

The system help file that the help information comes from can be stored in flash memory. If you upgrade your software release, you can also upload the associated new help file, then activate it by using the command:

```
set help=helpfile
```

To display the current help file, enter the command:

```
show system
```

Also, typing a question mark character at the end of a partially completed command displays a list of the parameters that may follow the current command line, with the minimum abbreviations in uppercase letters (Figure 1-1 on page 1-13). The current command line is then re-displayed, ready for further input.

Figure 1-1: Using the question mark character to display help for the current command on an AR410.

```
Manager > ADD ?

  Options : ACC APPletalk BGP CLASSifier BOOTp BRIDge DECnet FRamerelay GRE IP IPX
     ISDN LAPD LOG MIOX NTP OSPF PERM PPP RADius SA SCript SNmp STReam STT TRIGger
    TACacs USEr X25C X25T TDM

Manager > ADD ACC ?

  Options : CALL SCript DOmainname

Manager > ADD ACC CALL ?

  Options : DIrection DScript CScript RScript POrt ENcapsulation AUthentication
    DOmainname
```

Storing and Retrieving Configuration Information

At boot, the router executes commands in the boot script to configure the router. The default boot script is called `boot.cfg` but an alternative script file can be defined as the boot script by using the command:

```
set config={filename|none}
```

Subsequent commands entered at the command line or executed from a script affect only the dynamic configuration in memory, which is not retained over a power cycle. Changes are not automatically stored in non-volatile memory. When the router is restarted, the configuration is restored to the one defined by the boot script. If the router is restarted with the **restart** command on page 1-109, the script specified in the **restart** command is used.

To ensure that configuration changes made after boot are retained across a restart or power cycle, the modified configuration must be saved as a script file by using the command:

```
create config=filename
```

The **create config** command on page 1-70 writes the MD5 digest, not the plaintext, of passwords in commands to the configuration file. When a configuration script is executed, the command processor can determine whether the password value is plaintext or an MD5 digest.

If the file name specified is boot.cfg, or the file is set as the boot script with the **set config** command on page 1-110, the modified configuration is automatically restored after a restart or power cycle. If another name is specified, the configuration can be restored after a restart or power cycle by using the command:

```
activate script=filename
```

The Graphical User Interface (GUI)

You can configure and manage the router using its web-based Graphical User Interface (GUI), by browsing to any interface's IP address. The GUI offers an alternative to the CLI for many tasks, and is designed to make complicated tasks simpler and regularly performed tasks quicker.

The router's User Guide describes the GUI in detail, including

- information about supported operating systems and browsers
- detailed step-by-step instructions for accessing the router via the GUI
- information about the GUI's structure and navigation
- step-by-step instructions for upgrading to a new GUI (for example, when you upgrade to a new software release)
- troubleshooting tips

Enabling and Disabling the GUI

The GUI is enabled by default. To enable or disable the GUI, use the following commands:

```
enable gui
```

```
disable gui
```

When enabled, the GUI works when a valid resource file for the hardware model is present in flash memory and when the HTTP server is enabled (see "HTTP Client and Server" on page 1-45).

Displaying Information about the GUI

The GUI resource file has an 8-digit name with a .rsc file extension (for example, d450se01.rsc). To list files on the router, use the command:

```
show file
```

To display information about the current GUI resource file, use the command:

```
show gui
```

To display information about the router’s HTTP server, use the commands:

```
show http server
show http server SESSION
```

To check which GUI resource file is installed on the router, use the command:

```
show install
```

User Authentication Facility

The User Authentication Facility (UAF) controls access to the router’s command prompt, asynchronous services, and dialup services through a login name and password. A user is prompted to enter a login name and password when:

- The user attempts to access the router’s command prompt via a terminal connected directly to an asynchronous port set to Secure mode.
- The user attempts to access the router’s command prompt via a Telnet connection.
- The user attempts to access a dialup service via an asynchronous modem connected to an asynchronous port.
- The user enters the **login** command on page 1-101.

The UAF prompts the user for a login name and password (Figure 1-2). The user must enter appropriate responses, pressing [Return] after each response. Characters entered at the password prompt are not echoed to the screen for security reasons.

Figure 1-2: A typical login session for user BRUCE on router CMD

```
CMD login: bruce
password:



CMD >
```

If the user enters an invalid login name or password, the sequence is repeated a set number of times. If a valid login name and password are still not entered, the terminal or Telnet session is locked out for a period of time. The password prompt is withheld during this period, preventing the user from logging in or entering commands. The manager can specify the number of allowable login attempts and the length of the lockout period.

The password prompt is displayed regardless of whether a password is required for the login name entered by the user. This makes it more difficult for an intruder to discover valid login name/password combinations.

Users authenticated by the UAF can be operators or other routers. If the user is another router, the authentication occurs without appearing in a terminal screen.

The UAF supports the following methods of user authentication: an internal database called the *User Authentication Database*, and interrogation of external

RADIUS (Remote Authentication Dial In User Service), TACACS (Terminal Access Controller Access System) or TACACS+ servers.

The UAF first queries any TACACS+ servers that have been defined. If there are no defined TACACS+ servers or all the TACACS+ servers return a *reject* response, the UAF queries the User Authentication Database. If the supplied login name and password do not match an entry in the User Authentication Database, the UAF sends authentication requests to any RADIUS servers that have been defined. If there are no defined RADIUS servers or all the RADIUS servers return a *reject* response, the UAF sends authentication requests to any TACACS servers that have been defined. If the supplied login name and password match an entry in the User Authentication Database, or one of the defined TACACS+, RADIUS or TACACS servers returns an *accept* response to an authentication request, the login is accepted. If the supplied login name and password does not match an entry in the User Authentication Database, and all of the defined TACACS+, RADIUS or TACACS servers return *reject* responses to authentication requests, the login is rejected. The login is also rejected if Login is set to **no**.

The User Authentication Database

The User Authentication Database stores information about the users who are permitted to have access to the router's command prompt, asynchronous services, and dialup services. Users are identified by a login name. Each login name has an associated record in the database that specifies:

- The password that the user must enter to log into the router.
- The privilege level for the user: User, Manager, or Security Officer.
- Whether the user is permitted to use the **telnet** command on page 21-31 of Chapter 21, Terminal Server, or to connect to a Telnet service from a Telnet session.
- The IP address, network mask, and MTU (Maximum Transmission Unit) to use for PPP or SLIP connections to the router via an asynchronous port.
- A callback number for use with the PPP callback facility.
- Whether the user is permitted to log into the router and enter commands.

Adding Entries to the User Authentication Database

When the router is started up for the first time one account is created automatically. This account has the login name Manager, the password "friend", login = yes, and Manager privilege. This account cannot be deleted although the password may be changed.

The manager should change the password of the Manager account at the earliest opportunity. Leaving the Manager account with the default password is a security risk, as the account name and default password are well documented.

To add more users to the User Authentication Database, use the command:

```
add user=login-name password=password [callingnumber=number]
    [cbnumber=e164number] [description=description]
    [privilege={user|manager|securityofficer}] [telnet={yes|
    no}] [ipaddress=ipadd] [ipxnetwork=network]
    [netmask=ipadd] [mtu=40..1500] {login=true|false|on|off|
    yes|no}
```

The number of entries in the database is limited only by the amount of memory available. Only the login name and password are required. The default privilege level is User. Other information about a user that may be specified includes a description for the entry (such as the user's full name), the privilege level, whether the user is permitted to use the **telnet** command on page 21-31 of Chapter 21, Terminal Server or connect to a Telnet service, an IP number, network mask and MTU (Maximum Transmission Unit). The IP number, network mask and MTU are required if the user is to run asynchronous PPP or SLIP over an asynchronous modem connected to an asynchronous port. The callback number is required if the user is to make a PPP callback request with user authentication. See Chapter 9, Point-to-Point Protocol (PPP) for more information. The calling number is used for L2TP and ISDN services that provide caller ID information.

Modifying Entries in the User Authentication Database

To modify an entry in the database, use the command:

```
set user=login-name [password=password]
    [callingnumber=number] [cbnumber=e164number]
    [description=description] [privilege={user|manager|
    securityofficer}] [telnet={yes|no}] [ipaddress=ipadd]
    [ipxnetwork=network] [netmask=ipadd] [mtu=40..1500]
    login={true|false|on|off|yes|no]
```

An entry in the database can be deleted by using the command:

```
delete user=login-name
```

All entries in the database, except the Manager account, can be deleted by using the command:

```
purge user
```

The contents of the database can be displayed by using the command:

```
show user[=login-name]
```

Passwords

All users, including managers, should take care in selecting passwords. Tools exist that enable hackers to guess or test many combinations of login names and passwords easily. The UAF provides some protection against such attacks by allowing the manager to set the number of consecutive login failures allowed and a lockout period when the limit is exceeded.

However, the best protection against password discovery is to select a good password and keep it secret. When choosing a password:

- Make it six or more characters long. The UAF enforces a minimum password length, which can be changed by the manager. The default is six characters.
- Include both alphabetic (a–z) and numeric (0–9) characters.
- Include both uppercase and lowercase characters. The passwords stored by the router are case-sensitive, so “bgz4kal” and “Bgz4Kal” are different.
- Avoid words in a dictionary unless combined with other random alphabetic and numeric characters.
- **Do not** use the login name, or the word “password” as the password.
- **Do not** use your name, your mother's name, your spouse's name, your pet's name, or the name of your favourite cologne, actor, food, or song.

- **Do not** use your birth date, street number, or telephone number.
- **Do not** write down your password anywhere.

A manager can alter the password for any user by using the command:

```
set user=username password=password
```

This may be necessary if a user forgets the password. A log message is generated whenever the password for a manager account is changed.

A user who is logged in can change their own password by using the command:

```
set password
```

The command prompts for the old password, the new password, and confirmation of the new password. The new password and the confirmation must be identical for the change to take affect. This reduces the chance of a typing error causing the password to be different from what the user intended.

Database Security

A manager session left unattended is a severe security risk because the User Authentication Database can be modified from a manager session. To reduce the risk of unauthorised activity, a subset of manager commands (Table 1-3 on page 1-18), called *security commands*, have a *security timer*. Every time a security command is entered from a manager session, a security timer starts. If a security command is entered after the timer has expired, the manager is prompted to re-enter the password correctly before the command is actioned. If the password is not entered correctly, the password prompt is repeated a set number of times. If the correct password is still not entered, a log message is generated and the session is logged off.

The security timer lets a manager make successive additions and modifications to the database at one time without having to re-enter the password for every command.

The security timer does not provide a foolproof security mechanism. Managers should log out of a manager session before leaving a terminal unattended.

Table 1-3: Secure commands controlled by the security timer

Command	Description
add tacacs server	Adds a TACACS server to the list of TACACS servers used for user authentication.
add user	Adds a user to the User Authentication Database.
delete tacacs server	Deletes a TACACS server from the list of TACACS servers used for user authentication.
delete user	Deletes a user from the User Authentication Database.
purge user	Deletes all users except Manager from the User Authentication Database.
set manager asyn	Assigns a port semipermanent Manager privilege.
set user	Modifies a user record in the User Authentication Database.

If the router is operating in security mode, the manager must also be logged in to a user account with Security Officer privilege in order to execute commands in the previous table.

Logging In and Logging Out

A user is automatically prompted to enter a login name and password when accessing the router via Telnet or a terminal connected to an asynchronous port set to Secure mode, or when accessing a dialup service via an asynchronous modem connected to an asynchronous port.

There are other occasions when a user may wish to login manually. A user on a terminal connected to an asynchronous port that is not in secure mode may want to login to use facilities that are available only to logged in users, such as the **telnet** command on page 21-31 of Chapter 21, Terminal Server. A user who is already logged in may want to temporarily log in as another user in acquire different rights, such as Manager privilege.

To manually log into a router, use one of these synonymous commands:

```
login
logon
logi
```

To log out of a session, use one of these synonymous commands:

```
logoff
logout
lo
```

If a user starts a Telnet session with a router but does not log in within one minute, the router automatically times out and terminates the Telnet connection.

Recovering Lost Passwords

If a user forgets their password, the password can be reset from an account with Manager privilege by using the command:

```
set user=login-name password=password
```

Passwords for accounts with Manager privilege can be reset with the same command, provided the manager can login to at least one account with Manager privilege. Passwords for accounts with Security Officer privilege can be reset from any account with Security Officer privilege.

If passwords for all accounts with Manager or Security Officer privilege are lost, recovery is complex. Contact your authorised Allied Telesyn distributor or reseller for assistance.

Asynchronous Port Security

To set asynchronous ports to secure mode, use the command:

```
set asyn secure=on
```

See Chapter 7, Interfaces for a detailed description of the **set asyn** command on page 7-37 of Chapter 7, Interfaces. By default, all asynchronous ports are set to secure mode. Telnet sessions are always in secure mode. A user accessing the router via a terminal connected to an asynchronous port in secure mode, or via

Telnet, must login before the router accepts any other commands. When a user Telnets to a router the login and password prompts are always displayed. The password prompt is displayed even when the login name does not match an entry in the User Authentication Database. This makes it more difficult to discover a valid login name. When a login name and password are entered that do not match an entry in the database, and is not accepted by any defined TACACS servers, the login sequence is repeated. If successive login failures occur, the login prompt is withheld for a specified lockout period. This makes it very difficult for an intruder to gain entry with random login names and passwords. A log message is generated when the number of retries for a connection is exceeded and the lockout period is instigated. Telnet logins from an offending IP address are also locked out for this period once the permitted number of failures is exceeded. The number of login attempts permitted and the length of the lockout period can be configured with the command:

```
set user [loginfail=1..10] [lockoutpd=0..30000]
```

Telneting from the Router

The router provides three modes of access to host services:

- Using the **connect** command on page 21-15 of Chapter 21, Terminal Server to access asynchronous services. These are typically hosts connected directly to asynchronous ports on the router and defined as services using the **set service** command on page 21-21 of Chapter 21, Terminal Server.
- Using the **connect** command on page 21-15 of Chapter 21, Terminal Server to access Telnet services. These are typically Telnet hosts defined as services using the **set service** command on page 21-21 of Chapter 21, Terminal Server.
- Using the **telnet** command on page 21-31 of Chapter 21, Terminal Server to access Telnet hosts.

When the user is authenticated using TACACS+, they can only telnet from the switch if their TACACS+ privilege level is equal to or higher than the minimum TACACS+ privilege level required for using telnet on the router. By default, no TACACS+ users can use telnet on the router. See “TACACS+” on page 1-25 for more information about TACACS+. See “TACACS+ and Telneting from the Router” on page 1-29 for more information about how to allow TACACS+ authorised users to telnet from the router.

If the user is authenticated from the user database, each entry in the database has a **telnet** attribute that determines access modes that the user is permitted to use.

All users can use the **connect** command on page 21-15 of Chapter 21, Terminal Server to access asynchronous services, although users accessing the router via Telnet or a terminal attached to an asynchronous port in secure mode must login first to gain access to the command prompt.

Users logged into the router via a terminal attached to an asynchronous port can also use the **connect** command on page 21-15 of Chapter 21, Terminal Server to access Telnet services. In addition, if the user is logged into an account with the **telnet** attribute set to **on**, the user can use the **telnet** command on page 21-31 of Chapter 21, Terminal Server to telnet to remote hosts.

Users logged into the router via Telnet can, by default, use the **connect** command on page 21-15 of Chapter 21, Terminal Server to access asynchronous services. If the user is logged in to an account with the **telnet** attribute set to **on**, the user can also use the **connect** command on page 21-15 of Chapter 21, Terminal Server to access Telnet services and the **telnet** command on page 21-31 of Chapter 21, Terminal Server to telnet to remote hosts.

A manager can use the **telnet** attribute to allow users connected to the router via a terminal to access a restricted set of Telnet hosts, by defining those hosts as Telnet services (see the description of the **set service** command on page 21-21 of Chapter 21, Terminal Server) and setting the **telnet** attribute to **off** for selected accounts. Users logged into one of these accounts can use the **connect** command on page 21-15 of Chapter 21, Terminal Server to access the Telnet services but cannot use the **telnet** command on page 21-31 of Chapter 21, Terminal Server to access other Telnet hosts.

Counters

A number of counters record activity associated with the User Authentication Database. Counters relating to specific users in the database can be displayed with the command:

```
show user[=login-name]
```

To display global counters and configuration parameter, use the command:

```
show user configuration
```

All counters are stored in non-volatile storage so that they are retained across router reboots and power cycles.

To reset counters to zero for a specific user, use the command:

```
reset user=login-name
```

To reset counters to zero for all users, the global counters, or all counters, use the command:

```
reset user counter={user|global|all}
```

Semipermanent Manager Port

It is sometimes desirable to have an asynchronous port that has Manager privilege after a router reboot, without a manager having to log on. To set an asynchronous port from default to manager privilege, use the command:

```
set manager asyn=port-number
```

Only one port may be a semipermanent manager port. By default, no semipermanent manager port is defined. This command is one of the security commands (see "Database Security" on page 1-18).

When the router boots with a semipermanent manager port configured, the Manager account is automatically logged in to the port. The port has full Manager privilege except that Telneting from the port is not permitted. The security timer is reset so that the first time a security command is entered the user is challenged for the password for the Manager account.

RADIUS

RADIUS (Remote Authentication Dial In User Service) is a protocol for transferring authentication, configuration, and accounting information between a Network Access Server (e.g. a router) that desires to authenticate its links and a shared RADIUS Server. The RADIUS (authentication) server manages a database of users and provides authentication (verifying user name and password) and configuration information (e.g. IP address, subnet mask, etc.) to the client. The RADIUS (accounting) server stores accounting information about past sessions.

Privilege levels of users can be stored on the RADIUS server and returned with the user authentication so that the user database can be centrally administered from the RADIUS server. The user privilege level affects the commands that a user may execute. When the router is in security mode, it supports three levels of privileges: User, Manager, and Security Officer. Authenticated users can log directly into a device through the Command Line Interface (CLI), or the Graphical User Interface (GUI) with User, Manager, and Security Officer privileges. See "User Authentication Facility" on page 1-15 for more information about user privilege levels.

To enable the RADIUS server to authenticate users and include their privilege level, set up the server so that it returns an appropriate value in the Service-Type attribute. For Security Officer privilege, set the attribute to Administrative (6); for Manager privilege, set it to NAS Prompt (7); and for User privilege, set it to any other value or no value.

Security mode is designed to provide additional protection to routers fitted with encryption hardware or configured to provide sensitive security functions such as IP authentication, IPsec (Chapter 45, IP Security (IPsec)), Secure Shell (Chapter 43, Secure Shell), or encryption (Chapter 25, Compression and Encryption Services). Security mode is enabled using the command:

```
enable system security_mode
```

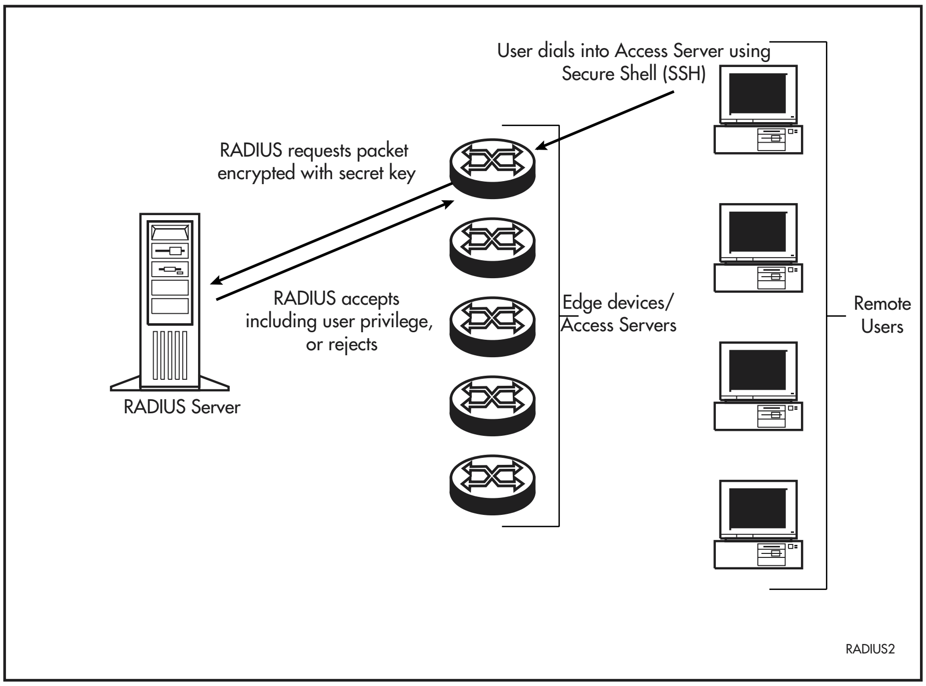
This command also creates a security mode enabler file in the router's file subsystem. This file cannot be manually modified, displayed, deleted, copied, or renamed. If the router is restarted, the startup process checks for the enabler file. If it is present, the router boots up in security mode; otherwise, the router boots up in normal mode. To restore the router to normal operating mode, use the command:

```
disable system security_mode
```

This command also deletes the security mode enabler file in the file subsystem. Sensitive data files such as encryption keys can be stored in the file subsystem only when the router is in security mode.

The router acts as a RADIUS client, sending requests to a RADIUS server.

Figure 1-3: Using a Radius Server for User Authentication.



A RADIUS server is added or deleted by using the commands:

```
add radius server=ipadd secret=secret
delete radius server=ipadd
```

The list of known RADIUS servers is displayed by using the command:

```
show radius
```

Table 1-4 on page 1-23 lists the RADIUS attributes supported by the router.

Table 1-4: RADIUS attributes supported by the router .

RADIUS Attribute Name	When Used	Description
User-Name	Authentication request Accounting request	The name of the user to be authenticated.
User-Password	Authentication request	The password of the user to be authenticated, or the user's input following an Access-Challenge.
CHAP-Password	Authentication request	The response value provided by a PPP CHAP user in response to a challenge.
NAS-IP-Address	Authentication request Accounting request	The identifying IP Address of the NAS that is requesting authentication of the user.
NAS-PORT	Authentication request	The physical port number of the NAS that is authenticating the user.
Service-Type	Authentication accept	Used to specify the privilege level where the user is logged into the router.
Calling-Station-Id	Authentication request	The number that the call to the NAS came from, using Automatic Number Identification (ANI) or similar technology.
Framed-IP-Address	Authentication accept	The address to be configured for the user.
Framed-IP-Netmask	Authentication accept	The IP Netmask to be configured for the user when the user is a router to a network.

Table 1-4: RADIUS attributes supported by the router (continued).

RADIUS Attribute Name	When Used	Description
Callback-Number	Authentication accept	A dialling string to be used for callback.
Framed-Route	Authentication accept	Provides routing information to be configured for the user on the NAS.
Framed-IPX-Network	Authentication accept	The IPX Network number to be configured for the user.
Session-Timeout	Authentication accept	The maximum number of seconds of service to be provided to the user before the session terminates.
Idle-Timeout	Authentication accept	The maximum number of consecutive seconds of idle connection allowed to the user before prompt or termination of the session.
Framed-AppleTalk-Network	Authentication accept	The AppleTalk Network number that the NAS should probe to allocate an AppleTalk node for the user.
Framed-AppleTalk-Zone	Authentication accept	The AppleTalk Default Zone to be used for this user.
CHAP-Challenge	Authentication request	The CHAP Challenge sent by NAS to a PPP CHAP user.
Acct-Status-Type	Authentication start	Whether the Accounting Request marks the beginning (Start) or end (Stop) of the user service.
Acct-Input-Octets	Authentication stop	The number of octets received from the port over the course of this service.
Acct-Output-Octets	Accounting stop	The number of octets sent to the port over the course of this service.
Acct-Session-Id	Accounting start Accounting stop	A unique accounting ID used to match start and stop records in a log file.
Acct-Session-Time	Accounting stop	The number of seconds that the user has received service.
Acct-Authentic	Accounting start	The method by which the user was authenticated.
Acct-Input-Packets	Accounting stop	The number of packets received from the port in the course of delivering this service to a Framed User.
Acct-Output-Packets	Accounting stop	The number of packets sent to the port in the course of delivering this service to a Framed User.
Acct-Terminate-Cause	Accounting stop	The mechanism or reason for terminating the session.

TACACS

The router supports the use of TACACS (Terminal Access Controller Access Control System) servers as an alternative method of user authentication. The router sends a TACACS request, which includes the username and password, to each TACACS server in turn. The TACACS server responds with an *accept* or *reject* response. When the server accepts, the user is authenticated. When the server rejects, it sends a request to the next server in the list until all are queried. When all servers on the list reject the request, user authentication is rejected.

There is a timeout period for TACACS requests; when a response is not received within the specified time, the request is retried. To configure the timeout period and the number of permissible retries, use the command:

```
set user [tacretries=0..10] [tactimeout=1..60]
```

Requests are sent to the TACACS servers on the list in a round-robin fashion until one server accepts it, or all servers reject it, or each server reaches its maximum number of retries.

To add a TACACS server to the list of defined servers, use the command:

```
add tacacs server=ipadd
```

where *ipadd* is the IP address of the TACACS server in dotted decimal notation.

To delete a TACACS server from the list of servers, use the command:

```
delete tacacs server=ipadd
```

To display a list of currently defined TACACS servers, use the command:

```
show tacacs server
```

TACACS+

The TACACS+ protocol is a simple TCP-based access control protocol. It supports authentication and authorisation services, and improves TACACS by:

- separating the functions of authentication, authorisation and accounting
- encrypting all traffic between the Network Access Server (NAS) and the daemon
- using TCP as its transport protocol for reliable delivery
- allowing authentication exchanges of arbitrary length and content, which allow any authentication mechanism to be used with TACACS+ clients
- being extensible to provide for site customisation and future development features.

TACACS+ allows the authentication, authorisation, and accounting services to be provided independently on separate access servers (TACACS+ servers). Each service can be tied into its own database or can use other services available on that server or on the network.

Authentication Services

The TACACS+ protocol forwards many types of username and password information. This information is encrypted over the network with MD5 (Message Digest 5). TACACS+ can forward the password types for ARAP, SLIP, PAP, CHAP, and standard Telnet. This lets clients use the same username and password for different protocols.

TACACS+ authentication supports multiple challenge and response demands from the TACACS+ server. This allows token-card vendors to provide advanced features like sending back a second token-generated number after the first one was manipulated by a security server.

Authorisation Services

Authorisation occurs after authentication. It is here that an *attribute value (AV) pair* is returned when configured. Attribute Value Pairs are configured on the TACACS+ server and passed to the router. The router takes the appropriate action based upon the pair passed to the router and the value of that pair. When the TACACS+ server sends an AV pair that is not supported by the router, that attribute is ignored.

The following AV Pairs are supported:

- Timeout

  This value specifies the length of time that the session can exist. After this value has expired, the session is either disconnected or the privilege of the user is reduced. The valid timeout range is 0 to 65535 (minutes).

- Idletime

  If no input or output traffic is received in this time period, the session is disconnected. The valid idletime range is 0 to 65535 (minutes).

- Privilege Level

  TACACS+ privilege level 0 is not mapped, privilege levels 1-6 are mapped to User, privilege levels 7-14 are mapped to Manager, and privilege level 15 are mapped to Security Officer.

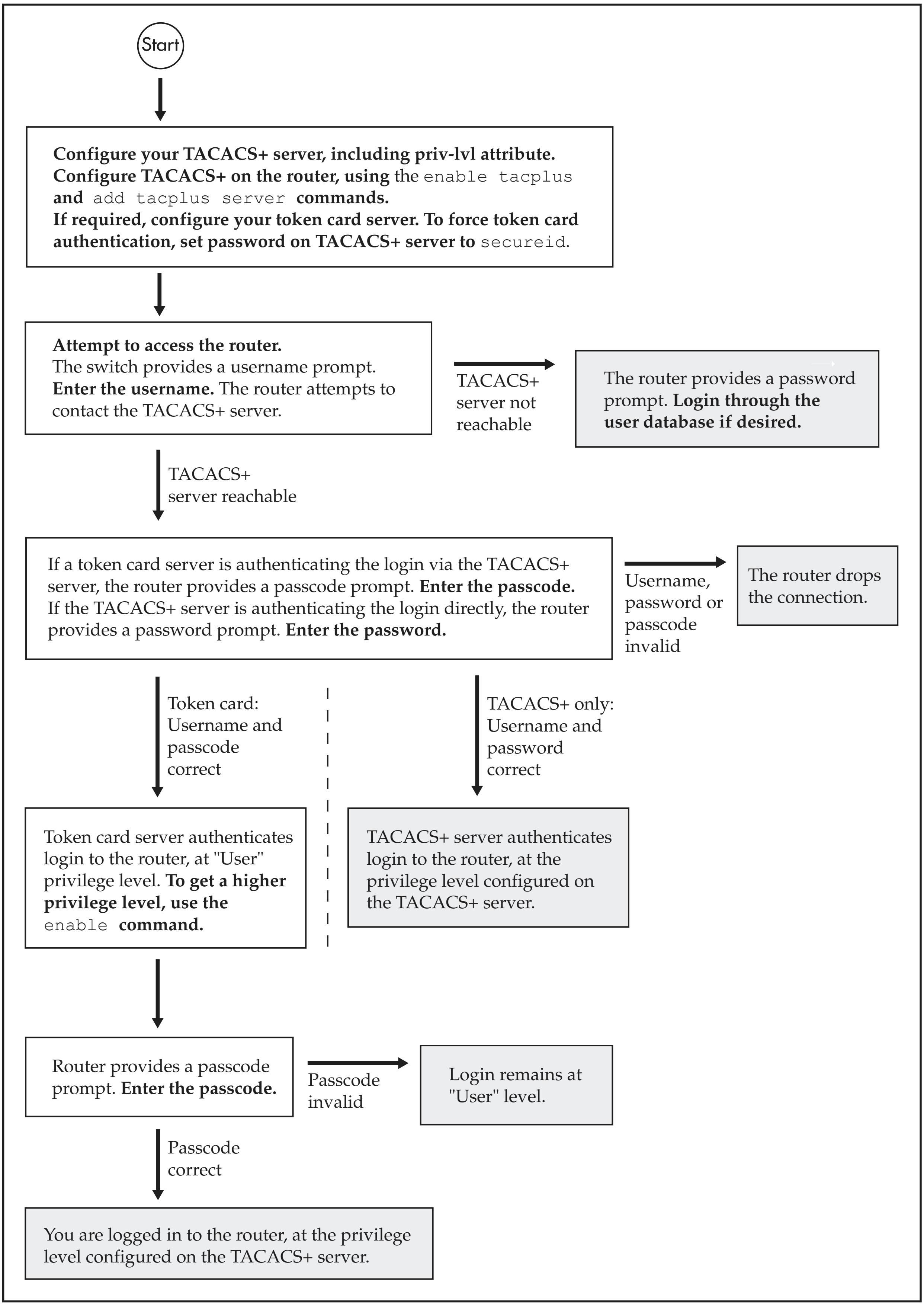
Configuring TACACS+

Use TACACS+ in one of the following ways:

- authentication through a TACACS+ server by itself, using a username/password pair
- in conjunction with a token card server, using a username/passcode pair. This provides stronger security.

Both procedures are summarised in Figure 1-4 on page 1-27, including the router's actions if the login fails.For more information about token card servers, and about using a TACACS+ server and a token card server together, see "Token Card Authentication" on page 1-29. For more information about using a TACACS+ server by itself, see "Logging onto the Router" on page 1-28.

Figure 1-4: The procedure for logging into the router and being authenticated with TACACS+


Start
Configure your TACACS+ server, including priv-lvl attribute. Configure TACACS+ on the router, using the enable tacplus and add tacplus server commands. If required, configure your token card server. To force token card authentication, set password on TACACS+ server to secureid.
Attempt to access the router. The switch provides a username prompt. Enter the username. The router attempts to contact the TACACS+ server.
TACACS+ server not reachable
The router provides a password prompt. Login through the user database if desired.
TACACS+ server reachable
If a token card server is authenticating the login via the TACACS+ server, the router provides a passcode prompt. Enter the passcode. If the TACACS+ server is authenticating the login directly, the router provides a password prompt. Enter the password.
Username, password or passcode invalid
The router drops the connection.
Token card: Username and passcode correct
TACACS+ only: Username and password correct
Token card server authenticates login to the router, at "User" privilege level. To get a higher privilege level, use the enable command.
TACACS+ server authenticates login to the router, at the privilege level configured on the TACACS+ server.
Router provides a passcode prompt. Enter the passcode.
Passcode invalid
Login remains at "User" level.
Passcode correct
You are logged in to the router, at the privilege level configured on the TACACS+ server.

Configuring the Router

To enable TACACS+, use the command:

```
enable tacplus
```

To tell the router to attempt authentication through a TACACS+ server, use the command:

```
add tacplus server=ipaddress [key=key] [port=port]
    [singleconnection={yes|no] [timeout=1..10]
```

For example, to add a TACACS+ server with IP address 192.168.0.1, key ABCD123 and a timeout of 5, use the command:

```
add tacplus server=192.168.0.1 key=abcd123 timeout=5
```

Configuring the TACACS+ Server

To determine the appropriate privilege level for the user, the router uses the TACACS+ priv-lvl value. You need to set the server to return an appropriate value (see Table 1-5).

Table 1-5: The values corresponding to privilege levels

Privilege Level	Value of TACACS+ priv-lvl
Security Officer	15
Manager	7-14
User	1-6
not mapped	0

Logging onto the Router

To access the router's CLI securely over a network, you must also use secure shell. See the Secure Shell chapter of the Software Reference for information and command syntax.

To log onto the router with the privilege level specified on the TACACS+ server

1. **Enter your username**

   On your terminal, terminal emulator or SSH window, enter your username at the username prompt.

2. **Enter your password**

Results

The username/password pair is either accepted or rejected. If the TACACS+ server accepts the pair, you are logged in at the appropriate security level. If the TACACS+ server rejects the pair, the router breaks the connection. To increase security, the router checks only the username/password against another authentication system (such as the user database) if the TACACS+ server is unavailable.

This procedure and its results are summarised in Figure 1-4 on page 1-27, including the router's actions if the login fails.

TACACS+ and Telneting from the Router

If your login to the router is authenticated using TACACS+, you can only telnet from the router if your TACACS+ privilege level is also equal to or higher than the minimum TACACS+ privilege level required for using telnet on the router. By default, no TACACS+ users can use telnet on the router. To set a privilege level, use the command:

```
set tacplus telnet={0..15|none}
```

A value of **none** is the default and disables telnet for all TACACS+ authenticated users. A value of **1** indicates that all users can telnet. A value of **7** indicates that Manager privilege or better is required. A value of **15** is equivalent to Security Officer privilege.

Note that a user can have a TACACS+ privilege level that is equivalent to User or Manager but be unable to use telnet on the router if the TACACS+ privilege level required for using telnet is higher than the privilege level they have been assigned. For example, if the TACACS+ privilege level required for using telnet on the router is set to 10 and there are two users with Manager privileges, one with a TACACS+ privilege level of 9 and one with a privilege level of 10, only the user with a privilege level of 10 can use telnet on the router.

To see the required privilege level, use the command:

```
show tacplus telnet
```

Token Card Authentication

Token card authentication is an authentication process that uses three pieces of information to authenticate users. This makes it more secure than systems that use two forms of identification. The three pieces of information are a:

- username
- 4-digit PIN, which the user must remember
- token card

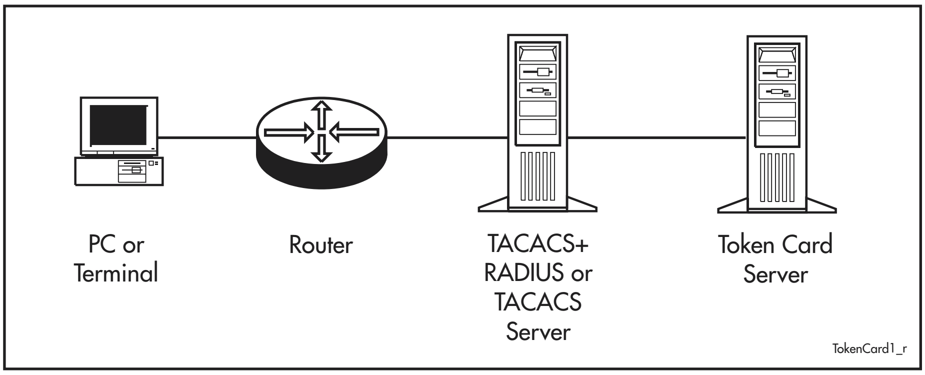
The *token card* is a form of identification that is about the size of a credit card and has a 9-digit LCD display. The number on the LCD display changes every 30 seconds and is synchronised to the token card server so the server can authenticate the 9-digit number.

The user’s 4-digit PIN followed by the number displayed on the token card forms a *passcode*. The token card server authenticates users based on their usernames and passcodes.

Token Card Authentication on the Router

The router communicates with a token card server through a TACACS+, RADIUS or TACACS server (Figure 1-5).

Figure 1-5: The elements of an authentication system that uses token card authentication.



The TACACS+, RADIUS or TACACS server is between the token card server and the router, and may hold further information about the user. For TACACS+ and RADIUS servers, this information can include a privilege level, so that the user can be authenticated on the router at Manager or Security Officer level. Token card servers return an accept or reject message, and therefore do not support different privilege levels.

Using Token Card with TACACS+

You can use a token card server in conjunction with a TACACS+ server to log users onto the router at User, Manager or Security Officer privilege level.

Configuring the TACACS+ Server

To determine the appropriate privilege level for the user, the router uses the TACACS+ priv-lvl value. You need to set the server to return an appropriate value (see Table 1-5 on page 1-28).

To ensure that the TACACS+ server uses the token card server for authentication, set the password attribute on the TACACS+ server to "secureid".

Logging onto the Router at User Privilege Level

To access the router securely over a network by using secure shell, configure secure shell on the router. See the Secure Shell chapter of the Software Reference for information and command syntax.

To log onto the router at user privilege level

1. **Enter your username**

   On your terminal, terminal emulator, or SSH window, enter your username at the username prompt.

2. **Enter your passcode**

   The passcode is your 4-digit pin followed by the 9-digit token card number.

Results

The username/passcode pair is either accepted or rejected. In summary, the message exchange between the router and the servers is:

1. The router sends the username and passcode to the TACACS+ server.
2. The server checks its database for a match, but does not find one, because it does not have a record of the passcode.
3. The server sends the username and passcode to the token card server.
4. The token card server checks its database for a match. If a match exists, it sends an accept message to the TACACS+ server. If no match exists, it sends a reject message.
5. The TACACS+ server returns the appropriate accept or reject message to the router.
6. If the token card server accepted the username/passcode pair, the user is logged into the router with "User" privilege.

   If the token card server rejected the username/passcode pair, the router drops the connection.

This procedure and its results are summarised in Figure 1-4 on page 1-27, including the router's actions if the login fails.

Logging onto the Router at Higher Privilege Levels

The TACACS+ server can also hold user privilege level information. See "Configuring the TACACS+ Server" on page 1-30 for information on appropriate settings for the server.

To log onto the router at manager or security officer privilege level

1. **Log into the router with user privilege.**

   See "Logging onto the Router at User Privilege Level" on page 1-30.

2. **Request a higher privilege level.**

   Enter the command:

   ```
   enable
   ```

   and then enter the passcode at the passcode prompt.

Results

The message exchange between the router and the server is:

1. The router queries the TACACS+ server.
2. The server returns the priv-lvl value that matches this username.
3. The user is logged into the router with the privilege level indicated by the priv-lvl value

Using Token Card with RADIUS or TACACS

You can use a token card server in conjunction with a:

- RADIUS server to log users onto the router at User, Manager or Security Officer privilege level
- TACACS server to log users onto the router at User privilege level only

Configuring your RADIUS Server

To determine the appropriate privilege level for the user, the router uses the RADIUS Service-Type attribute value. You need to set the server to return an appropriate value (see Table 1-6).

Table 1-6: The values corresponding to privilege levels.

Privilege Level	**Value of RADIUS Service-Type attribute**
Security Officer	Administrative (6)
Manager	NAS prompt (7)
User	any other value, or no value

Logging onto the Router at User Privilege Level

To access the router securely over a network by using secure shell, configure secure shell on the router. See Chapter 43, Secure Shell for information and command syntax.

To log onto the router at User privilege level

1. **Enter your username**

   On your terminal, terminal emulator, or SSH window, enter your username at the username prompt.

2. **Enter your passcode**

   The passcode is your 4-digit pin followed by the 9-digit token card number.

   Enter your passcode at the password prompt if appropriate. The router does not provide separate password and passcode prompts for RADIUS or TACACS servers.

Results

The username/passcode pair is either accepted or rejected. In summary, the message exchange between the router and the servers is:

1. The router sends the username and passcode to the RADIUS or TACACS server.
2. The server checks its database for a match, but does not find one, because it does not have a record of the passcode.
3. The server sends the username and passcode to the token card server.
4. The token card server checks its database for a match. If a match exists, it sends an accept message to the RADIUS or TACACS server. If no match exists, it sends a reject message.
5. The RADIUS or TACACS server returns the appropriate accept or reject message to the router.
6. If the token card server accepted the username/passcode pair, the user is logged into the router with user privilege.

   If the token card server rejected the username/passcode pair, the router's User Authentication Facility attempts to authenticate the user using the next possible approach (see "User Authentication Facility" on page 1-15).

Logging onto the Router at Higher Privilege Levels

If the router communicates with a token card server via a RADIUS server, the server can hold user privilege level information. See "Configuring your RADIUS Server" on page 1-32 for information on appropriate settings for the server.

To log onto the router at manager or security officer privilege level

1. **Log into the router with user privilege.**

   See "Logging onto the Router at User Privilege Level" on page 1-32.

2. **Request a higher privilege level.**

   For TACACS+, enter the command:

   ```
   enable
   ```

   For RADIUS, login with another username/password pair that has the appropriate privilege level, by using the command:

   ```
   login username
   ```

   Then enter the password at the prompt.

Results

The message exchange between the router and the server is:

1. The router sends the username/password pair to the server.
2. The server checks its database for a match. When a match exists, it sends an accept message to the router, including the Service-Type attribute value. When no match exists, it sends a reject message.
3. If the server accepted the username/password pair, the user is logged into the router with the privilege level indicated by the Service-Type attribute value (Table 1-6 on page 1-32).

If required, you can use a TACACS server to access the router at user level, and then a RADIUS server to obtain a higher privilege level.

RADIUS, TACACS and TACACS+ Debug Support

Access control packet debugging allows the contents of the packets to be viewed. The debugging commands allow both raw (hexadecimal dumps) and/or decoded (human-readable) packet displays. Information on any errors occurring in the transactions can be displayed once the appropriate debugging command is issued.

Only users with Security Officer privileges in system secure mode can enable RADIUS and TACACS+ debugging.

To enable debugging on TACACS+, use the command:

```
enable tacplus debug
```

S/Key and OTP One-Time Password Systems

S/key and OTP are *one-time password* systems designed to protect networks from attacks via electronic eavesdropping during user authentication. With both systems, a user never logs into a server on the network using the same password more than once. Since a specific one-time password can authenticate a user only once, even if the password is intercepted by a malicious user enroute to the authentication server (via a sniffer), by the time they try to gain access to the system with it, it is longer be valid.

The S/Key system generates one-time passwords by applying a one-way MD4 hash function to the concatenation of a user-specified *seed* and secret password. A seed is a user-defined string used during initialisation of the one-time password system on the authentication server. The secret password should never be transmitted across the network and hence is safe from eavesdroppers. At initialisation time, the S/Key system is given a user-specified sequence number, and the one-way function is applied that number of times to produce the first one-time password. The sequence number decrements each time the user logs in. The hash function is described as one-way since it is almost impossible to apply the inverse function to calculate the next password in the sequence.

The OTP system is based on the original S/Key implementation. In both systems, the one-time password generation process is similar, but with OTP, the user-specified seed is internally converted to lower case, and there are more stringent requirements on the length of the initialisation password (it must be 10-63 characters long as opposed to S/Key, where it must be 8 or more characters long).

This implementation provides support for both S/Key and OTP using the following one-way hash functions:

- OTP using MD4
- OTP using MD5.
- S/Key using MD4.
- S/Key using MD5.

Initialising the S/Key or OTP System on the Authentication Server

The authentication server must support either S/Key or OTP. The server must be initialised for each user requiring access to the router using one-time passwords. Initialisation should take place either on the server itself, or via a secure local terminal so that there is no chance of the S/Key or OTP initialisation password being intercepted during transit across a network. At initialisation time, each user must specify:

- A secret initialisation password
- A seed, made up of 1-16 alphanumeric characters
- An initialisation sequence number, from 1-999

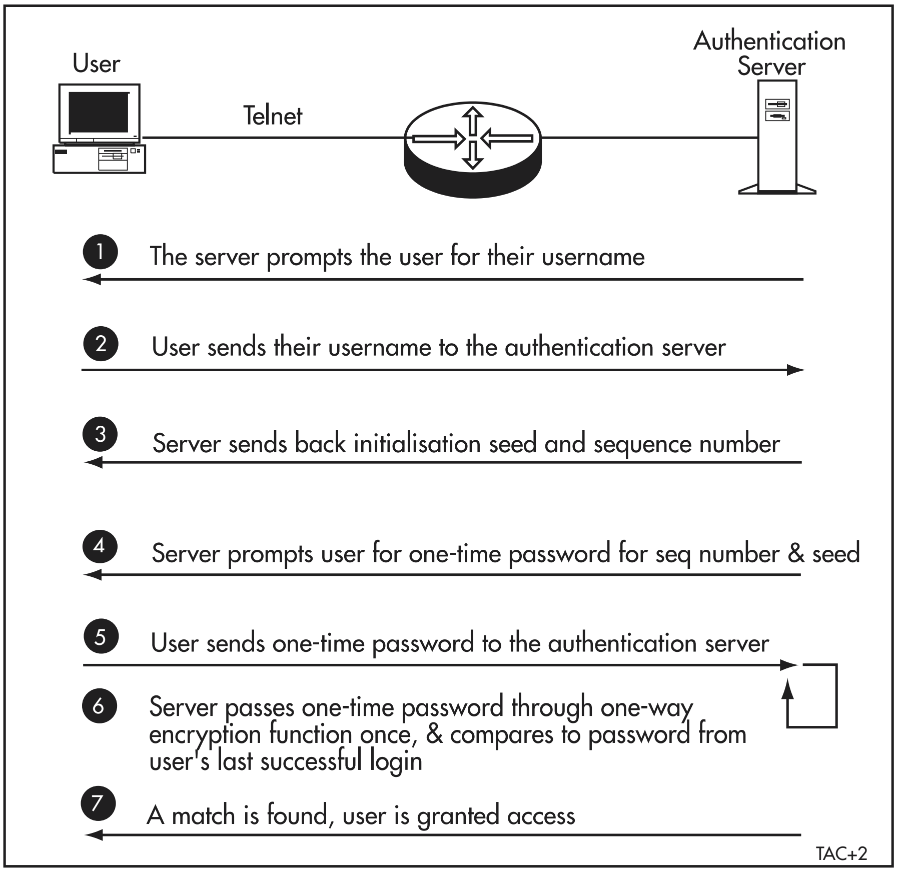
The server now accepts user authentication requests.

The steps for logging into a client are described below. The numbers correspond to those illustrated in Figure 1-6 on page 1-35.

1. The user is prompted for their username.
2. The client transmits the username to the authentication server.
3. The server searches through its database to find the current username. If the username is found, the server transmits the user's initialisation seed and current sequence number back to the client. If the username is not found the login is rejected.
4. The user is prompted to supply the one-time password for the given seed, and sequence number.
5. The one-time password is transmitted to the authentication server.
6. The server passes the received one-time password through the one-way encryption function once, and compares the result to the one-time password from the user's last successful login.
7. If they match, the authentication passes and the user is granted access.

The current sequence number is decremented by one each time a user successfully logs in to the system. The user must reinitialise the S/Key server before the sequence number equals zero.

Figure 1-6: Steps for logging into a client.


User
Authentication Server
Telnet
1 The server prompts the user for their username
2 User sends their username to the authentication server
3 Server sends back initialisation seed and sequence number
4 Server prompts user for one-time password for seq number & seed
5 User sends one-time password to the authentication server
6 Server passes one-time password through one-way encryption function once, & compares to password from user's last successful login
7 A match is found, user is granted access
TAC+2

Configuring S/Key and OTP on the Router

To set the method of authentication that the router is to use and the type of encryption, use the command:

```
set skey [method={skey|otp}] [encryption={md4|md5]
```

To calculate and display (Figure 1-7 on page 1-36) one-time passwords, use the **sequence** and **seed** parameters in the command:

```
show skey [sequence=seq_no seed=seed_name [number=value]]
```

where:

- *seq_no* is an integer from 1 to 9999 representing the sequence number of the last S/Key or OTP password to be generated.
- *seed_name* is the 1-16 alphanumeric user-defined string used to initialise the one-time password system on the authentication server.
- *value* is an integer from 1 to 99 representing the number of consecutive S/Key or OTP passwords to generate, finishing at *seq_no.*

To display the correct one-time passwords, users must supply their current sequence number and seed. They are then asked to enter the password used when initialising their current sequence of one-time passwords on the authentication server. The password is not echoed to the screen when entered. The output shows the sequence of S/Key or OTP one-time passwords to be used for a user's subsequent login attempts.

Figure 1-7: Example output from the **show SKEY SEQ=n SEED=seed** command.

```
Enter S/KEY initialisation password :
Computing SKEY passwords using MD4....
----------------------------------------------
Seq No        One-Time Password
95            IT DOLT ROOM NET GLUT ROWE
96            DARE MOS SARA GOAD MAO LEO
97            GUN TAIL MEND EAT INCH JOHN
98            EARN KID CARE HELD GIRD WINE
99            ADAM WARD DECK PLY EGAN WEED
----------------------------------------------
```

Remote Management

Managing remote routers is as easy as managing the local router to which the terminal is connected. From a terminal connected to any port (with either User or Manager privilege), use the command:

```
telnet ipadd
```

to Telnet to the remote router, specifying the remote router's IP address. If the connection is successful a login prompt from the remote router is displayed. Login using a login name that has been defined with Manager privilege (such as the default Manager login name), and enter the password.

To return to the local router, terminate the connection by using the command:

```
logoff
```

For more information about using Telnet, see Chapter 21, Terminal Server.

Monitoring and Fault Diagnosis

Event Logging

The router responds to certain significant events by generating an event log message. Each router maintains a local event log of the most recent log messages. To view the log, use the command:

```
show log
```

The logging facility provides a powerful, flexible and easily configurable tool for monitoring network activity and selecting and displaying the results. User-defined output definitions can filter, prioritise and output log messages to RAM, an asynchronous port, another router, a syslog server or an email address. See Chapter 33, Logging Facility for a detailed description of the logging facility.

Restarts

Some changes to configuration parameters require the router to be restarted for the changes to take affect. The router is restarted with the command:

```
restart {reboot|router} [config={filename|none}]
```

If the router encounters a fatal error condition from which it cannot recover, it automatically performs a restart. The reason for the restart may be determined by examining the router's exception list by using the command:

```
show exception
```

The conditions that the router encountered when it last restarted, such as the amount of RAM and the state of the battery-backed RAM, can be viewed with the command:

```
show startup
```

A complete snapshot of the state of the router prior to the last fatal condition can be displayed with the command:

```
show debug
```

CPU Utilisation

The CPU utilisation over the last second, ten seconds, one minute or since the router last restarted can be displayed with the command:

```
show cpu
```

Memory

The state of the router's buffer pool can be examined with the command:

```
show buffer
```

If the pool of free buffers drops below a critical threshold, the router progressively disables processes, resulting in a loss of functionality. This problem can potentially arise when a fast source sends enormous amounts of data to a slow destination or down a slow link. However, the cause is more likely to be a problem with the router itself. The problem can be corrected in the short term by restarting the router but report it to your supplier.

Fast buffer memory (on power PC based routers only) is cached by the CPU and is available only for program variable storage. It cannot be used for packet buffers.

The contents of memory can be examined with the command:

```
dump
```

Contents can be modified with the command:

```
modify
```

The **dump** command on page 1-83 and the **modify** command on page 1-104 are provided as diagnostic tools and should not be needed for normal operation of the router. Inappropriate use of these commands may cause a malfunction of the router, resulting in the loss of network services.

Flash Memory

Flash memory is a non-volatile, reusable memory device that allows the router to store large volumes of data (up to 8 MByte on the AR410 and AR410S, up to 16 MByte on the AR440S, AR441S, AR450S). Software releases and patches, and configuration files are stored in Flash memory by default. Releases can be remotely loaded into flash memory from any router port using the Loader Module. Multiple software releases can be loaded and then individually selected for use at runtime by the Install Module. Comprehensive management features are provided to examine the state of the flash memory and to view or modify the contents.

To enable flash memory to support applications other than software releases, it is structured like a disk subsystem with files that can be created, deleted, read, and written by any router module. Files can also be manipulated directly using the command line interface. This allows flash to be used to store any type of data, including releases, patches, configurations, and logs.

Physical Characteristics

Flash memory is a special type of non-volatile memory that can be erased and reprogrammed many times in-situ. Flash memory has advantages over other types of non-volatile memory in that it has a very large storage capacity and it does not require power from a battery to retain stored data. The main limitations of flash memory are that it has a fixed erase block size, so individual bytes cannot be changed without first clearing a whole block of data, and there is a limit on the number of erase cycles that can be done. However, the erase limit is very high, typically at least 100000 cycles, which would allow three erases per day for 100 years before the limit was exceeded.

See the *AR400 Series Router Hardware Reference* for more information on memory specifications.

The presence and amount of flash memory installed is displayed by using the command:

```
show system
```

More detailed information about the flash memory can be displayed by using the command:

```
show flash physical
```

The File Subsystem

The file subsystem provides a consistent file-based interface to the physical memory device on the router used for data storage, flash memory. The file subsystem allows data, such as software releases, licence information and configuration scripts, to be stored on the router in a file structure and manipulated in the same way with the same commands.

File Naming Conventions

The file subsystem provides a flat file system—directories are not supported. Files are uniquely identified by a file name in the following format:

```
[device:]filename.ext
```

where:

- *device* specifies the physical memory device where the file is stored. If *device* is not specified, the default is flash.
- *filename* is a descriptive name for the file, and may be one to twenty eight characters long. Valid characters are lowercase and uppercase letters, digits (0–9), and the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = “| \ [ ] ; : ? / , < >.
- *filename* (for the **show file** and **delete file** commands only) is a descriptive name for the file, and may be one to twenty eight characters long. Invalid characters are “ \ ; ? / , <. Valid characters are:
    - uppercase and lowercase letters
    - digits (0–9)
    - the characters ~ ’ ! @ # $ % ^ & ( ) _ - { } * > [ ] | :

  Wildcard characters * may appear anywhere in the filename. The wildcard character matches any string.

  Character ranges may be specified using the > character, for example a>z matches any letter in the alphabet. The + character may be used to specify a list of options, for example a*.scp+b*.scp would specify files that match a*.scp or b*.scp.

  Square brackets may be used, for example ppp*.[scp+cfg] matches scripts and configuration files whose names start with "PPP".

  The vertical bar | character matches any single character. For example, |||.scp matches script files with names three characters long (excluding extension and device name).

  If a colon is seen anywhere in the filename, the device parameter is ignored and it is assumed that the filename includes the device name.

- *ext* is a file name extension one to three characters long. Valid characters are:
    - uppercase and lowercase letters
    - digits (0–9)
    - hyphen (-)

  The extension is used by the router to determine the data type of the file and how to use the file (Table 1-7). If *ext* is specified, it must be separated from the filename by a period.

Table 1-7: File extensions and file types .

Extension	File type/function
CFG	Configuration or boot script
FBR	Flash Boot software Release
HLP	Help file
HTM	HTML file used by the HTTP server
LIC	Licence information
LOG	Log file
MDS	Modem script
PAT	Patch
PAZ	Compressed patch
REL	Software release
REZ	Compressed release
RSC	GUI resource file
SCP	Script
TXT	Generic text file
LFN	Extension used for the long file name translation table

The following is an example of a valid file name:

`flash:config.scp`	A script file.

The following are examples of illegal file names:

`flash:/sys/head_o.cfg`	"/" is not a valid delimiter character, and directories are not supported.

Long file names usage in software releases

All software releases support short filenames (DOS 8.3 format). Software release 2.5.1 and later support long file names in either DOS 16.3 or DOS 28.3 format. The following table summarises which software releases support different DOS filename formats.

Table 1-8: The DOS filename formats supported by different software releases

Software release	Dos 8.3 format	DOS 16.3 format	DOS 28.3 format
2.4.x and earlier	Yes	No	No
2.5.1 and later	Yes	Yes	No
2.6.4 and later	Yes	Yes	Yes

Upgrading to new software releases

When upgrading to software release 2.6.4 from previous software releases, file names retain their DOS naming format. DOS 8.3 format filenames remain in DOS 8.3 format and DOS 16.3 format filenames remain in DOS 16.3 format.

Regressing to previous software releases

If software release 2.6.4 is installed on the router and then a previous software release that supports **only** DOS 8.3 format is installed (Table 1-8), long file names that were in DOS 28.3 format are truncated to DOS 8.3 format. When software release 2.6.4 or later is reinstalled, these truncated file names are restored to their DOS 28.3 format and no information is lost. Support for long file names in only DOS 8.3 format is a feature of software releases prior to software release 2.5.1.

If software release 2.6.4 is installed on the router and then a previous software release that supports DOS 16.3 format is installed (see Table 1-8), long file names in DOS 28.3 format are permanently truncated to DOS 8.3 format. For example, the AB12345678.SCP file is permanently renamed AB123~01.SCP. Any long file names that were in DOS 28.3 format remain truncated in DOS 8.3 format when software release 2.6.4 is reinstalled. Support for long file names in DOS 16.3 format is a feature of software release 2.5.1 up to software release 2.6.4.

Using Wildcards to Specify Groups of Files

The asterisk character (*) may be used as a wildcard character in some commands to identify a groups of files to be processed by the command. The following are examples of valid wildcard expressions:

```
flash:*.*
*:*.rez
```

Working With Files

File names of up to twenty eight characters long, with extensions of three characters (DOS 28.3 format), are supported on the router. Files on the router are stored in flash using the DOS 8.3 format of eight characters long, with extensions of three characters. For example, the file `extralongfilenam.cfg` may be saved as `extral~1.cfg` in the Flash File System. Therefore, files can be accessed with two file names, either of which can be used for file management.

A translation table, named `longname.lfn`, converts file names between DOS 28.3 format and DOS 8.3 format. To reconcile file names the router consults the translation table which is synchronised with file contents in memory. If the translation table becomes corrupted it can be rebuilt from all valid files that are detected in memory. To resynchronise the translation table to the file contents in memory, use the command:

```
purge file translationtable={all|update}
```

The **update** option restores all valid long file names to the appropriate table entries after the table has been rebuilt. All long file names that are not reconciled to the new table, and all table entries that are not confirmed to be in memory, are deleted. This leaves a translation table that has maintained all of its previously valid data, and disposed of the rest. The table continues to support all subsequent long file name creation and management.

The **all** option completely rebuilds the translation table. All long file names are lost. The table continues to support all subsequent long file name creation and management.

To display the contents of the translation table, which converts file names between DOS 28.3 format and DOS 8.3 format, use the command:

```
show file=longfile.lfn
```

To display a directory of the files stored on the router, use the command:

```
show file
```

To limit the display to certain files, use the command:

```
show file=filename
```

filename may contain wildcard characters. Files can be permanently deleted using the command:

```
delete file=filename
```

filename may contain wildcard characters. Files can be created using the router's built-in editor by using the command:

```
edit [filename]
```

or by downloading the file via HTTP, TFTP or ZMODEM by using the command:

```
load file=filename
```

Flash File System

The Flash File System (FFS) provides additional functionality on top of that provided by the file subsystem, to manage the peculiarities of flash technologies. The additional functionality of the FFS includes:

- Header and data integrity is ensured with a checksum mechanism.
- All flash processes can recover from a power cycle without data loss.
- Automatic recovery of deleted file space by the compaction process.

Information about the state of the FFS can be displayed by using the command:

```
show flash
```

Working with FFS Files

FFS files can be managed like any other file on the router by using the standard file subsystem commands:

```
edit [filename]
delete file=filename
load file=filename
show file[=filename]
```

In addition, the following commands can be used to manage files stored in flash memory. To display a directory of the files stored in flash memory, use the command:

```
show ffile [check]
```

If **check** is specified, the file data checksum is also verified. This is an option because it takes longer to complete a check on large files. A file data check is also carried out each time the system reads a file.

A flash file can be deleted with the command:

```
delete ffile=filename
```

Wildcards are allowed in the *filename* and *ext* fields of the file name, but not in the device field. The file is marked deleted but the space that it occupies is not freed until the next compaction.

The flash memory can be completely erased with the command:

```
clear flash totally
```

This command totally erases all stored flash information and reformats the flash file structure.

Compaction

Flash memory has a granular erase structure that requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure, the FFS keeps track of the status of each file — whether it is being written, is complete, or is deleted. When the total amount of flash memory used for deleted files reaches a preset limit, a compaction process begins. Compaction searches through flash memory, copying good files to a new location. After the good files in an erase block have been copied, the block is cleared. Deleted files in the block being cleared are freed up to allow space for new files. When a large amount of flash memory is in use, compaction may take several seconds. However, flash memory operations continue to operate unaffected by the compaction process.

While flash is compacting, do not restart the router or use commands that affect the flash file subsystem. Do not restart the router, or create, edit, load, rename, or delete files until a message confirms that flash file compaction is complete. Interrupting flash compaction stops the process. Compaction of files will then be done on the next file delete if no **load** command is issued.

Compaction can be manually initiated with the command:

```
activate flash compaction
```

FFS Messages

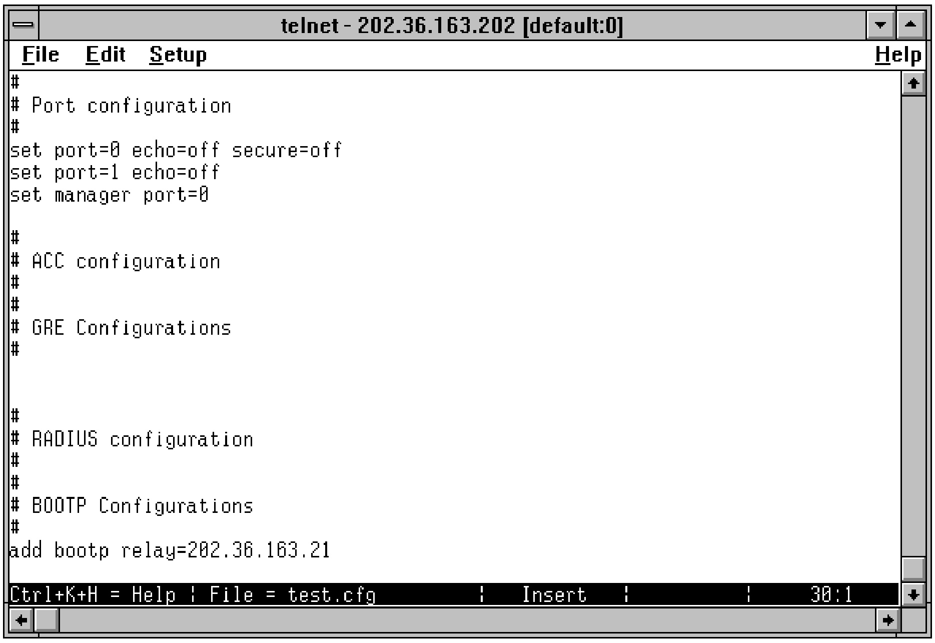
Some FFS processes generate messages in the system log (displayed with the **show log** command on page 33-34 of Chapter 33, Logging Facility) which include FFS message codes. See “Flash File System Message Codes” on page C-7 of Appendix C, Reference Tables for a complete list of the possible codes and their meanings.

The Built-in Editor

The router has a built-in full-screen text editor for editing ASCII text files stored on the router file subsystem.

The editor uses VT100 command sequences and should be used with a VT100-compatible terminal, terminal emulation program, or Telnet client. The VT100 screen supports 24 lines, unlike a PC. Lines 1–23 are used to display the text of the file being edited, and line 24 is used as the status bar and command line (Figure 1-8 on page 1-44). The status bar displays the current file name, line and column position in the file and the editing mode (overstrike or insert). When additional command information is required such as a file name or search text, a prompt is in the status bar.

Figure 1-8: The editor screen layout.



```
#
# Port configuration
#
set port=0 echo=off secure=off
set port=1 echo=off
set manager port=0

#
# ACC configuration
#
#
# GRE Configurations
#


#
# RADIUS configuration
#
#
# BOOTP Configurations
#
add bootp relay=202.36.163.21

Ctrl+K+H = Help ¦ File = test.cfg       ¦    Insert    ¦          ¦     30:1
```

The editor is invoked with the command:

`edit [`*`filename`*`]`

The file name is optional as a file can be loaded, or a new file can be created from within the editor itself. The editor is currently limited to editing one file at a time. To overcome this limitation use the cut and paste facility to transfer text between files.

Before starting the editor make sure your terminal, terminal emulation program or Telnet client is 100% compatible with a VT100 terminal.

Help can be obtained at any time while in the editor by pressing [Ctrl/K,H]; that is, holding down the Ctrl key and pressing in turn the K key then the H key.

HTTP Client and Server

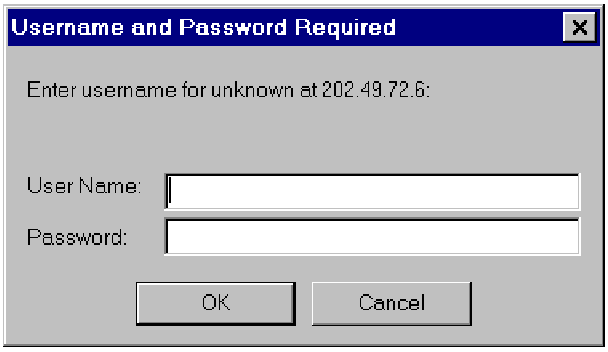
The router has a built-in HTTP client and server. The HTTP server is compatible with any HTTP/1.1-compliant browser and allows the router to serve HTML pages out of flash memory to a remote web browser. The HTTP server is enabled by default. To disable the HTTP server, or to enable the HTPP server after it has been disabled, use the commands:

```
disable http server
```

```
enable http server
```

When a user attempts to access the router via a web browser, the HTTP server requests authentication from the browser. The browser prompts the user for a username and password (Figure 1-9).

Figure 1-9: Logging in to the router from a web browser.



Username and Password Required
Enter username for unknown at 202.49.72.6:
User Name:
Password:
OK
Cancel

The username and password entered by the user must match a user defined in the User Authentication Database (see "The User Authentication Database" on page 1-16).

By default, the router's homepage is homepage.htm. This is the page the HTTP server returns when it receives a request that does not specify a particular page, and when no web-based GUI is installed on the router. When there is a web-based GUI, the router returns the GUI homepage when a request does not specify a page.

All "get", configure, and monitor requests as well as authorisation failures are logged to the Logging Facility (see Chapter 33, Logging Facility). Debugging can be enabled or disabled by using the commands:

```
enable http debug={all|auth|msg|session}
```

```
disable http debug={all|auth|msg|session}
```

Debug messages display authorisation attempts, HTTP "get" and "post" requests and responses, and TCP state changes. The currently enabled debugging options can be displayed by using the command:

```
show http debug
```

The following command restarts the HTTP server, disables debugging, and clears all counters:

```
reset http server
```

To display the current status of the HTTP server, use the command:

```
show http server
```

To display information about the currently active sessions on the HTTP server, use the command:

```
show http session
```

The HTTP client enables the router to act as a browser by sending HTTP "get" or "post" requests to another HTTP server. The HTTP client is used by the Configuration Wizard to download updates from a support web site. To display the current status of the HTTP client, use the command:

```
show http client
```

Resolving Uniform Resource Locators (URLs)

When the HTTP server receives a request for a URL, it uses the following methods to resolve the URL:

- When the URL matches the name of a file stored in the router's flash memory, the file is loaded and sent to the browser.
- When the URL does not match the name of a file stored in flash, the HTTP server searches a list of dynamically generated HTML pages for a match. When a match is found, the page is generated and sent to the browser.
- When the URL does not match the name of a file stored in flash or the name of a dynamically generated HTML page, the HTTP server returns the HTML error 404, indicating the URL could not be found.

Mail Subsystem

The router has a built-in email client and SMTP (Simple Mail Transfer Protocol) server to enable email messages to be sent from the router to remote mail systems using SMTP. The email client generates messages that comply with RFC 822, *Standard for the Format of ARPA Internet Text Messages*. The external SMTP server must be compliant with RFC 821, *Simple Mail Transfer Protocol*, for the transmission of mail messages. Note that Microsoft mail servers are not RFC 821 compliant.

The SMTP server transmits email messages only; it cannot accept emails from other mail systems.

A mail message is transmitted by using the command:

```
mail to=destination {file=filename|message=message}
    [subject=subject] [etrn=mail-domain]
```

from the router's command line prompt or from a script. Messages can also be transmitted automatically by the:

- Trigger Facility (Chapter 30, Trigger Facility)
- Logging Facility (Chapter 33, Logging Facility)
- Firewall (Chapter 41, Firewall).

The body of the message may contain either a single character string or the contents of a file in the router's flash memory.

The current state of the mail subsystem and the messages queued for transmission can be displayed by using the command:

```
show mail
```

Messages that are queued awaiting transmission can be deleted by using the command:

```
delete mail=id
```

The progress of mail messages can be monitored using the mail subsystem’s debugging option, which is enabled or disabled with the commands:

```
enable mail debug
disable mail debug
```

Configuration Examples

The following procedures illustrate the steps required to configure the mail subsystem and transmit email messages. It is assumed that IP has already been enabled and correctly configured on the router.

To configure the mail subsystem

1. **Configure a DNS Server.**

   Configure the IP address of the DNS server that the mail subsystem is to use to resolve email addresses into IP addresses. Without a DNS server the mail subsystem does not function.

   ```
   set ip nameserver=192.168.5.3
   ```

2. **Configure the mail host name.**

   Configure the host name used by the mail subsystem when communicating with other mail systems. Normally this is the fully qualified domain name of the router. Without a host name the mail subsystem does not function.

   ```
   set mail hostname=ho1.company.com
   ```

3. **Check the configuration.**

   Check that the mail subsystem is correctly configured and enabled.

   ```
   show mail
   ```

To send a file via email from the command prompt

1. **Send the file as the body of a mail message.**

   Text format files with CFG, SCP, and TXT extensions can be transferred from the router to a remote user in the body of an email message. For example, configuration scripts can be sent to a central host for management and change control. In this example, the boot.cfg file is sent to the network administrators email address *netman@company.com*:

   ```
   mail to=netman@company.com subject="Boot script for
      ho1.company.com" file=boot.cfg
   ```

2. **Check the progress of the message.**

   The progress of the message as it is transmitted to the remote mail system can be monitored by using the command:

   ```
   show mail
   ```

To transmit messages automatically using the Trigger Facility

1. **Create a script to generate a mail message.**

   Create a script called `mailcpu.scp` using the router's built-in editor that sends a message to the network administrator:

   ```
   edit mailcpu.scp
   ```

   The script contains the following line:

   ```
   mail to=netman@company.com subject="WARNING: Load high"
      message="CPU utilisation exceeded 80%"
   ```

   It is not necessary to identify the router in the *subject* field or the message because the mail system automatically inserts the router's host name in the *From* field of the message header.

2. **Create a trigger to activate the script.**

   Enable the Trigger module and create a trigger to activate the script when the router's CPU utilisation rises above 80%:

   ```
   enable trigger

   create trigger=1 cpu=80 direction=up script=mailcpu.scp

   show trigger=1
   ```

Loading Files onto the Router

The Loader module loads and stores releases, patches, PKI certificates, GUI resource files and other files into flash. The Loader module uses Trivial File Transfer Protocol (TFTP), Hypertext Transfer Protocol (HTTP), Lightweight Directory Access Protocol (LDAP) or ZMODEM over an asynchronous port, to retrieve files from a network host.

The loader can be configured with the command:

```
set loader
```

This command sets defaults for the name of the file to load, the network host to load it from, and the memory location in which to store the file. These defaults can be overridden when the load actually takes place. A time delay between initiating a load and the start of the load can also be configured.

The configuration of the Loader module can be displayed with the command:

```
show loader
```

This shows the default configuration for the Loader module as well as the status of any current file transfer.

To actually initiate a load, use the command:

```
load
```

This command uses defaults for the Loader module or values specified on the command line. The following command displays the progress of the load:

```
show loader
```

The current load can be stopped at any time by using the following command, which leaves the Loader module ready to load again:

```
reset loader
```

Only one file can be loaded at a time. Another load cannot be initiated while loading is in progress.

Trivial File Transfer Protocol (TFTP)

Trivial File Transfer Protocol (TFTP) is the TCP/IP standard protocol for file transfer with minimal capability and minimum overhead, based on UDP. This protocol is used for downloading patches and releases, and may be used to download other files. See "Software Releases and Patches" on page 1-51 for information about downloading patches and Chapter 46, Public Key Infrastructure (PKI) for information about certificates and CRLs.

The user can specify the TFTP server pathname to load from, the TFTP server file name to load and, if required, rename the file as it is saved to the router memory. If the TFTP server file is renamed with a filename greater than 8 characters long, with an extension of 3 characters (DOS 8.3 format), a DOS 8.3 format file name is dynamically allocated by the translation table. The file is saved to memory under this name and an entry is added to the translation table ("Working With Files" on page 1-41). All file management of this file is achieved by consulting the translation table.

To load a file onto the router with TFTP, use the command:

```
load [method=tftp] [delay=delay] [destfile=destfilename]
    [destnation={bootblock|flash}] [server={hostname|ipadd}]
    [srcfile|file=filename]
```

Hypertext Transfer Protocol (HTTP)

The router has a built-in HTTP client and server, which is described in "HTTP Client and Server" on page 1-45.

To load a file onto the router with the HTTP client, use the command:

```
load [method={http|web|www}] [delay=delay]
    [destfile=destfilename] [destination={bootblock|flash}]
    [httpproxy={hostname|ipadd} [password=password]
    [proxyport=1..65535]] [server={hostname|ipadd}]
    [servport={1..65535|default}] [srcfile|file=filename]
    [username=username]
```

Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is a lightweight network-layer protocol for accessing X.500-like directories. LDAP runs over TCP and uses a client/server model. Entries in an LDAP-accessible directory tree may be identified by a distinguished name (DN).

To load a file onto the router with LDAP, use the command:

```
load [method=ldap] [attribute={cert|crl|cacert}]
    [baseobject=dist-name] [delay=delay]
    [destfile=destfilename] [destination=flash]
    [password=password] [server={hostname|ipadd}]
    [servport={1..65535|default}] [username=username]
```

To remove all open LDAP requests and return the LDAP module to its original empty state, use the command:

```
purge ldap
```

This command is most likely to be useful if an LDAP request locks.

To see a summary of the outstanding LDAP requests, use the command:

```
show ldap
```

To see more detailed information about one or all LDAP requests, use the command:

```
show ldap request
```

For debugging purposes, some or all of the LDAP data can be displayed on the terminal as it is received. To enable debugging, use the command:

```
enable ldap debug
```

To disable debugging, use the command:

```
disable ldap debug
```

Distinguished Names (DN)

In an X.509 certificate, the identity of the certificate's subject is given in the form of a distinguished name. A distinguished name is a comma-separated list of parameters which together uniquely identify the subject of the certificate. A distinguished name also forms part of a Lightweight Directory Access Protocol (LDAP) URL, which is used to access an X.500-like directory.

A distinguished name can be set for the router by using the command:

```
set system distinguishedname=dist-name
```

The Certification Authority which is issuing the router with a certificate may require that a particular distinguished name be used. Otherwise, a logical distinguished name should be chosen.

The syntax of a distinguished name is:

```
distinguishedname="[cn=common-name, ]
    [list of [dc=domain-name-component, ]]
    [ou=organisation-unit-name, ] o=organisation-name, ]
    [street=street-address, ] [st=state-or-province-name, ]
    [l=locality-name, ] [c=country-name, ]"
```

where:

- *common-name*, *locality-name*, *organisation-name*, *organisation-unit-name*, *state-or-province-name* and *street-address* are all strings consisting of any printable characters, excepting quotation marks. The special characters {, = + < > # ; \ <CR> } must be escaped by typing a \ before the character, as defined in RFC 1779, *A String Representation of Distinguished Names*, unless they are used for their prescribed purpose. For example, to include a \ in the string, type \\ and to include a #, type \#.
- *domain-name-component* is a section of a fully-qualified domain name, in the format "dc=foo, dc=bar, dc=com" for the domain name foo.bar.com.
- *country-name* is a string consisting of any printable characters. Country names are generally given in the form of the two-letter ISO 3166 code for the country, for example, "us", "de", or "nz".

An example of a distinguished name, for Janet Bloggs at examplecompany.com, is:

```
distinguishedname="cn=Janet Bloggs, dc=examplecompany,
   dc=com, ou=Example Department, o=Example Company,
   street=Somewhere St, c=us"
```

This example is very complete, for illustrative purposes. The domain name and the parameter set which provides the person's physical location (organisation unit, organisation, street address and locality) are two different methods of specifying the address. Combining them is unlikely to be useful.

The order of the parameters is significant because a distinguished name is not be recognised if the order of the parameters is changed after definition. In general, the name should begin with the most specific element (the common name) and end with the most general (the country).

Spaces between the comma-separated items in the list are optional and are ignored by the router. When the distinguished name contains spaces, either between the items or within any item, quotation marks are required.

LDAP URLs

The location of a file in an LDAP-accessible directory tree is given by an LDAP Universal Resource Locator (URL). An LDAP URL consists of the word "ldap", followed by an address, an optional port number and a distinguished name.

The syntax of an LDAP URL is:

```
ldapurl=ldap://address[:port]/[base-object]
```

where:

- *address* is an IP address in dotted decimal notation or a host name from the host name table. See the **ping** command on page 14-129 of Chapter 14, Internet Protocol (IP) for information on determining the host name.
- *port* is an integer between 1 and 65535.
- *base-object* is a distinguished name, as defined in "Distinguished Names (DN)" on page 1-50.

If an LDAP URL contains any spaces, the URL must be in double quotes.

Software Releases and Patches

Software releases contain the code that enables the router to run. Patches contain small interim fixes and enhancements to the code. Both releases and patches can be downloaded to upgrade your router (see "Downloading Releases, Patches and GUI resource files into the Router" on page 1-55).

Releases

A software release contains a copy of the system software that executes on the router. Releases are given numbers that look like "2.6.1". In this case the major release number is "2", the minor release number is "6", and the interim release number is "1". A release is stored in flash memory.

A standard release is a single file with a name in the format:

```
mmm-rrr.rel
```

where *mmm* identifies the router model (Table 1-9 on page 1-52) and `rrr` is the release number (e.g. `231` for Software Release 2.3.1).

A compressed release is a single file with a name in the format:

```
mmm-rrr.rez
```

Table 1-9: Software Release filename formats.

Filename Format	Model
52-rrr.REL, 52-rrr.REZ	AR725, AR745, AR410 and AR410S router
54-rrr.REL, 54-rrr.REZ	AR440S, AR441S, AR450S router

Compressed releases are supported by the base flash boot block and the file required for a compressed release is:

```
mm-rrr.rez
```

Releases stored in flash are subject to licencing. A flash release may be downloaded into the router, but cannot be used until the correct licence information is entered.

Licence information is supplied by your authorised distributor or reseller with each software release.

The licence is encoded and is specific to a particular router and a particular release. A licence may be a full licence or a 30-day licence. A 30-day licence expires after 30 days; a full licence does not have an expiry date. To enable a release licence use the command:

```
enable release
```

To disable a release licence, use the command:

```
disable release
```

To display the current status of release licences in the router, use the command:

```
show release
```

A number of releases can be stored in the router at once. The flash bootblock or EPROM release is always available, and one or more releases may be stored in the flash file system. The router contains Install information that specifies the release (EPROM or one of the flash releases) to be loaded at boot. This information may be changed at any time. The Install information can be overridden so that the release in the flash bootblock or EPROM is loaded.

A software release is specific to a particular router series. It is not possible to run a release on a router series other than that for which the release was made. The same router release may, however, run on several models in the same series. If an attempt is made to load the wrong software release into the router, the boot process fails.

Patches

A router patch contains changes to the router software. A patch usually contains fixes to software errors, although enhancements to the software may sometimes be released as patches. Patches are identified by names like “231-01”. In this case, “2.3.1” is the release that the patch modifies, and “01” is a version number that identifies the patch in a series (1, 2, 3...) of patches. Patches are specific to a particular release, and thus to a particular router family. Attempts to use a patch with a non-matching release results in failure.

A standard patch is a single file with a name in the format:

```
mmrrr-pp.pat
```

where *`mm`* identifies the router model (Table 1-9 on page 1-52), *`rrr`* is the release number (e.g. `231` for Software Release 2.3.1), and *`pp`* is the version number of the patch.

Compressed patches consist of a single file:

```
mmrrr-pp.paz
```

Patches are loaded into flash.

The Install information that specifies which release to use also contains information about the patch. It is possible to load a router with a number of different patches, but only one patch can be run at a time.

Router Startup Operations

When the router boots, the following sequence of operations is performed:

1. Perform startup self tests.
2. Perform the install override option.
3. Load the FLASH boot release as the Install boot.
4. Inspect and check Install information.
5. Load the required release as the main boot.
6. Start the router.
7. Execute the boot script if one has been configured.

If a terminal is connected to asyn 0, a series of status and progress messages, similar to those shown in Figure 1-10, are displayed during the startup process.

Figure 1-10: Router startup messages.

```
INFO: Self tests beginning.
INFO: RAM test beginning.
PASS: RAM test, 4096k bytes found.
INFO: BBR tests beginning.
PASS: BBR test, 128k bytes found.
PASS: BBR test. Battery OK.
INFO: Self tests complete
INFO: Downloading router software.
Force EPROM download (Y) ?
INFO: Initial download succeeded
INFO: Executing configuration script <boot.cfg>
INFO: Router startup complete

Manager >
```

The startup self tests check the basic operation of the router. A router that passes these tests should be able to at least proceed far enough to perform the load of the FLASH boot release and to start operating.

The install override option is designed to allow a mandatory router boot from the FLASH boot release. The message:

```
Force EPROM download (Y)?
```

is displayed on the terminal connected to asyn 0 and the router pauses. If a key is not pressed within a few seconds, the startup process continues and all steps in the sequence are executed. If the [Y], [S] or [Ctrl/D] keys on the terminal are pressed immediately after the message is displayed, the router startup process can be altered (Table 1-10). To override the install sequence, press the required key repeatedly during startup.

Table 1-10: Router startup sequence keystrokes.

Pressing key...	Forces the router to...
Y	Load the FLASH boot release, with no patch, and skip straight to step 6.
S	Start with the default configuration. Any boot script is ignored.
[Ctrl/D]	Enter diagnostics mode.

The FLASH boot release is always loaded first when starting the router. This release contains all the code required to obtain and check the Install information. The first boot is called the *Install* boot. The Install information is inspected and the router set up to perform another load. Even if the actual release required is the FLASH boot release, another load is always performed. At this point the patch load, if required, is also performed.

The router startup occurs immediately after the install override option, or after the Install information check. This performs a full startup of router software and initiates the normal operation of the router.

Finally, if a boot script has been defined, the script is executed.

Downloading Releases, Patches and GUI resource files into the Router

The Loader module is responsible for loading and storing releases and patches into flash. The Loader module uses the Trivial File Transfer Protocol (TFTP), Hypertext Transfer Protocol (HTTP) or ZMODEM over an asynchronous port, to retrieve files from a network host. The FFS module is used to create, write, and destroy release and patch files.

Information on using the Loader module is in “Loading Files onto the Router” on page 1-48. Once the release or patch file is loaded, its presence can be checked by using the commands:

```
show file
```

A release or patch file can be removed with the command:

```
delete file
```

Files to be loaded by the Loader module must be resident on a TFTP server accessible via the network, or accessible via the ZMODEM protocol over an asynchronous port. Release and patch files are ASCII files, and consist of a header followed by a sequence of Motorola S-records containing the actual code for the release or patch. The header has a standard format, which provides information about the release or patch to the router.

The header in the release or patch file should not be altered. At best, this causes the file load or install to fail; at worst, the router could be put into a state where it cannot boot correctly until field service action is taken.

Install Information

The Install module maintains install information and loads the correct install at boot. An *install* is a record identifying a release and an optional patch. The Install module has three types of installs: temporary, preferred, and default.

The default install is the install of last resort. The release for the default install cannot be changed by the manager and is always the FLASH boot release. The patch for the default install may be set by the manager.

The temporary and preferred installs are completely configurable. Both the release and an associated patch may be set. The release may be the flash boot release or a release stored in the flash file system.

The three different installs are required to handle the following situations:

- A default install is required to handle the case when only the FLASH boot release is present.
- A temporary install is required to allow a release and/or patch to be loaded once only, in case it causes a router crash.
- A preferred install is required because the default install cannot be anything other than the FLASH boot release.

The install information is inspected in a strict order. The temporary install is inspected first. If this install information is present, the temporary install is loaded. At the same time, the temporary install information is deleted. This ensures that if the router reboots immediately as the result of a fatal condition caused by the temporary install, the temporary install is not loaded a second time.

If there is no temporary install defined, or the install information is invalid, the preferred install is inspected. If present, this install is loaded. The preferred install information is never deleted.

If neither temporary nor preferred installs are present, the default install is used. The default install is always in the router because the Install module restores it when it is not present.

To change the install information in the router, use the command:

```
set install
```

To delete a particular install (except the default install), use the command:

```
delete install
```

To display the current install information, including which install is currently running in the router, and how the install information was checked at the last reboot, use the command:

```
show install
```

Examples

Installing a Standard Release using TFTP

This example assumes that the router is correctly configured to allow TFTP to function. This means that IP has been configured and the router communicates with the designated TFTP server. The TFTP server is assumed to be functioning correctly and the release and patch files are assumed to be present in the server's TFTP directory. The router has no release or patch files, and is running the FLASH boot release. The IP address of the server is 172.16.1.1. The name of the release file being loaded is `8-231.rel` and the name of the patch file is `8231-01.pat`.

To install a standard release

1. **Configure the loader.**

   The Loader module is set up with defaults to make the process of downloading files in future simpler. All release and patch files in this router are stored in flash.

   ```
   set loader server=172.16.1.1 dest=flash
   ```

2. **Download the release file to the router.**

   The release file is downloaded to the router with the command:

   ```
   load file=8-231.rel
   ```

   The process of downloading a release file can take some time, even if the router and the TFTP server are connected by high speed links. An indicative time for downloading a release over Ethernet is 5 to 10 minutes. The progress of the download can be monitored with the command:

   ```
   show load
   ```

When the download has completed, the presence of the file in flash can be displayed with the command:

```
show file
```

This shows the file 8-231.rel is present.

3. **Enter the licence information for the release.**

   To allow this file to be used as a release file, a licence must be entered with the command:

   ```
   enable release=8-231.rel password=ce645398fbe number=2.3.1
   ```

   The password is provided by your authorised distributor or reseller and is unique for the release number (in this case 2.3.1), the file name and the router's serial number.

4. **Test the release.**

   The release can now be tested with the command:

   ```
   set install=temporary release=8-231.rel
   ```

   The install information can be checked with the command:

   ```
   show install
   ```

   The router is then rebooted, and the install is checked again. The display in the install history should indicate that the temporary install is loaded.

5. **Make the release the default (permanent) release.**

   If the router operates correctly with the new release, the release may be made permanent with the command:

   ```
   set install=preferred release=8-231.rel
   ```

   Every time the router reboots from now on, the new release is loaded from flash.

Installing a Standard Patch

This example illustrates how to install a standard patch on a router.

To install a standard patch

1. **Download the patch file to the router.**

   Download the patch file 8231-01.pat into the router with the command:

   ```
   load file=8231-01.pat
   ```

   This download takes a lot less time than the download of the release file, and is verified by showing the file in flash.

2. **Test the patch.**

   As with the release, the patch should first be checked by incorporating it into a temporary install with the command:

   ```
   set install=temporary release=8-231.rel patch=8231-01.pat
   ```

   The router is then rebooted and the install is checked again. The display in the install history should indicate that the temporary install is loaded.

3. **Make the patch the default (permanent) patch.**

   If the router operates correctly with the new patch, the patch may be added to the preferred install with the command:

   ```
   set install=preferred patch=8231-01.pat
   ```

   The release information is still present in the preferred install and does not have to be re-entered.

Installing a Compressed Release

This example is identical to the previous example, except that a compressed release and patch are installed.

To install a compressed release

4. **Configure the loader.**

   The Loader module is set up with defaults to make the process of downloading files in future simpler. All release and patch files in this router are stored in flash.

   ```
   set loader server=172.16.1.1 dest=flash
   ```

5. **Download the release files to the router.**

   The compressed release files are downloaded to the router with the command:

   ```
   load file=8-231.rez
   ```

   The process of downloading a release file can take some time, even if the router and the TFTP server are connected by high speed links. An indicative time for downloading a release over Ethernet is 5 to 10 minutes. The progress of the download can be monitored with the command:

   ```
   show load
   ```

   When the download has completed, the presence of the files in flash can be displayed with the command:

   ```
   show file
   ```

   This shows the file 8-231.rez is present.

6. **Enter the licence information for the release.**

   To allow these files to be used as release files, a licence must be entered for each file with the command:

   ```
   enable release=8-231.rez password=ce645398fbe number=2.3.1
   ```

   The password is provided by your authorised distributor or reseller and is unique for the release number (in this case 2.3.1), the file name and the router's serial number.

7. **Test the release.**

   The release can now be tested with the command:

   ```
   set install=temporary release=8-231.rez
   ```

   The install information can be checked with the command:

   ```
   show install
   ```

   The router is then rebooted, and the install is checked again. The display in the install history should indicate that the temporary install is loaded.

8. **Make the release the default (permanent) release.**

   If the router operates correctly with the new release, the release may be made permanent with the command:

   ```
   set install=preferred release=8-231.rez
   ```

   Every time the router reboots from now on, the new release is loaded from flash.

Installing a Compressed Patch

This example illustrates how to install a compressed patch on a router running Software Release 2.3.1.

To install a compressed patch

1. **Download the patch files to the router.**

   Download the patch file 8231-01.paz into the router with the command:

   ```
   load file=8231-01.paz
   ```

   This download takes a lot less time than the download of the release files, and is verified by showing the files in flash.

2. **Test the patch.**

   As with the release, the patch should first be checked by incorporating it into a temporary install with the command:

   ```
   set install=temporary release=8-231.rez patch=8231-01.paz
   ```

   The router is then rebooted, and the install is checked again. The display in the install history should indicate that the temporary install is loaded.

3. **Make the patch the default (permanent) patch.**

   If the router operates correctly with the new patch, the patch may be added to the preferred install with the command:

   ```
   set install=preferred release=8-231.rez patch=8231-01.paz
   ```

   The release information is still present in the preferred install and does not have to be re-entered.

Special Feature Licences

A special feature licence and password are required to activate some special features over and above the standard software release. Typically, these special features are covered by government security regulations. Special feature licences and passwords are quite separate and distinct from the standard software release licences and passwords.

A special feature licence may be either a 30-day trial licence or a full (unlimited time) licence and is specific to a router serial number. Special feature licences cannot be transferred from one router to another.

The password for a special feature licence is a string of at least 16 hexadecimal characters, and encodes the special feature or features covered by the licence, the licence type (30-day trial licence or full licence) and the router serial number. The password information is stored in flash memory.

Special feature licences are enabled with the command:

```
enable feature=featurename password=password
```

Special feature licences are disabled with the command:

```
disable feature={featurename|index}
```

A list of current special feature licences can be displayed with the command:

```
show feature[={featurename|index}]
```

Passwords must be ordered from your local authorised distributor or reseller. You must specify the special features to be licenced and the serial number(s) of the router(s) where the special feature licences are to be enabled.

Command Reference

This section describes the commands available on the router to support day-to-day operational and management activities.

The shortest valid command is denoted by capital letters in the Syntax section. See "Conventions" on page xcv of Preface for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

activate flash compaction

Syntax

```
ACTivate FLash COMPACTION
```

Description

This command activates the flash compaction process. Compaction is the process of cleaning up garbage (deleted files) by searching through flash memory copying valid files to a new block and erasing the old blocks. The compaction process is usually automatic when garbage reaches a preset limit, so manual compaction is not required during normal operation. This command can be used to recover garbage space before the automatic compaction threshold is reached.

Compaction is necessary because flash memory has a granular erase structure that requires data to be erased in large blocks rather than as individual bytes. To allow files to be mapped onto this structure, the FFS keeps track of the status of each file — whether it is being written, is complete, or is deleted. When the total amount of flash memory used for deleted files reaches a preset limit, a compaction process begins. Compaction searches through flash memory, copying good files to a new location. After the good files in an erase block have been copied, the block is cleared. This operation deletes files in the block being cleared and frees space for new files. If a large amount of flash memory is in use, compaction may take several seconds. However, flash memory operations continue unaffected by the compaction process.

While flash is compacting, do not restart the router or use commands that affect the flash file subsystem. Do not restart the router, or create, edit, load, rename, or delete files until a message confirms that flash file compaction is complete. Interrupting flash compaction stops the process. Compaction of files will then be done on the next file delete if no **load** command is issued.

While compaction is underway, the following command indicates an FFS global operation is compacting:

```
show flash
```

When compaction is complete, the global operation returns to none.

Related Commands show flash

add alias

Syntax ADD ALIas=*name* STRing=*substitution*

where:

- *name* is a character string 1 to 132 characters long. It may contain any printable character. If *name* contains spaces, it must be in double quotes. It is case-sensitive.
- *substitution* is a character string 1 to 132 characters long. It may contain any printable character. If *substitution* contains spaces, it must be in double quotes. It is case-sensitive.

Description This command adds a new alias for a longer character sequence. When the user presses [Enter] to execute the command line, the command processor first checks the command line for aliases and substitutes the replacement text. The command line is then parsed and processed normally. Alias substitution is not recursive—the command line is scanned once for aliases. An alias may represent either part of a command, or a complete command.

The **alias** parameter specifies the name of the alias. This is the text that the user enters on the command line.

The **string** parameter specifies the substitution string. When the command processor parses the command line, all occurrences of the alias are replaced by this string.

Examples To create an alias “`df`” that expands to “`delete file=1-190.rez`”, use the command:

```
add ali=df str="delete file=1-190.rez"
```

Thereafter, the following commands are equivalent:

```
df
```

```
del file=1-190.rez
```

Related Commands show alias
delete alias

add radius server

Syntax

```
ADD RADius SERVER=ipadd SECret=secret POrt=port-number
   ACCPort=port-number [LOCal={NONE|1..15}]
```

where:

- *secret* is a character string 1 to 63 characters long that is case-sensitive. It may contain uppercase and lowercase letters, digits (0–9), and the underscore character (_). If the string contains spaces, it must be in double quotes.
- *ipadd* is an IP address in dotted decimal notation.
- *port-number* is a port number from 0 to 65535.

Description This command adds a RADIUS server to the list of known RADIUS servers. RADIUS servers are used for user authentication.

The **server** parameter specifies the IP address of the RADIUS server in dotted decimal notation. The server must not already be in the list of known RADIUS servers. If **server** is specified but **port** and **accport** are not, then the RADIUS server is used for both authentication and accounting, and requests are sent to the default ports (1645 and 1646). Use the **port** and **accport** parameters to prevent the RADIUS server being used for authentication or accounting, or to specify a different port number to use.

The **secret** parameter specifies a shared secret used in communications between the router and the RADIUS server. The secret is used by the router to encrypt the password field in authentication requests sent to the RADIUS server, and by the RADIUS server to authenticate the router's request. The secret is case-sensitive.

The **port** parameter specifies a non-standard port number for communication with the RADIUS server. Setting the port number to zero means that the server is not to be used for RADIUS authentication (it may be required for RADIUS accounting).

The **accport** parameter specifies a port number for communication with the RADIUS server running RADIUS accounting (RFC 2139). Setting the port number to zero means that the server is not to be used for RADIUS accounting (it may be required for RADIUS authentication).

The **local** parameter specifies a local interface to be used as the source for all RADIUS packets the switch generates and subsequently sends to this RADIUS server. The local interface IP address will also be used as the NAS IP address in these outgoing packets. The local interface must already be configured and fall in the range 1-15. If the parameter is either not set or the option NONE is specified, the switch will select a source from the current available interfaces instead.

By default the RADIUS server uses port number 1645 to connect to RADIUS servers for authentication, and port number of 1646 for RADIUS accounting. The RADIUS accounting port is not the official port number (1813) but is the port number used by a number of commonly available packages.

Examples To add a RADIUS server with an IP address of 192.168.17.11 and "Valid8Me" as the shared secret, use the command:

```
add radius server=192.16817.11 secret=Valid8Me local=5
```

To add a RADIUS server for accounting with an IP address of 192.168.17.12 and "Valid8Me" as the shared secret, use the command:

```
add rad server=192.16817.11 sec=Valid8Me po=0 accp=1813
```

Related Commands **delete radius server**
show radius

add tacacs server

Syntax `ADD TACacs SERVER=`*`ipadd`*

where *ipadd* is an IP address in dotted decimal notation

Description This command adds a TACACS server to the list of TACACS servers used for authenticating login names.

The **server** parameter specifies the IP address of the server in dotted decimal notation. An unlimited number of TACACS servers may be defined, although two or three would be a sensible maximum number.

Examples To add a TACACS server with the IP address 172.16.8.5 use the command:

```
add tac server=172.16.8.5
```

Related Commands **delete tacacs server**
show tacacs server

add tacplus server

Syntax

```
ADD TACPlus SERVer=ipaddress [Key=key] [PORT=port]
   [SINGLEconnection={Yes|No] [TIMEOUT=1..10][LOCAL={NONE|
   1..15}]
```

where:

- *ipaddress* is an IP address in dotted decimal notation.
- *key* is a string of up to 64 characters.
- *port* is an integer value.

Description

This command adds a TACACS+ server.

The **server** parameter specifies the IP address of the TACACS+ server to identify. A network can have different TACACS+ servers for the purposes of authentication, authorization and accounting.

The **key** parameter specifies the encryption key to be used for encrypting and decrypting all traffic between the router and the TACACS+ server. It is a shared secret key between the router and the TACACS+ server. It overrides the default key, which is a global key.

The **port** parameter specifies the TCP port number to be used when making connections to the TACACS+ server. The default port number is 49.

The **timeout** parameter specifies the period of time (in seconds) that the router waits for a response from the TACACS+ server before it times out. The default is 5 seconds.

The **singleconnection** parameter specifies whether multiple TACACS+ sessions are supported on a single TCP session. If **yes** is specified, the router opens and maintains a single TCP connection for multiple TACACS+ sessions. If **no** is specified, the router opens one TCP connection for each TACACS+ session. It is more efficient for one TCP connection to support multiple TACACS+ sessions. The default is **no**.

The **local** parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS+ server. The local interface must already be configured and fall in the range 1-15. If either the parameter is not set or the option NONE is specified the switch will select a source from the current available interfaces instead.

Figure 1-11: Example output from the SHOW PIM RPCANDIDATE command...

```
PIM4 RP Candidate
---------------------------------------------------------------------------
Priority .......................... 192
Interface ......................... vlan1
       Group address/Mask ................ 224.1.1.1 / 255.255.255.255
       Group address/Mask ................ 224.2.2.0 / 255.255.255.0
```

Examples

To add a TACACS+ server to IP address 192.168.196.22, with the key *"akey4atr2supportacacsplus"*and a timeout of 3 seconds, use the command:

```
add tacp serv=192.168.196.22 K=akey4atr2supportacacsplus
   timeout=3 single=n
```

Related Commands

delete tacplus server
set tacplus server
show tacplus server

add user

Syntax

```
ADD USEr=login-name LOgin={True|False|ON|OFf|Yes|No}
    PAssword=password [CALLingnumber=number]
    [CBNUMber=e164number] [Description=description]
    [MASk=ipadd] [IPaddress=ipadd] [IPXnetwork=network]
    [NETmask=ipadd] [MTu=40..1500] [PRivilege={USer|
    MAnager|SEcurityofficer}] [TElnet={Yes|No}]
```

where:

- *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.
- *password* is a character string 1 to 32 characters long. Valid characters are any printable character. If the string contains spaces, it must be in double quotes.
- *number* is an ISDN number, 1 to 32 characters long. Valid characters are any printable characters, but the calling number it is to match is likely to contain only decimal digits. If the string contains spaces, it must be in double quotes.
- *e164number* is a valid phone number. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string 1 to 23 characters long. Valid characters are any printable character. If the string contains spaces, it must be in double quotes.
- *ipadd* is an IP address in dotted decimal notation.
- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.

Description

This command adds a user to the User Authentication Database. The **user** parameter specifies the login name for the user. It is not case sensitive.

The **login** parameter specifies whether users with a User privilege can log into the router. If **false**, the user is authenticated by the User Database but is not allowed to log into the router. If **true**, the user logs into the router and enters commands. The default is false.

The **password** parameter specifies the password for the user. The password is case sensitive. It is intended that this parameter initially set a password for the user and that the user will later change it to a private string by using the **set password** command on page 1-120. A password set with the **set password** command may contain any printing character. A configurable minimum password length is enforced. The default is 6 characters.

The **callingnumber** parameter specifies the calling number to be used to authenticate incoming calls from L2TP and ISDN services that provide caller ID information.

The **cbnumber** parameter specifies the ISDN number to use when making a call back to a remote user using the PPP callback facility.

The **description** parameter specifies a descriptive text for the entry, such as the full name and location of the user. This string may contain any printing character and the case is preserved in output.

The **ipaddress** parameter specifies an IP address for the user. The value must be a valid IP address in dotted decimal form.

The **mask** parameter specifies the address mask which extends the range of IP addresses. If the mask parameter is not present a mask of 255.255.255.255 is used. The address and mask must be internally consistent in that the result of ANDing the address and mask should be the address.

The **mtu** parameter specifies a Maximum Transmission Unit value for the user. The value must be a decimal integer from 40 to 1500 inclusive.

The **netmask** parameter and the **mask** parameter are synonymous.

The **ipaddress**, **mask** and **mtu** parameters are required if the user is to login in order to make a PPP or SLIP connection to the router over a modem connected to an asynchronous port.

The **ipxnetwork** parameter specifies the Novell network number assigned to the user accessing a Novell internetwork. See Chapter 19, Novell IPX for more information. The network number may be cleared by setting **ipxnetwork** to **none** instead of a network number. The default is none.

The **privilege** parameter specifies the privilege level for the user. The default is **user**. A user with User privilege has access to a limited subset of commands, generally commands that affect the user’s own session or asynchronous port. A user with Manager privilege has access to the complete router command set when the router is operating in normal mode, or a subset of commands when the router is operating in security mode. A user with Security Officer privilege has access to the full set of commands, and in particular, can access security commands while the router is operating in security mode.

The **telnet** parameter specifies whether the user is permitted to use the **telnet** command on page 21-31 of Chapter 21, Terminal Server to Telnet to another host, or the **connect** command on page 21-15 of Chapter 21, Terminal Server to access a Telnet service when logged in via Telnet.

Examples To add a user with the login name “bruce”, the password “sbfd4Q”, login=yes, and Manager privilege, use the command:

```
add use=bruce description="Bruce Wilson" pa=sbfd4Q pr=ma lo=y
```

To add a user with the login name “accounts”, the password “Cash4Cast”, and User privilege without access to the command line, and specify an IP address, network mask, and MTU so that the user can make SLIP connection to the router, use the command:

```
add use=accounts description="Accounting Data Entry"
   pa=Cash4Cast pr=us ipaddress=192.168.35.17
   netmask=255.255.255.0 mt=1500 lo=n
```

To add a user with the login name "cipher", password "sbr4y3", login=yes, and Security Officer privilege, use the command:

```
add user=cipher password=sbr4y3 privilege=security officer
    login=yes
```

Related Commands

delete user
disable system security_mode
disable user
enable system security_mode
enable user rso
purge user
reset user
set user
show user

add user rso

Syntax

```
ADD USEr RSO IP=ipadd [MASK=ipadd]

ADD USEr RSO IP=ipadd[-ipadd]

ADD USEr RSO IP=ipv6add[/prefix-length]

ADD USEr RSO IP=ipv6add[-ipv6add]
```

where:

- *ipadd* is an IPv4 address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address (see "IPv6 Addresses and Prefixes" on page 15-4 of Chapter 15, Internet Protocol Version 6 (IPv6)).
- *prefix-length* is an integer from 1 to 128.

Description

This command adds an IP address or address range to the list of remote access users eligible for Remote Security Officer access. The specified address or range must not already exist in the list, but it may overlap other addresses or ranges already in the list. This command can be issued only by a user with Security Officer privilege.

The **ip** parameter specifies the base IP address for this range of Remote Security Officer addresses. Base IP addresses defined with successive invocations of this command should be unique since the base IP address identifies the Remote Security Officer access entry. The **ip** parameter may be one of the following:

- an IPv4 address and optional mask
- an IPv4 address range
- an IPv6 address and optional prefix length
- an IPv6 address range

If a single IPv6 address is specified without a prefix length, the default prefix length is 128. If a range of IPv6 addresses is specified, a prefix length is not required.

The **mask** parameter specifies an address mask that extends the range of IPv4 addresses. This parameter is only valid if the **ip** parameter specifies a single base IPv4 address. The address and mask must be internally consistent such that the result of ANDing the address and mask should be the address. The default is 255.255.255.255.

Examples To add the IPv4 address 192.168.11.7 as a Remote Security Officer, use the command:

```
add user rso ip=192.168.11.7
```

To add the IPv4 address range 192.168.13.1 to 192.168.13.45 as Remote Security Officers, use the command:

```
add user rso ip=192.168.13.1-192.168.13.45
```

To add all IP addresses in the network 172.30.1.0 as Remote Security Officers, use the command:

```
add user rso ip=172.30.1.0 mask=255.255.255.0
```

To add the IPv6 address 3ffe::1:1 as a Remote Security Officer, use the command:

```
add user rso ip=3ffe::1:1
```

To add the IPv6 address range 3ffe::1:/64 as Remote Security Officers, use the command:

```
add user rso ip=3ffe::1:/64
```

To add the IPv6 address range 2ffe::1:13 to 2ffe::1:72 as Remote Security Officers, use the command:

```
add user rso ip=2ffe::1:13-2ffe::1:72
```

Related Commands

delete user rso
disable user rso
enable user rso
show user rso

clear flash totally

Syntax

`CLear FLash TOTally`

Description

This command completely clears the flash memory to an erased state. Clearing the flash memory is not required for normal operation. This command intended as a troubleshooting tool to allow the flash file system to be returned to a known state.

This command destroys all existing files and reformat the flash memory. Files cannot be salvaged after the flash memory has been erased.

Erasing flash may take up to a minute. While it is under way, the **show flash** command on page 1-150 indicates that the FFS global operation is in the “erasing” state. When the erasure is complete, a message is displayed and the global operation returns to “none”.

Related Commands

show flash

copy

Syntax

COPy *filename1.ext filename2.ext*

Where:

- *filename1* is the name of an existing file.
- *filename2* is a valid filename, between 1 and 16 characters long, that does not already exist. Valid characters are lowercase letters (a–z), uppercase letters (A–Z), digits (0–9), and the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }. Invalid characters are * + = “ | \ [ ] ; : ? / , < >.
- *ext* is a 3-letter file extension and can be any text file (for example, txt, cfg, scp, hlp, htm, spa or mds) **except** ukf or stk. The original file and the copy must have the same extension.

Description

This command copies a text file in flash memory.

Example

To copy the file `admin.cfg` to the file `admin2.cfg`, use the command:

```
cop admin.cfg admin2.cfg
```

Related Commands

delete file
show file

create config

Syntax

```
CREate CONfig=filename
```

where *filename* is a file name in the format `device:filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

The *device* variable is optional and specifies the physical memory device where the file is stored, which is flash. If *device* is specified, it must be separated from the rest of the file name by a colon (:). If *device* is not specified, the default is flash.

Description

This command creates a script file containing the commands required to recreate the current dynamic configuration of the router. This command can be issued only by a user with Security Officer privilege.

The **config** parameter specifies the name of the script or configuration file to create. The file extension must be “`scp`” or “`cfg`”. If the file already exists, it is replaced. If the file does not exist, it is created.

The **create config** command on page 1-70 writes the MD5 digest, not the plaintext, of passwords in commands to the configuration file. When a configuration script is executed the command processor can determine whether the password value is plaintext or an MD5 digest.

The configuration of a specific software module cannot be saved with this command. To save the configuration of a specific software module, use the **show config** command on page 1-133 to display the configuration, capture the output and save it to a file.

Examples

To save the current dynamic configuration as the default boot script boot.cfg, use the command:

```
cre con=boot.cfg
```

Related Commands

restart
set config
show config

delete alias

Syntax

```
DELete ALIas=name
```

where *name* is a character string 1 to 132 characters long. It may contain any printable character. If *name* contains spaces, it must be in double quotes. It is case-sensitive.

Description

This command deletes an existing alias. Occurrences of the alias string in the command line are no longer expanded to the substitution text.

The **alias** parameter specifies the name of the alias to be deleted.

Example

To delete an alias with name “ii”, use the command:

```
del ali=ii
```

Related Commands

add alias
show alias

delete file

Syntax

```
DELete FIle=filename
```

where *filename* is a file identifier in the format `[device:]name.ext`. Optionally, `device` specifies the physical memory device where the file is stored, which is flash. Invalid characters are “ \ ; ? / , <. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { } * > [ ] | :

Wildcard characters * may appear anywhere in the filename. The wildcard character matches any string.

Character ranges may be specified using the > character, for example a>z matches any letter in the alphabet. The + character may be used to specify a list of options, for example a*.scp+b*.scp would specify files that match a*.scp or b*.scp.

Square brackets may be used, for example ppp*.[scp+cfg] matches scripts and configuration files whose names start with “ppp”.

The vertical bar | character matches any single character. For example, |||.scp matches script files with names three characters long (excluding extension and device name).

If a colon is seen anywhere in the filename, it is assumed that the filename includes the device name. Otherwise, it is assumed that the file is stored in flash.

Description

This command deletes the specified file or files. This command can be issued only by a user with Security Officer privilege.

The GUI resource file that the router is currently set to use can be deleted when the GUI is disabled. GUI resource files have an RSC extension. Use the **show install** command and check the "Current Install" section to see which resource file is currently set. See the **disable gui** command on page 1-78 for more information about disabling the GUI.

Caution must be taken when deleting files, such as patches, releases, licences and configurations, since they contain information which is vital to the intended operation of the router.

Examples To delete all the patch files on the router, use the command:

```
delete file=*:*.pat
```

To delete the release file 28-72.REL, use the command:

```
del fi=28-72.rel
```

Related Commands

rename
show file

delete install

Syntax

```
DELete INSTall={TEMPorary|PREFerred|DEFault}
```

Description This command deletes a specific install from the install information. In the case of the default install, patch information is deleted because the release information must always be left intact in the default install.

The Install module maintains install information and loads the correct install at boot. An *install* is a record identifying a release and an optional patch. The Install module has three types of installs: temporary, preferred, and default.

The default install is the install of last resort. The release for the default install cannot be changed by the manager and is always the EPROM release. The patch for the default install may be set by the manager.

The temporary and preferred installs are completely configurable. Both the release and an associated patch may be set. The release may be EPROM or one stored in FFS.

Examples To delete the temporary install, use the command:

```
del inst=temp
```

Related Commands

set install
show install

delete mail

Syntax `DELete MAIL=`*`id`*

where *id* is a hexadecimal number from 0x0 to 0xffff

Description This command deletes a specific mail message from the transmission queue.

The **mail** parameter specifies the message id of the mail message to be deleted. The message id can be determined from the output of the **show mail** command on page 1-163.

Examples To delete the mail message with a message id of 0x231b, use the command:

```
del mail=231b
```

Related Commands **mail**
show mail

delete radius server

Syntax `DELete RADius SERVer=`*`ipadd`*

where *ipadd* is an IP address in dotted decimal notation

Description This command deletes a RADIUS server from the list of known RADIUS servers. RADIUS servers are used for user authentication.

The **server** parameter specifies the IP address of the RADIUS server, in dotted decimal notation. The server must be in the list of known RADIUS servers.

Examples To delete the RADIUS server with the IP address of 192.168.17.11, use the command:

```
del rad serv=192.168.17.11
```

Related Commands **add radius server**
show radius

delete tacacs server

Syntax

```
DELete TACacs SERVer=ipadd
```

where *ipadd* is an IP address in dotted decimal notation

Description

This command deletes a TACACS server from the list of TACACS servers used for authenticating login names. The **server** parameter specifies the IP address of the server in dotted decimal notation.

Examples

To delete the TACACS server with the IP address 172.16.8.5 use the command:

```
del tac serv=172.16.8.5
```

Related Commands

add tacacs server
show tacacs server

delete tacplus server

Syntax

```
DELete TACPlus SERVer=ipaddress
```

where *ipaddress* is an IP address in dotted decimal notation

Description

This command deletes a TACACS+ server.

The **server** parameter specifies the IP address of the TACACS+ server, which must already be defined using the **add tacplus server** command.

Example

To delete the TACACS+ server with IP address 192.168.196.22, use the command:

```
del tacp serv=192.168.196.22
```

Related Commands

add tacplus server
set tacplus server
show tacplus server

delete user

Syntax `DELete USEr=login-name`

where *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.

Description This command deletes a user from the User Authentication Database. The **user** parameter specifies the login name for the user. It is case insensitive.

If the router is operating in security mode, you cannot delete every user with Security Officer privilege. At least one user with Security Officer privilege must exist in the User Authentication Database for the router to operate in security mode.

Related Commands

add user
disable user
enable user rso
purge user
reset user
set user
show user

delete user rso

Syntax

```
DELete USEr RSO IP=ipadd[-ipadd]
```

```
DELete USEr RSO IP=ipv6add/prefix-length
```

```
DELete USEr RSO IP=ipv6add[-ipv6add]
```

where:

- *ipadd* is an IPv4 address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address in slash notation (see "IPv6 Addresses and Prefixes" on page 15-4 of Chapter 15, Internet Protocol Version 6 (IPv6)).
- *prefix-length* is an integer from 1 to 128.

Description

This command deletes an IP address or address range from the list of remote access users eligible for Remote Security Officer access. The specified address or range must already exist in the list. Remote Security Officers who currently have Security Officer privilege lose it immediately. This command can be issued only by a user with Security Officer privilege.

The **ip** parameter specifies the base IP address for this range of Remote Security Officer addresses. It must match exactly an entry in the list of remote access users. Other overlapping but non-identical entries in the list are not affected. The **ip** parameter may be one of the following:

- an IPv4 address
- an IPv4 address range
- an IPv6 address and prefix length
- an IPv6 address range

If a single IPv6 address is specified, the prefix length must also be specified. If a range of IPv6 addresses is specified, a prefix length is not required.

Examples

To delete the IPv4 address 192.168.11.7 from the list of Remote Security Officers, use the command:

```
del user rso ip=192.168.11.7
```

To delete all IP addresses in the network 172.30.1.0 from the list of Remote Security Officers, use the command:

```
delete user rso ip=172.30.1.0 mask=255.255.255.0
```

To delete the IPv4 address range 192.168.13.1 to 192.168.13.45 from the list of Remote Security Officers, use the command:

```
del user rso ip=192.168.13.1-192.168.13.45
```

To delete the IPv6 address 3ffe::1:/64 from the list of Remote Security Officers, use the command:

```
del user rso ip=3ffe:1:/64
```

To delete the IPv6 address range 2ffe::1:13 to 2ffe::1:72 from the list of Remote Security Officers, use the command:

```
del user rso ip=2ffe::1:13-2ffe::1:72
```

Related Commands

add user rso
disable user rso
enable user rso
show user rso

disable feature

Syntax

```
DISable FEAture={featurename|index}
```

where:

- *featurename* is a character string 1 to 12 characters long. Valid characters are any printable character.
- *index* is a decimal number in the range 1 to the number of special feature licences.

Description

This command disables the specified special feature licence. The **feature** parameter specifies either the name assigned to the special feature when it was enabled with the **enable feature** command on page 1-88, or the index number of the special feature as in the output of the **show feature** command on page 1-144. The special feature must exist on the router and currently be enabled.

This command can be issued only by a user with Security Officer privilege.

Examples

To disable the special feature licence “Triple DES”, use the command:

```
dis fea="triple des"
```

To disable the special feature licence with index 2, use the command:

```
dis fea=2
```

Related Commands

enable feature
show feature

disable gui

Syntax

```
DISable GUI
```

Description

This command disables the web-based graphical user interface. If a GUI is installed, it is enabled by default.

The GUI resource file that the router is currently set to use can be deleted when the GUI is disabled. GUI resource files have an RSC extension. Use the **show install** command and check the "Current Install" section to see which resource file is currently set.

Related Commands

enable gui
reset gui
show gui

disable http debug

Syntax

```
DISable HTTP DEBug={ALL|AUTH|MSG|SESSion|STATe}
```

Description

This command disables some or all HTTP server debugging. Debug output is sent to the terminal session or Telnet connection from which the command was entered.

The **debug** parameter specifies the type of debugging to be disabled. If **all** is specified, all debugging is disabled. Debugging is disabled by default.

For **auth**, debugging of authentication attempts is disabled.

For **msg**, debugging is disabled for HTTP "get" and "set" requests and responses.

For **session**, debugging is disabled for TCP state changes and session activity.

For **state**, debugging is disabled for state changes in the state machine. The **state** debug shows each event that occurs, the current state and the new state.

Examples

To disable HTTP server debugging, use the command:

```
dis http deb
```

Related Commands

enable http debug
show http debug

disable http server

Syntax `DISable HTTP SERVer`

Description This command disables the HTTP server. The HTTP server serves HTML pages out of the router's flash memory to a web browser, and allows users to login into the router. The server is enabled by default.

Examples To disable the HTTP server, use the command:

```
dis http serv
```

Related Commands

enable http server
reset http server
show http server
show http server session

disable ldap debug

Syntax `DISable LDAP DEBug`

Description This command disables LDAP debugging. By default, debugging is disabled.

Examples To turn LDAP module debugging off, use the command:

```
dis ldap deb
```

Related Commands

enable ldap debug
show ldap

disable mail debug

Syntax `DISable MAIL DEBug`

Description This command disables the display of debugging information for mail. By default, debugging is disabled.

Examples To disable the display of debugging information for mail, use the command:

```
dis mail deb
```

Related Commands

enable mail debug
show mail

disable radius debug

Syntax

```
DISable RADius DEBug={ALL|PKT|DECODE|ERROR} [,...]
```

Description

This command disables the debugging option for all RADIUS servers.

Examples

To disable the debugging of raw packets sent to and received from all RADIUS servers, use the command:

```
dis rad deb=pkt
```

Related Commands

enable radius debug
show radius debug

disable release

Syntax

```
DISable RELease=release-name
```

where *release-name* is the name of a release file, in the format `[device:]filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

Description

This command removes the licence for the specified release file.

The **release** parameter specifies the name of the release file. If a device is not specified, the default is flash.

Examples

To disable release `28-761.rel`, use the command:

```
dis rel=28-761.rel
```

Related Commands

enable release
show release

disable system security_mode

Syntax

```
DISable SYStem SECurity_mode
```

Description

This command disables security mode on the router. When the router is operating in security mode, a subset of router commands, called the security commands, require Security Officer privilege to execute. Sensitive data files such as encryption key files can be stored in the router's file subsystem when the router is in security mode.

Disabling security mode deletes sensitive data files, such as encryption keys, from the router's file subsystem.

Security mode should be enabled on any router that is fitted with a hardware encryption device or is configured to provide secure features like encryption, authentication or Secure Shell.

Examples

To disable security mode, use the command:

```
dis sys sec
```

Related Commands

add user
enable system security_mode
set user
show system
show user

disable tacacs debug

Syntax

```
DISable TACacs DEBug={ALL|PKT|DECode|ERRor} [,...]
```

Description

This command disables the debugging option for all TACACS servers.

Examples

To disable the debugging of raw packets sent to and received from all TACACS servers, use the command:

```
dis tac deb=pkt
```

Related Commands

enable tacacs debug
show tacacs debug

disable tacplus

Syntax

```
DISable TACPlus
```

Description This command disables TACACS+ operation on the router.

Example To disable TACACS+, use the command:

```
dis tacp
```

Related Commands **enable tacplus**

disable tacplus debug

Syntax

```
DISable TACPlus DEBug
```

Description This command disables debugging for all TACPLUS servers.

Examples To disable the debugging of all TACACS+ servers, use the command:

```
disa tacp deb
```

Related Commands **enable tacplus debug**

disable user

Syntax

```
DISable USEr=login-name
```

where *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.

Description This command temporarily disables a user login name in the User Authentication Database. The login name must be currently enabled. Logins attempts through the User Authentication Database using the login name are ignored. This command has no effect on user authentication through TACACS+, TACACS, or RADIUS servers.

The **user** parameter specifies the login name for the user. It is case insensitive.

Related Commands

add user
delete user
enable user rso
purge user
reset user
set user
show user

disable user rso

Syntax

```
DISable USEr RSO
```

Description

This command disables Remote Security Officer access. Remote Security Officers who have Security Officer privilege immediately lose it. This command can be issued only by a user with Security Officer privilege.

Examples

To disable Remote Security Officer access, use the command:

```
dis use rso
```

Related Commands

add user rso
delete user rso
enable user rso
show user rso

dump

Syntax

```
DUMP [ADDR=address] [LEN=length] [SIZE={BYTE|LONG|WORD}]
    [SPace={SD|SP|UD|UP|UR}]
```

where:

- *address* is the first address (in hexadecimal) to be dumped.
- *length* is the number of bytes (in hexadecimal) to dump.

Description

This command displays the contents of the router's memory. This command can be issued only by a user with Security Officer privilege.

The block of memory to be displayed is specified by the **addr**, **len**, and **space** parameters. The **space** parameter specifies the CPU address space to be dumped (Table 1-11)

Table 1-11: Router CPU address spaces .

SPACE value	CPU address space
UD	User Data
UP	User Program
UR	User Reserved
SD	Supervisor Data
SP	Supervisor Program

The **size** parameter specifies whether the data should be displayed grouped as BYTEs, LONGWORDs or WORDs. Note that LEN is always in bytes, regardless of the value of **size**.

If the **len**, **size**, or **space** parameters are omitted then they default to the value they had at the previous invocation of the command. If the **addr** parameter is omitted, it increments to dump the block of memory immediately following the block dumped by the previous invocation. If the **addr** parameter is given without a value, then it dumps the block of memory previously dumped.

It is possible to use this command to dump I/O devices. This may interrupt the operation of the router. The **dump** command is provided mainly as a diagnostic tool. It should not be needed for normal operation of the router.

A typical display is shown in Figure 1-12 on page 1-84. The left-hand column shows the address of the data in each row. The next eight columns give the data starting at the address for the next 16 bytes. The right-most column is an ASCII representation of the data in the row, with non-printing characters represented by a dot.

Figure 1-12: Example output from the **dump** command.

```
00000000  0001 667c 0001 667c 0000 b424 0001 667c             ..f|..f|...$..f|
00000010  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000020  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000030  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000040  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000050  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000060  0001 66d4 0001 6b14 0001 667c 0001 667c             ..f|..f...k...f|
00000070  0001 667c 0001 1308 0001 6aa4 0001 66c8             ..f|......j...f.
00000080  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
00000090  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000a0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000b0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000c0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000d0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000e0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
000000f0  0001 667c 0001 667c 0001 667c 0001 667c             ..f|..f|..f|..f|
```

Examples The command used to produce the output shown above was:

```
dump addr=0 len=100 size=word spa=sd
```

Related Commands modify

edit

Syntax `EDit [filename]`

where:

- *filename* is a filename in the format `[device]:filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:
  - uppercase and lowercase letters
  - digits (0–9)
  - the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

  Optionally, *device* specifies the physical memory device where the file is stored, which is flash. If *device* is specified, it must be separated from the rest of the file name by a colon (:). The file extension *ext* is any valid file type that contains text such as `.CFG`, `.SCP` and `.TXT`.

Description This command invokes the router’s built-in full-screen text editor to edit an ASCII text file. If a filename is specified, then the editor loads the file if it exists on the system. This command can be issued only by a user with Security Officer privilege.

The editor uses VT100 command sequences (Table 1-12 on page 1-85) and should be used with a VT100-compatible terminal, terminal emulation program, or Telnet client.

Table 1-12: Editor functions and keystrokes.

Cursor Movement		**Deletion**	
↑ or Ctrl/Z	Up one line	Ctrl/T	Delete word right
↓ or Ctrl/X	Down one line	Ctrl/Y	Delete line
→	Right one character		
←	Left one character	**Block Operations**	
Ctrl/B	Start of file	Ctrl/K,B	Begin block mark
Ctrl/D	End of file	Ctrl/K,D	Unmark block
Ctrl/A	Start of line	Ctrl/K,U	Cut block to buffer
Ctrl/E	End of line	Ctrl/K,C	Copy block to buffer
Ctrl/U	Up one screen	Ctrl/K,V	Paste block from buffer
Ctrl/V	Down one screen	Ctrl/K,Y	Delete block
Ctrl/F	Word right		

Search		**Exit**	
Ctrl/K,F	Find text	Ctrl/K,X	Exit editor; save file
Ctrl/L	Repeat last find	Ctrl/C	Quit editor; don’t save file

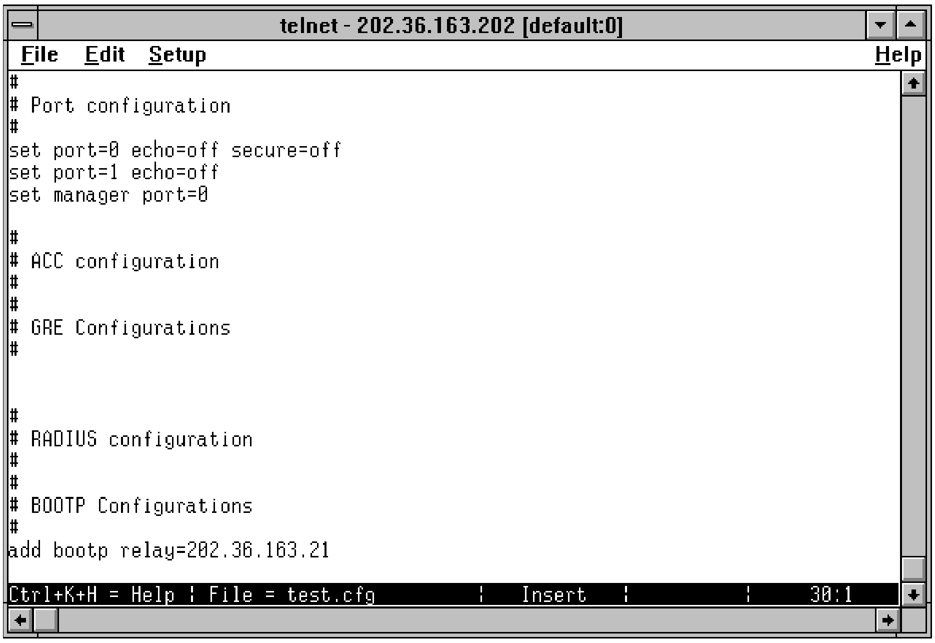
Miscellaneous			
Ctrl/I	Insert mode	Ctrl/O	Overstrike mode

Table 1-12: Editor functions and keystrokes. (continued)

Ctrl/W	Refresh the screen	Ctrl/K,H	Display help screen
Ctrl/K,O	Open a file		

The VT100 screen supports 24 lines, unlike a PC. Lines 1–23 display the text of the file being edited, and line 24 is the status bar and command line (Figure 1-13 on page 1-86). The status bar displays the current file name, line, and column position in the file and the editing mode (overstrike or insert). When additional command information is required, such as a file name or search text, a prompt is in the status bar.

Figure 1-13: The editor screen layout.



```
#
# Port configuration
#
set port=0 echo=off secure=off
set port=1 echo=off
set manager port=0

#
# ACC configuration
#
#
# GRE Configurations
#



#
# RADIUS configuration
#
#
# BOOTP Configurations
#
add bootp relay=202.36.163.21

Ctrl+K+H = Help ¦ File = test.cfg      ¦   Insert   ¦          ¦    30:1
```

The editor is invoked with the command:

```
edit [filename]
```

The file name is optional as a file can be loaded, or a new file can be created from within the editor itself. The editor is currently limited to editing one file at a time. To overcome this limitation use the cut and paste facility to transfer text between files.

Before starting the editor make sure your terminal, terminal emulation program or Telnet client is 100% compatible with a VT100 terminal.

Help can be obtained at any time while in the editor by pressing [Ctrl/K,H]; that is, holding down the Ctrl key and pressing in turn the K key then the H key.

Examples To edit a file called "flash:show.scp", use the command:

```
ed flash:show.scp
```

Related Commands

delete file
load
show file

enable

Syntax `ENAble`

Description This command sets the privilege level of a user to the level stored on a TACACS+ server, for a user whose login has been authenticated by a token card server via the TACACS+ server. Through the TACACS+ server, this command enables token card authorisation to result in login at Manager or Security Officer privilege level. The required privilege level must be configured on the TACACS+ server, using the TACACS+ **priv-lvl** value (Table 1-6 on page 1-32).

Before entering this command, the user has User privileges. When the user enters this command, the router queries the TACACS+ server, which returns the **priv-lvl** value that matches this username. The user is then logged into the router with the privilege level indicated by the **priv-lvl** value.

Example After authentication by the token card server, to log on at the privilege level that has been configured on the TACACS+ server, use the command:

```
ena
```

Related Commands **add tacplus server**
show tacplus server

enable feature

Syntax

```
ENAble FEAture=featurename PASSword=password
```

where:

- *featurename* is a character string 1 to 12 characters long. Valid characters are any printable character.
- *password* is a character string at least 16 characters long. Valid characters are hexadecimal characters (0–9, a–f, A–F).

Description

This command enables the special feature licence identified by the special feature licence name and password. This command can be issued only by a user with Security Officer privilege.

The **feature** parameter specifies a user-defined name for the special feature licence that appears in the output of the **show feature** command on page 1-144 and is used to identify the special feature licence in other commands.

The **password** parameter specifies the password for the special feature licence. The password identifies the special feature(s) being licenced, the licence type (30-day trial licence or full licence) and the router serial number. The password information is stored in the router's flash memory.

Examples

To enable the special feature licence "Triple DES" with password "591a9d5d9b2e8969cbf7", use the command:

```
ena fea="3des" pass="591a9d5d9b2e8969cbf7"
```

Related Commands

disable feature
show feature

enable gui

Syntax

```
ENAble GUI
```

Description

This command enables the web-based graphical user interface. If a GUI is installed, it is enabled by default. Even when enabled, the GUI works when:

- there is a valid resource file for the hardware model being used.
- the HTTP server is enabled.

The GUI resource file that the router is currently set to use can be deleted when the GUI is disabled. GUI resource files have an RSC extension. Use the **show install** command and check the "Current Install" section to see which resource file is currently set

Related Commands

disable gui
reset gui
show gui

enable http debug

Syntax

```
ENAble HTTP DEBug={ALL|AUTH|MSG|SESSion|STATe}
```

Description

This command enables HTTP server debugging. Debug output is sent to the terminal session or Telnet connection from which the command was entered. To enable combinations of debugging options, enter multiple commands. Debugging is disabled by default.

The **debug** parameter specifies the type of debugging to be enabled.

For **all**, all debugging is enabled.

For **auth**, debugging of authentication attempts is enabled.

For **msg**, debugging is enabled for HTTP "get" and "set" requests and responses.

For **session**, debugging is enabled for TCP state changes and session activity.

For **state**, debugging is enabled for state changes in the state machine. The **state** debug shows each event that occurs, the current state and the new state.

Examples

To enable debugging of authentication attempts and HTTP "get" and "set" messages, use the commands:

```
ena http deb=auth

ena http deb=msg
```

Related Commands

disable http debug
show http debug

enable http server

Syntax

```
ENAble HTTP SERVer
```

Description

This command enables the HTTP server. The HTTP server serves HTML pages out of the router’s flash memory to a web browser, and allows users to login into the router. The server is enabled by default.

Examples

To enable the HTTP server, use the command:

```
ena http serv
```

Related Commands

disable http server
reset http server
set http server
show http server
show http server session

enable ldap debug

Syntax

```
ENAble LDAP DEBug
```

Description

This command enables LDAP trace debugging, which allows the user to “trace” the execution of LDAP requests by displaying step by step information. By default, debugging is disabled.

Examples

To turn on LDAP trace debugging, use the command:

```
ena ldap deb
```

Related Commands

disable ldap debug
show ldap

enable mail debug

Syntax

```
ENAble MAIL DEBug
```

Description

This command enables the display of debugging information for mail. When debugging is enabled, messages recording the progress of email messages are displayed to the terminal from which the command was entered. By default debugging is disabled.

Examples

To enable mail debug, use the command:

```
ena mail deb
```

Related Commands

disable mail debug
show mail

enable radius debug

Syntax

```
ENAble RADius DEBug={ALL|PKT|DECODE|ERROR} [,...]
```

Description

This command enables the debugging option for all RADIUS servers.

The **debug** parameter specifies which debugging options are to be enabled. The value may be a single option or a comma-separated list of options.

If **all** is specified, all debugging options are enabled.

If **pkt** is specified, the raw RADIUS packets are debugged.

If **decode** is specified, decoded packets are debugged.

If **error** is specified, error messages regarding RADIUS transactions are displayed.

Examples

To enable the debugging of raw packets sent to and received from all RADIUS servers, use the command:

```
ena rad deb=pkt
```

To enable the debugging of all decoded packets and error messages for all RADIUS servers, use the command:

```
ena rad deb=decode,error
```

Related Commands

disable radius debug
show radius debug

enable release

Syntax

```
ENAble RELease=release-name [PASSword=password]
   NUMber=release-number
```

where:

- *release-name* is the name of a release file, in the format `device:]filename.ext`. Invalid characters are * + = " | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:
  - uppercase and lowercase letters (A–Z and a–z)
  - digits (0–9)
  - the characters ~ ' ! @ # $ % ^ & ( ) _ - { }
- *password* is the password to licence this release, expressed as a string of hexadecimal characters (A–F, 0–9). It is not case sensitive.
- *release-number* is the release number for this release.

Description

This command enables a release file in the router.

The **release** parameter specifies the name of the release file. If a device is not specified, the default is `flash`.

The **password** parameter specifies the password for this release, encoded as a sequence of hexadecimal digits. The password is supplied by your authorised distributor or reseller, and is specific to a particular release and router serial number. The password enables the release with either a full licence or a 30-day licence.

If the **password** parameter is not present, the router looks for a reason to be able to generate a password for this release. Valid reasons include the router EPROMs having the same major and minor version numbers as the release being licenced, or a valid release licence being found with the same major and minor version numbers as the release being licenced. If either of these reasons is found the router generates a password internally, otherwise the command does not complete. If the EPROMs or a valid full release licence are found to provide the reason for generating a release licence, a full licence is generated. If the only valid release licences found are 30 day trial licences, a 30-day trial licence is generated.

The **number** parameter specifies the software release for the release file being licenced. This is entered in dotted decimal form, for example "2.3.1".

Examples

To enable release 28-231.rel with the password CE645398FBE for software release 2.3.1, use the command:

```
ena rel=28-231.rel pass=ce645398fbe num=2.3.1
```

Related Commands

disable release
show release

enable system security_mode

Syntax

```
ENAble SYStem SECurity_mode
```

Description

This command enables security mode on the router. When the router is operating in security mode, a subset of router commands, called the security commands, require Security Officer privilege to execute. Sensitive data files, such as encryption key files, can be stored in the router's file subsystem when the router is in security mode.

Security mode should be enabled on any router that is fitted with a hardware encryption device or is configured to provide secure features like encryption, authentication or Secure Shell.

If the router is operating in security mode, Security Officer privilege is required to execute many commands. Security mode cannot be enabled unless at least one user with Security Officer privilege exists in the User Authentication Database.

Examples

To enable security mode, use the command:

```
ena sys sec
```

Related Commands

add user
disable system security_mode
set user
show system
show user

enable tacacs debug

Syntax

```
ENAble TACacs DEBug={ALL|PKT|DECode|ERRor} [,...]
```

Description This command enables the debugging option for all TACACS servers.

The **debug** parameter specifies which debugging options to enable. The value may be a single option or a comma-separated list of options.

If **all** is specified, all debugging options are enabled.

If **pkt** is specified, raw TACACS packets are debugged.

If **decode** is specified, decoded packets are debugged.

If **error** is specified, error messages regarding TACACS transactions are displayed.

Examples To enable the debugging of raw packets sent to and received from all TACACS servers, use the command:

```
ena tac deb=pkt
```

To enable the debugging of all decoded packets and error messages for all TACACS servers, use the command:

```
ena tac deb=dec,err
```

Related Commands disable tacacs debug
show tacacs debug

enable tacplus

Syntax

```
ENAble TACPlus
```

Description This command enables TACACS+ operation on the router. TACACS+ is enabled by default.

Example To enable TACACS+, use the command:

```
ena tacp
```

Related Commands disable tacplus

enable tacplus debug

Syntax

```
ENAble TACPlus DEBug
```

Description This command enables debugging for all TACACS+ servers.

Examples To enable the debugging of all TACACS+ servers, use the command:

```
ena tacp deb
```

Related Commands **disable tacplus debug**

enable user rso

Syntax

```
ENAble USEr RSO
```

Description This command enables Remote Security Officer access. Authorised IP addresses must be added with the **add user rso** command on page 1-67 before Remote Security Officer access can be used. This command can be issued only by a user with Security Officer privilege.

Examples To enable Remote Security Officer access, use the command:

```
ena use rso
```

Related Commands

add user rso
delete user rso
disable user rso
show user rso

help

Syntax

```
HELP [topic]
```

where *topic* is information to display

Description This command displays online help for commands. If a topic is not specified, a list of available topics is displayed. If a topic is specified and is available, a list of commands relating to the topic is displayed.

The system help file must be assigned using the **set help** command on page 1-111.

Examples To get help on IP, use the command:

```
help ip
```

Related Commands

set help
show system

load

Syntax

```
LOAd [METhod=TFtp] [DELay=delay] [DESTFile=destfilename]
    [DEStination={BOOTblock|FLash}]
    [{FIle|SRCFile}=filename] [SErver={hostname|ipadd|
    ipv6add}]
```

```
LOAd [METhod={HTTP|WEB|WWW}] [DELay=delay]
    [DESTFile=destfilename] [DEStination={BOOTblock|FLash}]
    [{FIle|SRCFile}=filename] [HTTPproxy={hostname|ipadd}
    [PASSword=password] [PROxyport=1..65535]]
    [SErver={hostname|ipadd|ipv6add}] [SERVPort={1..65535|
    DEFault}] [USERName=username]
```

```
LOAd [METhod=LDAP] [ATTribute={CERT|CRL|CACERT}]
    [BASeobject=dist-name] [DELay=delay]
    [DESTFile=destfilename] [DEStination={BOOTblock|FLash}]
    [PASSword=password] [SErver={hostname|ipadd}]
    [SERVPort={1..65535|DEFault}] [USERName=username]
```

```
LOAd [METhod=ZModem] [ASYn=port] [DELay=delay]
    [DESTFile=destfilename] [DEStination={BOOTblock|FLash}]
    [{FIle|SRCFile}=filename]
```

```
LOAd [METhod=NONE] [ASYn=port] [DELay=delay]
    [DESTFile=destfilename] [DEStination={BOOTblock|FLash}]
    [{FIle|SRCFile}=filename]
```

where:

- *delay* is a time delay, in seconds.
- *hostname* is a character string 1 to 40 characters long.
- *ipadd* is an IP address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address.
- *filename* is a character string 1 to 256 characters long. This is a full path name for the file to load, in the syntax of the server from which the file is loaded.
- *destfilename* is a character string 5 to 20 characters long, specifying the name of the destination file in the router file system.
- *dist-name* is an X.500 distinguished name, as described in "Distinguished Names (DN)" on page 1-50.
- *password* is a character string 1 to 60 characters long, used for basic server authentication.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with asyn0.
- *username* is a character string 1 to 60 characters long, used for basic server authentication.

Description

This command downloads a file to the router using Trivial File Transfer Protocol (TFTP), HyperText Transfer Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), ZMODEM or direct input from an asynchronous port. Any parameters not specified use the defaults set with the **set loader** command on page 1-114. Some parameters are invalid or have different meanings depending on the method used to download the file. This command can be issued only by a user has Security Officer privilege.

While flash is compacting, do not restart the router or use commands that affect the flash file subsystem. Do not restart the router, or create, edit, load, rename, or delete files until a message confirms that flash file compaction is complete. Interrupting flash compaction stops the process. Compaction of files will then be done on the next file delete if there is no **load** command issued.

The **attribute** parameter is a keyword specifying the type of object to retrieve from an LDAP repository. A list of currently recognised keywords and their respective object types can be found in the following table.

Table 1-13: Keywords recognised by the **attribute** command, and their object types.

Keyword	Object type
CERT	userCertificate
CRL	certificateReservationList
CACERT	cACertificate

The **baseobject** parameter specifies the repository location of the object to load, in the LDAP distinguished name format, and is required if the load method is LDAP. If the string contains spaces, it must be in double quotes. The special characters {, = + < > # ; \ <CR> } must be escaped by typing a \ before the character, as defined in [RFC 1779], unless they are used for their prescribed purpose. For example, to include a \ in the string, type \\ and to include a #, type \#.

The **delay** parameter specifies the delay in seconds between initiating the file download and the download actually starting. This feature allows reconfiguration of ports and devices after initiating the download. For example, a manager may be at a remote site with a single PC that is to act as both the access device to the router and the TFTP server. By specifying a delay, the manager has time to reconfigure the PC from terminal emulation mode to TFTP server mode before the download starts. The **delay** parameter is optional.

The **destfile** parameter specifies the name of the destination file in the router file system. When method is set to LDAP, the extension of the destination file must be valid for the type of object being loaded (either `"cer"`, `"crl"`, or `"csr"`). When using the HTTP method and a **destfile** is necessary, it must be present on the command line when the **file** or **srcfile** parameter is present or it has no effect.

The **destination** parameter specifies where the file is to be stored. If **bootblock** is specified, the file is stored in the special boot code area of flash reserved for the router boot code. Only boot code release files (with an FBR extension) may be loaded to the boot code area. If **flash** is specified, the file is stored in the Flash File System (FFS) on the router. If **destination** is not specified, and has not been set with the **set loader** command on page 1-114, the default is flash.

The boot code should not normally need to be upgraded. While loading a new router boot code file onto the flash boot code area, the router must not lose power. When the router goes through a power cycle while writing to the bootblock, the code used to reboot the router will be incomplete, and the router cannot be rebooted.
The router does not respond to any interfaces while the boot block is being written. The router should be idle while the boot block is being reloaded. The router must have sufficient free buffers (about 600) when starting the download in order to store the entire boot code.

The **httpproxy** parameter specifies the proxy server used to handle HTTP requests. Either the IP address or the fully qualified domain name of the proxy server may be specified. If a domain name is specified, the router performs a DNS lookup to resolve the name.

The **method** parameter specifies the method to use when downloading the file. If **http** is specified, HTTP is used to download the file. The options WEB and WWW are synonyms for HTTP. If LDAP is specified, LDAP is used to download the file. If **tftp** is specified, TFTP is used to download the file. If **zmodem** is specified, the ZMODEM protocol is used to download the file. If **zmodem** is specified, the **asyn** parameter is required unless it has been set with the **set loader** command on page 1-114. If **none** is specified, text files can be downloaded and all input received via the port is directed to the specified file on the router's file subsystem. The file transfer is terminated by the first control character received that is not a CR or LF character. The FILE parameter is not used when **method** is set to **zmodem** or **ldap**. The **asyn** parameter is not valid when **method** is set to **http**, **web**, **www**, **ldap**, or **tftp**. If the **method** parameter is set to **cflash**, the file indicated by the **file** parameter is loaded from CompactFlash to the destination device. This command converts Motorola S-Record files to binary files. The default is **tftp**.

The **password** parameter specifies the password for the LDAP or HTTP methods if server authentication is required.

The **asyn** parameter specifies the asynchronous port via which the file is to be downloaded when the **method** parameter is set to **zmodem** or **none**. If **method** is set to **zmodem** or **none**, the **asyn** parameter is required unless it has been set with the **set loader** command on page 1-114.

The **proxyport** parameter specifies the port on a proxy server. The **proxyport** parameter is valid if **method** is **http** and **httpproxy** is specified. The default is 80.

The **server** parameter specifies the IP address or the hostname (a fully qualified domain name) of the HTTP, LDAP or TFTP server from which the file is loaded. If a host name is specified, a DNS lookup is used to translate this to an IP address. See **set ip nameserver** command on page 14-151 of Chapter 14, Internet Protocol (IP) for more information about setting up name servers. The **ping** command on page 14-129 of Chapter 14, Internet Protocol (IP) can be used to verify that the router can communicate with the server via IP. The **server** parameter is required if **method** is **http**, **ldap** or **tftp** unless it has been set by the **set loader** command on page 1-114. The **server** parameter is not valid when **method** is set to **zmodem** or **none**. The following are examples of valid server names for method HTTP or LDAP:

```
host.company.com

192.168.3.4
```

The **servport** parameter optionally specifies the port on the HTTP or LDAP server from which the file is loaded. If this is not specified (or specified using the **default** keyword) and no default has been set using the **set loader** command, a default is invoked according to the current load method. In this case, **servport** takes a value of 80 for **http**, and 389 for **ldap**.

The **srcfile** or **file** parameter specifies the name of the file in the syntax of the server from which the file is to be downloaded. The **file** parameter is required unless it has been set with the **set loader** command on page 1-114. The **file** parameter is a full path name rather than just a file name. The only restriction is that the last part of the file parameter must be a valid file name for the Loader module. When **method** is set to **tftp**, **http**, **zmodem**, or **none**, valid file names are in the format `filename.ext` where *`filename`* is one to twenty eight characters long and *`ext`* is three characters long. The following are examples of valid file names for the **tftp**, **zmodem**, or **none** methods:

```
\user\public\filename.ext ; UNIX or DOS server
[network.cfg]filename.ext ; DEC VAX server
```

Starting at the end of the file name and working backwards, the first character not valid in file names delimits a valid file name for the router. If the slash at the beginning of the path is omitted in this command, the **load** command adds it. The following are examples of valid file names for the **http** method:

```
/path/filename.ext

path/filename.ext
```

The **username** parameter specifies the username for the **ldap** or **http** methods when server authentication is required.

Examples To download a release using the defaults set previously with the **set loader** command on page 1-114, use the command:

```
loa
```

To download the 28-761.rel release into the Flash File System from a TFTP server with IP address 172.16.8.5 and a one minute delay, use the command:

```
loa fi=28-761.rel des=fl se=172.16.8.5 del=60
```

To load a show.scp script from asynchronous port 1, use the command:

```
loa fi=show.scp asy=1
```

To load the show.scp script from asynchronous port 1 using the ZMODEM protocol, use the command:

```
loa asy=1 met=zmodem
```

To download the 8-191.rez file from the downloads directory on the web server at `www.company.com`, when a name server has been set, use the command:

```
loa met=http des=fl fi=/downloads/8-191.rez
   se=www.company.com
```

To download the 8-191.rez file from the download directory on the web server at www.company.com (with IP address 192.168.1.1) when a name server is not defined, use the command:

```
loa met=http des=fl fi=/downloads/8-191.rez se=192.168.1.1
```

To download the 8-191.rez file from the downloads directory on the web server at www.company.com using a proxy server at 192.168.1.2 and the default proxy port, use the command:

```
loa met=http des=fl fi=/downloads/8-191.rez http=192.168.1.1
   se=www.company.com
```

To download new code to the special boot area of flash, use the command:

```
loa fi=ar410B10.fbr ser=172.16.8.5 des=boot
```

To download reallylongfile.rez into the flash from a TFTP server with IP address 172.16.8.5 with a one minute delay, use the command:

```
loa fi=reallylongfile.rez des=flash se=172.16.8.5 del=60
```

The filename is similar to really~1.rez and saved to flash memory. All consequent edition, display, and upload reconciliations are completed by consulting the longname.lfn table file. This table provides either the name reallylong.rez or really~1.rez as a valid ID for file management.

To download reallylongfile.rez and save it as temporary.rez into flash from a TFTP server with IP address 172.16.8.5 with a one minute delay, use the command:

```
loa fi=reallylongfile.rez des=fl se=172.16.8.5 del=60
   destf=temporary.rez
```

The filename is tempor~1.rez and the file is saved to flash memory. All consequent edition, display, and upload reconciliations are completed by consulting the longname.lfn table file. This table provides either the name temporary.rez or tempor~1.rez as a valid ID for file management.

Related Commands

set loader
show loader
upload

login

Syntax

```
LOGIn [login-name]
```

where *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.

Description

This command is used to login to the router. The User Authentication Facility prompts the user for a login name (if not specified) and a password. The user must enter appropriate responses, pressing [Return] after each response. Characters entered at the password prompt are not echoed to the screen for security reasons.

The password prompt is displayed regardless of whether a password is required for the login name entered by the user. This makes it more difficult for an intruder to discover valid login name/password combinations.

If the user enters an invalid login name or password, the sequence is repeated a set number of times. If a valid login name and password has still not been entered the terminal or Telnet session is locked out for a period of time. During this period the password prompt is withheld, preventing the user from logging in or entering commands. The manager can specify the number of login attempts allowed and the length of the lockout period using the **set user** command on page 1-127.

This command is not normally required. The user is automatically prompted to enter a login name and password when accessing the router via Telnet or a terminal connected to an asynchronous port set to SECURE mode, or when accessing a dialup service via an asynchronous modem connected to an asynchronous port.

This command might be used to login from a terminal connected to an asynchronous port that is not in Secure mode in order to use facilities available to logged-in users, or to login as another user in order to acquire different rights, such as Manager privilege.

The **logon** command is an alias for **login**.

If a user starts a Telnet session to the router but does not login within one minute, the router automatically times out the session and terminates the Telnet connection.

Related Commands logoff

logoff

Syntax `LOgoff`

Description This command is used to log out from the router. For a terminal attached to an asynchronous port, the port returns to its default prompting state, either the login prompt for a port in Secure mode, or the command prompt. For a Telnet session the TCP connection is terminated. The **logout** command is an alias for **logoff**.

Related Commands login

mail

Syntax

```
MAIL TO=destination {FIle=filename|MESSage=message}
   [SUBject=subject] [ETRN=mail-domain]
```

where:

- *destination* is a character string 3 to 131 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character (_).
- *filename* is a filename in the format `[device]:filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:
  - uppercase and lowercase letters
  - digits (0–9)
  - the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

  Optionally, *device* specifies the physical memory device where the file is stored, which is flash. If *device* is specified, it must be separated from the rest of the file name by a colon (:). The file extension *ext* is any valid file type that contains text such as `.CFG`, `.SCP` and `.TXT`.
- *message* is a character string 1 to 131 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the space character, and the underscore character (_). If *subject* contains spaces, it must be in double quotes.
- *subject* is a character string 1 to 131 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), the space character, and the underscore character (_). If *subject* contains spaces, it must be in double quotes.
- *mail-domain* is a character string 3 to 63 characters long. Valid characters are uppercase and lowercase letters, digits (0-9), and the underscore character (_).

Description This command sends an email message or the contents of a file to the specified email address. This command can be issued only by a user with Security Officer privilege.

The **to** parameter specifies the email address where the email is to be sent. This is normally in the form `user@company.net`. However, if only the IP address of the destination mail host is known, on RFC 821 compliant mail servers, the IP address can be used by enclosing it in square brackets, e.g. `user@[202.49.73.5]`. Note that Microsoft mail servers are not RFC 821 compliant.

The **file** parameter specifies the name of a file on the router to send in the body of the email. The file type must be *text* and it must exist on the system.

The **message** parameter specifies a single line of text to send in the body of the email. The **message** and **file** parameters are mutually exclusive.

The **subject** parameter specifies the subject line to appear in the email. This field is not required but should normally be present in an email.

The **etrn** parameter sends an ETRN request (as defined in RFC 1985) to the remote mail server to forward any queued mail messages for the specified mail domain or host name. This can be used to assist mail servers that are connected to the Internet via dial-up rather than permanent connections. A trigger can be created to send an ETRN message to the email service provider each time the router connects to the Internet. Some mail servers reject email messages from hosts without reverse DNS entries.

Examples

To send an email message to user@testcom.com, use the command:

```
mail to=user@testcom.com SUBJ="Test Message" mess="Greetings
   from router 192.168.14.1"
```

To send an ETRN request to the mail server mserver1.isp.com to forward mail queued for users in the email domain “company.com”, use the command:

```
mail to=postman@mserver1.isp.com etrn=company.com
```

Related Commands

delete mail
set mail
show mail

modify

Syntax

```
MODify ADDR=address Size={Byte|Long|Word} VAL=value-list
    [SPace={SD|SP|UD|UP|UR}]
```

where:

- *address* is the base address of the block of memory to modify.
- *value-list* is either a list of up to five numbers (in hexadecimal) separated by commas (e.g. VAL=12,4ac,0,14e,65), or a text string of up to twenty characters surrounded by double quotes (e.g. VAL="string").

Description

This command modifies (overwrites) the contents of the router's memory. You can modify any memory or I/O device but this may interrupt the operation of the router. This command can be issued only by a user with Security Officer privilege.

The values to be written to memory are specified by the **val** parameter and are written to contiguous memory locations starting at the memory address specified by the **addr** parameter. The **size** parameter specifies how the values are written: **byte**, **long**(word), **word**. The **space** parameter is optional and can be used to select any of the valid CPU address spaces (Table 1-11 on page 1-83). If **space** is not specified, it defaults to **sd**.

The **modify** command is provided mainly as a diagnostic tool. It should not be needed for normal operation of the router.

Examples

This example modifies the first two words of memory starting at memory location 0x00000000:

```
mod addr=0 s=word val=5,6AA4
```

Related Commands

dump

purge file translationtable

Syntax

```
PURge FIle Translationtable={ALl|UPdate}
```

Description This command resynchronises the translation table to the file contents in memory. It is possible that the translation table and memory could become unsynchronised, for example in the event of a power outage during a file creation or deletion operation. This could possibly result in files in memory that are not represented in the translation table, and visa versa.

The **all** option completely rebuilds the translation table. All long file names are lost. The table continues to support subsequent long file name creation and management.

The **update** option restores all valid long file names to the appropriate table entries after the table has been rebuilt. Long file names that are not reconciled to the new table and table entries that are not confirmed to be in memory are deleted. This leaves a translation table that has maintained all of its previously valid data, and disposed of the rest. The table continues to support all subsequent long filename creation and management.

Examples To rebuild the translation table and remove all long file names, use the command:

```
pur fi t=al
```

To rebuild the translation table and attempt to recover all long file name data, use the command:

```
pur fi t=up
```

Related Commands show file

purge ldap

Syntax

```
PURge LDAP
```

Description This command removes all open LDAP requests and resets the LDAP module to its original empty state. It is most likely to be useful if an LDAP request locks.

Examples To reset the LDAP module, use the command:

```
pur ldap
```

Related Commands show ldap

purge user

Syntax `PURge USEr`

Description This command deletes all users from the User Authentication Database. The Manager account remains but the password is set to the default password, *friend*. Global configuration parameters and counters are not affected. To clear these counters use the **reset user** command on page 1-108.

Related Commands

add user
delete user
disable user
enable user rso
reset user
set user
show user

rename

Syntax REName *src-filename dest-filename*

where *src-filename* and *dest-filename* are file identifiers in the format `[device:]name.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

Description This command renames files and requires a user with Security Officer privilege.

The source file name must identify an existing file, and the destination file name must not already be in use. If the source file is not a text file, then the source and destination file extensions must be the same.

Caution must be taken when renaming files, such as patches, releases, licences and configurations, since they contain information which is vital to the intended operation of the router.

Examples To rename the "`boot.cfg`" file to "`saveboot.cfg`", use the command:

```
ren boot.cfg saveboot.cfg
```

Related Commands

delete file
show file

reset cpu utilisation

Syntax

```
RESET CPU UTILisation
```

Description This command resets CPU utilisation percentages back to 0%.

Examples To reset the CPU utilisation, use the command:

```
reset cpu util
```

Related Commands show cpu

reset gui

Syntax

```
RESET GUI
```

Description This command is used after a new GUI resource file has been loaded so that the router reads the updated file without the user rebooting the router.

Example To use details from the GUI resource file that has just been loaded onto the router, use the command:

```
reset gui
```

Related Commands

disable gui
enable gui
load
set install
show gui

reset http server

Syntax

```
RESET HTTP SERVer
```

Description This command resets the HTTP server. The server is restarted, debugging is disabled, and all counters are reset to zero (0).

Examples To reset the HTTP server, use the command:

```
reset http serv
```

Related Commands

disable http server
enable http server
set http server
show http server

reset loader

Syntax

```
RESET LOAder
```

Description

This command aborts the current file transfer being undertaken by the Loader module. All resources used by the transfer are released and any file in the process of being created is deleted. The Loader module becomes immediately ready for a new load to be initiated.

Related Commands

load
set loader
show loader

reset user

Syntax

```
RESET USEr[=login-name] [COUNTER[={ALL|GLOBAL|USER}]]
```

where *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.

Description

This command is used to reset User Authentication Database counters for one or all users, or to reset global counters for the User Authentication Facility.

If a login name is specified with the **user** parameter, the **counter** parameter is optional (only **user** can be specified) and the activity counters for the specified user are reset. The login name is not case sensitive.

If a login name is not specified, then the **counter** parameter is required to specify which counters should be reset. If **user** is specified, activity counters for all users are reset. If **global** is specified, global counters for the User Authentication Facility are reset. If **all** is specified, all counters are reset.

Examples

To reset the activity counters for user "Bruce", use the command:

```
reset use=bruce
```

To reset the activity counters for all users, use the command:

```
reset use counter=user
```

To reset the global counters, use the command:

```
reset use counter=global
```

Related Commands

add user
delete user
disable user
enable user rso
purge user
set user
show user

restart

Syntax

```
RESTART {REBoot|ROUter} [CONfig={filename|NONE}]
```

where *filename* is a file name in the format [`device:`]`filename.ext`. Invalid characters are * + = " | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ' ! @ # $ % ^ & ( ) _ - { }

Description

This command restarts the router with either the current configuration file (set with the **set config** command on page 1-110) or the specified configuration file.

If **reboot** is specified the router performs a cold start (hardware reset) and executes the default configuration file, if one is defined. The **config** parameter may not be specified.

If **router** is specified the router performs a warm start of all software modules (the hardware is not reset) and executes the default configuration file, if one is defined, (under SNMP this appears as a coldStart Trap). The **config** parameter may be used to specify a script or configuration file other than the current default. The file extension must be "`scp`" or "`cfg`". If **none** is specified, the router reboots without executing any configuration file.

If the router is operating in security mode and a configuration script is specified, the configuration script must create a user with Security Officer privilege, so that when the router restarts in security mode there is at least one user with sufficient privilege to execute critical commands. The router displays a warning message to this effect and prompt for a confirmation.

Examples

To restart the router using the "test.cfg" configuration file instead of the default configuration file, use the command:

```
restart rou config=test.cfg
```

Related Commands

show config
show exception
show startup

set config

Syntax

```
SET CONfig={filename|none}
```

where *filename* is a file name in the format [`device:`]`filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

Description

This command sets the script file that the router is to use as its default configuration. The file name is stored in flash.This command can be issued only by a user with Security Officer privilege. If the router is operating in security mode, the configuration script must create a user with Security Officer privilege, so that when the router restarts in security mode there is at least one user with sufficient privilege to execute critical commands. The router displays a warning message to this effect and prompts for a confirmation.

The **config** parameter specifies the name of the script or configuration file to use. The file extension must be “`scp`” or “`cfg`”. The file must already exist on the router. The commands in the script file are executed when the router is rebooted or performs a warm restart. If **none** is specified, the router boots with no configuration file.

Examples

To set the default configuration file to "boot.cfg", use the command:

```
set con=boot.cfg
```

Related Commands

restart
create config
show config

set help

Syntax `SET HELP=`*`helpfile`*

where *helpfile* is a file name in the format [`device:`]`filename.ext`. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

Description This command sets the system help file used by the **help** command on page 1-95. The **help** parameter specifies the name of the text file containing the help text for the router. If a device is not specified, the default is flash.

Examples To set the name of the help file to "e72-01", use the command:

```
set help=e72-01.hlp
```

Related Commands **help**
show system

set http server

Syntax

```
SET HTTP SERVer [POrt=port] [SECurity=ON|OFF|ENAbled|
   DIsabled|True|False] [SSLKey=key-id]
```

where:

- *key-id* is a number from 0 to 65535.
- *port* is a number from 0 to 65535.

Description This command sets the security options for when the router acts as an HTTP server.

The **port** parameter specifies on which TCP port number that the HTTP Server listens. The default is port 80.

The **security** parameter specifies whether the HTTP server accepts SSL secured HTTPS connections, or unsecured HTTP connections. When **security** is set to **on**, **enabled**, or **true**, all connections made to the server must be SSL connections. When set to **off**, **disabled**, or **false**, all connections made to the server must be non-SSL connections. The default is **off**.

Setting **security=on** enables SSL on the router. See Chapter 51, Secure Sockets Layer (SSL) for details on configuring SSL.

The **sslkey** parameter must contain a valid private key ID in order for SSL to operate. This parameter is required when the **ssl** parameter is on.

Example To enable the HTTP server for SSL secured connections with the SSL identification key “5”, use the command:

```
set http serv sec=on sslk=5
```

Related Commands

enable http server
reset http server
show http server
set ssl

set install

Syntax

```
SET INSTall={TEMPorary|PREFerred|DEFault} [GUI=file-name]
    [RELease=[release-name] [PATch[=patch-name]]
```

where:

- *release-name* is the name of a release file in the following format: [device:]filename.ext. Invalid characters are * + = “ | \ [ ] ; : ? / , < >, and wildcards are not allowed. Valid characters are:
  - uppercase and lowercase letters
  - digits (0–9)
  - the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }
- *file-name* is the name of the GUI resource file to be used.
- *patch-name* is the name of the patch file to set in this install.

Description

This command sets up release, GUI, and patch information for one of the installs. It requires a user with Security Officer privilege.

The **install** parameter specifies which install is to be set. The Install module maintains installation information and loads the correct information at boot. An *install* is a record identifying a release, an optional patch, and a GUI resource file. The Install module has three types of installs: temporary, preferred, and default.

The default install is the install of last resort. The release for the default install cannot be changed by the manager and is always the EPROM release. The patch for the default install may be set by the manager.

The temporary and preferred installs are completely configurable. The release, an associated patch, and a GUI resource file may be set.

The **release** parameter specifies the release file for this install. The release file is a filename in the following format for files in the file subsystem: [device:]filename.ext . The default device is flash.

The **gui** parameter specifies the resource file used when the GUI is accessed. The resource file name includes a product code, a language code and a version code, for example, `r_410e05.rsc` on an AR410 or `d450se01.rsc` on an AR450S.

The resource file must exist in flash, possess a valid checksum, be compatible with the product model it is being loaded onto, and be compatible with the current software release. By specifying a null string for filename such as “set install=preferred gui=”, no resource file is used and so the GUI is unavailable. The GUI is also unavailable if the **set install=preferred gui=none** command is entered.

Changing the resource file causes an implicit **reset gui** to be performed. The router reinitialises and reconstructs its index of pointers into the resource file so that the new GUI resource file is accessed correctly.

The installed GUI resource file can be deleted when the GUI is disabled. Use the **show install** command on page 1-158 and check the "Current Install" section to see which resource file is currently installed.

The **patch** parameter specifies the patch file for this install, and is a file name in the format `[device:]filename.ext`. The default is flash. If a patch name is not given, patch file information for a given install is removed and the release file is loaded as the install.

If the **patch** parameter is not present, patch file information for a given install is removed and the release file is loaded as the install.

A patch file cannot be set up for an install unless a release file is already set up, or a release file is specified in the same command. This stops the inadvertent setting of an install to be just a patch file. When the router reboots in such a case the particular install is ignored, which may have undesirable effects on operations.

Examples To set up the release file 8-240.rez, use the command:

```
set inst=pref rel=8-240.rez
```

To set the GUI resource file to `d450se03.rsc`, use the command:

```
set inst=pref gui=d450se03.rsc
```

Related Commands

delete install
reset gui
show install

set loader

Syntax

```
SET LOAder [ATTribute={CErt|CRl|CAcert|DEFault}]
   [BASeobject={dist-name|DEFault}] [DElay={delay|
   DEFault}] [DESTFile=destfilename]
   [DEStination={BOOTblock|FLash}] [HTTPproxy={hostname|
   ipadd|DEFault}] [METhod={HTTP|LDAP|TFtp|WEB|WWW|ZModem|
   NONE|DEFault}] [PASSword=password] [ASYn={port|
   DEFault}] [PROxyport={1..65535|DEFault}] [SRCFile|
   FIle=filename] [SErver={hostname|ipadd|ipv6add|
   DEFault}] [SERVport={1..65535|DEFault}]
   [USErname=username]
```

where:

- *dist-name* is an X.500 distinguished name, as described in "Distinguished Names (DN)" on page 1-50.
- *delay* is a time delay, in seconds.
- *destfilename* is a character string 5 to 20 characters long, specifying the name of the destination file in the router file system.
- *hostname* is a character string 1 to 40 characters long.
- *ipadd* is an IP address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address.
- *password* is a character string 1 to 60 characters long used for basic server authentication.

- *port* is the number of an asynchronous port. Ports are numbered sequentially from `asyn0`.
- *filename* is a character string 1 to 256 characters long. This is a full path name for the file to load in the syntax of the server where the file is to be loaded.
- *username* is a character string 1 to 60 characters long, used for basic server authentication.

Description This command sets defaults for the **load** command on page 1-96. All values that can be specified with the **load** command can also be specified as defaults with the **set loader** command. Parameters not specified in the **load** command use this default.

All parameters except **destfile**, **srcfile**, and **file** can be returned to their defaults with the **default** option.

The **attribute** parameter is a keyword specifying the type of object to retrieve from an LDAP repository. A list of currently recognised keywords and their respective object types can be found in Table 1-13 on page 1-97. If **default** is specified, this parameter is set to **cert**.

The **baseobject** parameter is required if the load method is LDAP and specifies the repository location of the object to load in the LDAP distinguished name format. If the string contains spaces, it must be in double quotes. The special characters {, = + < > # ; \ <CR> } must be escaped by typing a \ before the character, as defined in [RFC 1779], unless they are used for their prescribed purpose. For example, to include a \ in the string, type \\ and to include a #, type \#.

The **delay** parameter specifies the delay, in seconds, between initiating the file download and the download actually starting. This feature is provided to allow reconfiguration of ports and devices after initiating the download. For example, a manager may be at a remote site with a single PC that must act as both the access device to the router and the TFTP server. By specifying a delay, the manager has time to reconfigure the PC from terminal emulation mode to TFTP server mode before the download starts. The **delay** parameter is optional. If **default** is specified, this parameter is set to no delay.

The **destfile** parameter specifies the name of the destination file in the router file system. When **method** is set to **ldap**, the extension of the destination file must be valid for the type of object being loaded (`"cer"` or `"crl"`).

The **destination** parameter specifies where the file is to be stored. If **bootblock** is specified, the file is stored in the special boot code area of flash reserved for the router boot code. Only boot code release files (with extension `.fbr`) may be loaded to the boot code area. If **flash** is specified, the file is stored in the Flash File System (FFS) on the router. Patch files, release files, and script files may be stored in flash. If **destination** is not specified, and has not been set with the **set loader** command on page 1-114, the default is flash.

The boot code should not normally need to be upgraded. While loading a new router boot code file onto the flash boot code area, the router must not lose power. When the router goes through a power cycle while writing to the bootblock, the code used to reboot the router will be incomplete, and the router cannot be rebooted.
The router does not respond to any interfaces while the boot block is being written. The router should be idle while the boot block is being reloaded. The router must have sufficient free buffers (about 600) when commencing the download to be able to store the entire boot code.

The **httpproxy** parameter specifies the proxy server used to handle HTTP requests. Either the IP address or the fully qualified domain name of the proxy server may be specified. If a domain name is specified, the router performs a DNS lookup to resolve the name. If **default** is specified, this parameter is set to the default, which has no value set for **httpproxy** and clears previous default settings.

The **method** parameter specifies the method to use when downloading the file. If **http** is specified, HTTP is used to download the file. The **web** and **www** options are synonyms for HTTP. If **ldap** is specified, LDAP is used to download the file. If **tftp** is specified, TFTP is used to download the file.

If **zmodem** is specified, the ZMODEM protocol is used to download the file. If **zmodem** is specified, the **asyn** parameter must be specified. If **none** is specified, text files can be downloaded and all input received through the port is directed to the specified file on the router’s file subsystem. The file transfer is terminated by the first control character received that is not a CR or LF character.

The **file** parameter is not used when **method** is set to **zmodem** or **ldap**. The **asyn** parameter is not valid when **method** is set to **http**, **web**, **www**, **ldap**, or **tftp**. If **default** is specified, the parameter is set to **tftp**.

The **password** parameter (and/or the **username** parameter) sets a default to use under the HTTP or LDAP method when server authentication is required. If **default** is specified, the previous default is cleared and server authentication is not used.

The username and password defaults cannot be the text string “default” (or part thereof, not case sensitive). If the user requires that the username or password be the word “default”, it must be specified on the command line when the **load** command is invoked.

The **asyn** parameter specifies the asynchronous port via which file is to be downloaded when the **method** parameter is set to **zmodem** or **none**. If **method** is set to **zmodem** or **none**, the **asyn** parameter is required. If **default** is specified, previous defaults are cleared and the parameter is set to no ASYN.

The **proxyport** parameter specifies the port on a proxy server. The **proxyport** parameter is valid if **method** is **http** and **httpproxy** is specified. If **default** is specified, this parameter is set to 80.

The **srcfile** or **file** parameter specifies the name of the file, in the syntax of the server from which the file is downloaded. The **file** parameter is a full path name rather than just a file name. The only restriction is that the last part of the parameter must be a valid file name for the Loader module. When **method** is set to **tftp**, **http**, **zmodem** or **none**, valid file names have the format filename.ext where *filename* is one to twenty eight characters long and `ext` is three characters long. The following are examples of valid file names for **tftp**, **zmodem**, or **none** methods:

```
\user\public\filename.ext ; UNIX or DOS server
[network.cfg]filename.ext ; DEC VAX server
```

Starting at the end of the file name and working backwards, the first character not valid in file names delimits a valid file name for the router. If the slash at the beginning of the path is omitted in this command, the **load** command adds it. The following are examples of valid file names for HTTP method:

```
/path/filename.ext
```

```
path/filename.ext
```

The **server** parameter specifies the IP address or the host name (a fully qualified domain name) of the TFTP server or HTTP server from which the file is loaded. If a host name is specified, a DNS lookup is used to translate this to an IP address. See **set ip nameserver** command on page 14-151 of Chapter 14, Internet Protocol (IP) for more information about setting up name servers. The **ping** command on page 14-129 of Chapter 14, Internet Protocol (IP) can verify that the router can communicate with the server via IP. The **server** parameter is not used when **method** is set to **zmodem** or **none**. The following are examples of valid server names when **method** is set to **http**:

```
host.company.com
```

```
192.168.3.4
```

If **default** is specified, previous defaults are cleared and no value is set for **server**.

The **servport** parameter optionally specifies the port on the HTTP or LDAP server from which the file is loaded. If **default** is specified and a load starts, a default is invoked according to the load method. In this case, **servport** takes a value of 80 for HTTP, and 389 for LDAP.

The **username** parameter (and/or the **password** parameter) sets a default to use under the HTTP or LDAP methods if server authentication is required. If **default** is specified, previous defaults are cleared and server authentication is not used.

The username and password defaults cannot be set to the actual text string “default” (or partial and not case sensitive). If the user requires that either the username or password be the word “default”, it must be specified on the command line when the **load** command is invoked.

Examples To set the default download parameters to be release `28-72.rel` downloaded into the Flash File System from the TFTP server with IP address 172.16.8.5, with a delay of one minute, use the command:

```
set load file=28-72.rel destination=flash server=172.16.8.5
   delay=60
```

To clear defaults previously set with the **set loader** command (except the filename), and restore defaults to the loader module, use the command:

```
set loader attribute=default delay=default
   destination=default httpproxy=default method=default
   password=default asyn=default proxyport=default
   server=default servport=default username=default
```

Related Commands load
show loader

set mail

Syntax `SET MAIL HOSTname=`*`hostname`* `[SMTPserver=`*`ipadd`*`]`

where:

- *hostname* is a character string 1 to 63 characters long. Valid characters are any character except spaces (“ “), control characters (ASCII 0–31 and 127) and the special characters “()<>@,;:\"[]”.
- *ipadd* is the IP address in dotted decimal notation.

Description This command sets the host name used by the mail system when it communicates with other mail systems. The **hostname** parameter is typically the fully specified domain name of the router, e.g. `router1.myorg.com`. The host name appears in the *From* field of the message header when the message is received by the remote mail system. The mail system is not enabled until the host name has been specified.

The **smtpserver** parameter specifies the IP address of the mail server where the mail from the router is to be sent. When set, the address pre-empts the use of a DNS lookup for the domain name of the destination email address specified in the **mail** command.

Examples To set the mail host name to router1.myorg.com, use the command:

```
set mail host=router1.myorg.com
```

To set the mail destination SMTP server to 192.168.6.100 for admin.myorg.com, use the command:

```
set mail host=admin.myorg.com smtp=192.168.6.100
```

Related Commands show mail
mail

set manager asyn

Syntax

```
SET MAnager ASYn={port-number|NONE}
```

where *port-number* is the number of the port. Ports are numbered sequentially starting with `asyn` 0

Description

This command sets the semipermanent manager port. If a valid port number is specified, the port becomes the semipermanent manager port. If the specified port was secure before the command was entered, it loses its secure setting. If another port is currently the semipermanent manager port, then that port loses its semipermanent Manager privilege and becomes a secure port. If **none** is specified, the current semipermanent manager port loses its semipermanent Manager privilege and becomes a secure port. There can be only one semipermanent manager port at a time.

This command is one of the security commands (see "Database Security" on page 1-18). When the security timer expires before the command is entered, the manager is prompted to re-enter the password for the login name where the command was issued.

Examples

To set `asyn` 0 as the semipermanent manager port, use the command:

```
set ma asy=0
```

To remove the semipermanent manager port, use the command:

```
set ma asy=none
```

Related Commands

login
show manager asyn
set asyn in Chapter 7, Interfaces

set password

Syntax `SET PASSword`

Description This command changes the password for the user currently logged into the port where the command was issued. When properly logged in, the user is prompted for the current password, the new one, and confirmation of the new one. The passwords are not echoed to the screen. If a user is not logged into the port, an error message is displayed.

The new password and the confirmation must be identical for the change to take affect. This reduces the chances of a typing error causing the password to be different from what the user intended.

A log message is generated whenever the password for an account with Manager privilege is changed. A configurable minimum password length is enforced. The default is 6 characters.

Examples To change the password for the current user, use the command:

```
set password
old password:
new password:
Confirm:
```

Related Commands **add user**
set user

set skey

Syntax `SET SKEY [METhod={SKEY|OTP}] [ENCryption={MD4|MD5]`

Description This command sets the method of one-time password authentication to use, and the type of encryption to use during one-time password generation with the **show skey** command. S/Key commands have a User privilege level.

The **method** parameter specifies whether to use the S/Key or OTP authentication technique. The default is **skey**.

The **encryption** parameter specifies whether to use MD4 or MD5 encryption. The default is **md4**.

Examples To set up one-time passwords using the OTP method and MD5 encryption, use the command:

```
set skey met=otp enc=md5
```

Related Commands **show skey**

set system contact

Syntax

```
SET SYStem CONtact=contact-name
```

where *contact-name* is a character string 1 to 255 characters long. Valid characters are any printable character. If the string includes spaces, it must be in double quotes.

Description

This command assigns a string that defines the contact name for this router. Commands from the command line are limited to 128 characters including the prompt. The text is in the output of the **show system** command on page 1-171. It also updates the MIB object *sysContact*, which can then be read using SNMP.

Examples

To set the contact name for this router to “Bruce Johns, 64-3-343-0803”, use the command:

```
set sys con="Bruce Johns, 64-3-343-0803"
```

Related Commands

set system location
set system name
set system territory
show system

set system country

Syntax

```
SET SYStem COUntry={AUSTRAlia|AUSTRIa|BELgium|CANada|
    DENMark|EIRE|FINLand|FRAnce|GERmany|ICELand|ITAly|
    NETHerlands|NEWZealand|NORWay|PORTUgal|SINGapore|SPAIn|
    SWITZerland|SWEden|TURkey|UAE|UK|USA|NONE}
```

Description This command sets the country identifier for the router, and sets corresponding defaults for ATM.

The **country** parameter specifies for which country to set the ATM defaults (Table 2-12). If **none** is specified, then the ATM default VPI/VCI pair is set to 0/35 and the encapsulation to **llc**. The default is **none**.

Table 1-14: Default settings for ATM virtual channels dependent on the **country** parameter

country parameter	Default VPI	Default VCI	Default encapsulation mode
NONE (default)	0	35	LLC
AUSTRAlia	8	35	LLC
AUSTRIa	8	48	VCMux
BELgium	8	35	VCMux
CANada	0	35	LLC
DENMark	0	35	LLC
EIRE	0	38	VCMux
FINLand	0	40	LLC
FRAnce	8	35	VCMux
GERmany	1	32	LLC
GREece	8	35	VCMux
ICELand	0	35	VCMux
ITAly	8	35	VCMux
NETHerlands	0	48	VCMux
NEWZealand	0	100	VCMux
NORWay	8	35	LLC
PORTUgal	8	32	LLC
SINGapore	0	100	VCMux
SPAIn	8	32	LLC
SWITZerland	8	48	VCMux
SWEden	8	35	LLC
TURKey	8	35	LLC
UAE	0	100	VCMux
UK	0	38	VCMux
USA	0	35	LLC

Example To set the county to Australia use the command.

```
set sys cou=austra
```

Related Commands

add atm channel
set atm channel
show system

set system distinguishedname

Syntax

```
SET SYStem DIStinguihsedname={dist-name|NONE}
```

where *dist-name* is an X.500 distinguished name as described in "Distinguished Names (DN)" on page 1-50

Description

This command sets the router's distinguished name for PKI and ISAKMP to use.

The **distinguishedname** parameter specifies the name. If **none** is specified, the router's distinguished name is an empty string. The **none** option can be used when the router will be identified in a PKI certificate by an alternative name, such as its IP address. For compatibility with other implementations of PKI, we recommend this not be done.

Examples

To set the router's distinguished name, use the command:

```
set sys dis="cn=router1, o=company1, c=us"
```

Related Commands

show system

set system location

Syntax

```
SET SYStem LOCation=location
```

where *location* is a character string 1 to 255 characters long. Valid characters are any printable character. If the string includes spaces, it must be in double quotes.

Description

This command assigns a string defining the physical location of this router. For example "Laboratory, First Floor, Head Office Building". Commands run from the command line are limited to 128 characters including the prompt. The text is in the output of the **show system** command on page 1-171. It also updates the MIB object *sysLocation*, which can then be read using SNMP.

Examples

To set the location for this router to "Laboratory, First Floor, Head Office Building", use the command:

```
set sys loc="Laboratory, First Floor, Head Office Building"
```

Related Commands

set system contact
set system name
set system territory
show system

set system name

Syntax

```
SET SYStem NAMe=name
```

where *name* is a character string 1 to 255 characters long. Valid characters are any printable character. If the string includes spaces, it must be in double quotes.

Description

This command assigns a string defining the name of this router. By convention this is the full domain name of the IP entity. For example, `nd1.co.nz`. Commands run from the command line are limited to 128 characters including the prompt. The text is in the output of the **show system** command on page 1-171. It also updates the MIB object *sysName*, which can then be read using SNMP.

Examples

To set the name for this router to “nd1.co.nz”, use the command:

```
set sys nam=nd1.co.nz
```

Related Commands

set system contact
set system location
set system territory
show system

set system territory

Syntax

```
SET SYStem TERritory={AUStralia|CHIna|EURope|JAPan|KORea|
   NEWZealand|USA}
```

Description

This command assigns a territory identifier for the router. The territory identifier is used by the Q.931, PRI, and PBX modules to set defaults that are appropriate for the territory in which the router is being operated. The default territory is **europe**.

If the router territory identifier is changed, parameters in the Q.931, PRI, and PBX modules that are influenced by the territory in which the router is being operated are automatically changed to values appropriate for the new territory setting. If the current territory value is specified, i.e. the territory is unchanged, then the module parameters are restored to the defaults for that territory.

Examples

To set the name for this router to Australia, use the command:

```
set sys ter=aus
```

Related Commands

set system contact
set pri
set q931
set system country command on page 1-122 of Chapter 1, Operation
set system location
set system name
show pri configuration
show pri state
show q931
show system

set tacplus key

Syntax

```
SET TACPlus Key=key
```

where *key* is a string of up to 64 characters

Description This command sets a new global key for TACACS+ servers. The **key** parameter specifies the new global secret key.

Examples To modify the global key on the TACACS+ server, use the command:

```
set tacp k=trynot2useMe2atAll
```

Related Commands show tacplus key

set tacplus server

Syntax

```
SET TACPlus SERVer=ipaddress [Key=key] [POrt=port]
    [SINGLEconnection={Yes|No}] [TIMEOUT=1..10]
    [LOCAL={NONE|1..15}]
```

where:

- *ipaddress* is an IP address in dotted decimal notation.
- *key* is a key string of up to 64 characters.
- *port* is an integer value.

Description This command modifies parameters already set for a TACACS+ server.

The **server** parameter specifies the IP address of the TACACS+ server to be modified.

The **key** parameter specifies the secret key to be modified.

The **port** parameter specifies the TCP port to be modified.

The **timeout** parameter specifies the period of time (in seconds) that the router waits for a response from the TACACS+ server before it times out.

The **singleconnection** parameter specifies whether multiple TACACS+ sessions are supported.

The **local** parameter specifies a local interface to be used as the source for all TACACS+ packets the device sends to this TACACS server. The local interface must already be configured and fall in the range 1-15. If either the parameter is not set or **none** is specified, the switch selects a source from the current available interfaces instead.

Examples To change timeout from 3 seconds to 2 seconds and change the **singleconnection** parameter to **yes**, use the command:

```
set tacp serv=192.168.196.22 k=newkey4atr2supportacasplus
   timeout=2 single=y
```

Related Commands add tacplus server
delete tacplus server
show tacplus server

set tacplus telnet

Syntax `SET TACPlus TELnet={0..15|None}`

Description This command determines whether or not TACACS+ authenticated users can telnet from the router.

The **telnet** parameter specifies the minimum TACACS+ privilege level required for using telnet on the router. A value of **none** disables telnet for all TACACS+ authenticated users. A value of **1** indicates that all users can telnet. A value of **7** indicates that Manager privilege or better is required. A value of **15** is equivalent to Security Officer privilege. The default is **none**.

Examples To allow telnet for TACACS+ authenticated Security Officers, use the command:

```
set tacp tel=15
```

Related Commands show tacplus telnet

set time

Syntax `SET [TIme=`*`time`*`] [DAte=`*`date`*`]`

where:

- *time* is the time in 24 hour format `(hh:mm:ss)`.
- *date* is the date in `d-mmm-yyyy`, `dd-mmm-yy`, or `dd-mmm-yyyy format` where the month is the first three letters of the month (e.g. APR). The day of the month can be one or two digits, and the year can be two or four digits.

Description This command sets the time and/or date stored in the router's real-time clock.

Examples The following commands set the router's real-time clock to 10 p.m. on 29 January 2004:

```
set ti=22:00:00
set da=29-JAN-04
```

Related Commands show time

set user

Syntax

```
SET USEr=login-name [CALLingnumber=number]
    [CBNUMber=e164number] [DESCription=description]
    [Ipaddress=ipadd] [IPXnetwork=network]
    [LOgin={True|False|ON|OFf|Yes|No}] [MASk=ipadd]
    [MTu=40..1500] [NETMASK=ipadd] [PAssword=password]
    [PRivilege={USer|MAnager|SEcurityofficer}]
    [TELnet={Yes|No}]

SET USEr [LOGINFail=1..10] [LOCkoutpd=0..30000]
    [MANpwdfail=1..5] [Securedelay=10..600]
    [MInpwdlen=1..23] [TACRetries=0..10] [TACTimeout=1..60]
    [LOgin={True|False|ON|OFf|Yes|No}]
```

where:

- *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.
- *password* is a character string 1 to 32 characters long. Valid characters are any printable character. If the string contains spaces, it must be in double quotes.
- *number* is an ISDN number 1 to 32 characters long. Valid characters are any printable characters, but the calling number it is to match is likely to contain only decimal digits. If the string contains spaces, it must be in double quotes.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string 1 to 23 characters long. Valid characters are any printable character. If the string contains spaces, it must be in double quotes.
- *ipadd* is an IP address in dotted decimal notation.
- *network* is a valid Novell network number, expressed as a hexadecimal number. Leading zeros may be omitted.

Description

This command modifies a user record in the User Authentication Database or alters global parameters affecting the User Authentication Facility.

The first variant of the command is used to alter a user record in the User Authentication Database. The **user** parameter specifies the login name of a user in the database. Other parameters specified on the command modify the information stored in the database for that user. The second variant of the command is used to alter the global security parameters for the User Authentication Facility.

The **callingnumber** parameter specifies the calling number to be used to authenticate incoming calls from L2TP and ISDN services that provide caller ID information.

The **cbnumber** parameter specifies the ISDN number to use when making a call back to a remote user using the PPP callback facility.

The **description** parameter specifies text for the entry such as the full name and location of the user. This string may contain any printing character and the case is preserved in output.

If you use this command to change a user's privilege level, you must also specify a case-sensitive **password**. This parameter should be initially set for the user and the user can change it to a private one by using the command:

```
set password
```

A password set with the **set password** command on page 1-120 may contain any printing character. A configurable minimum password length is enforced. The default is 6 characters. The **password** parameter is required when the **privilege** parameter is specified.

The **privilege** parameter specifies the privilege level for the user. The default is User. A user with User privilege has access to a limited subset of commands, generally commands that affect the user's own session or asynchronous port. A user with Manager privilege has access to the complete router command set when the router is operating in normal mode, or a subset of commands when the router is operating in security mode. A user with Security Officer privilege has access to the full set of commands, and in particular, can access security commands while the router is operating in security mode. The **password** parameter is required when the **privilege** parameter is specified.

The **telnet** parameter specifies whether the user is permitted to use the **telnet** command on page 21-31 of Chapter 21, Terminal Server to telnet to another host, or the **connect** command on page 21-15 of Chapter 21, Terminal Server to access a Telnet service when logged in through Telnet.

The **ipaddress** parameter specifies an IP address for the user. The value must be a valid IP address in dotted decimal form.

The **mtu** parameter specifies a Maximum Transmission Unit value for the user. The value must be a decimal integer from 40 to 1500 inclusive.

The **netmask** parameter and the MASK parameter are synonymous.

The **ipaddress**, **mask** and **mtu** parameters are required if the user is to login in order to make a PPP or SLIP connection to the router over a modem connected to an asynchronous port.

The **ipxnetwork** parameter specifies the Novell network number assigned to the user accessing a Novell internetwork. See Chapter 19, Novell IPX for more information. The network number may be cleared by setting this parameter to **none** instead of a network number. The default is none.

The **loginfail** parameter sets the number of successive login failures a user may make before the login prompt is withheld for the lockout period. The default is 3.

The **lockoutpd** parameter sets the number of seconds that the login prompt is withheld when the number of login retries exceeds the value set by **loginfail**. The default is 600 seconds.

The **manpwdfail** parameter sets the number of successive attempts a manager may make to enter the correct password while entering a security command before the session is automatically logged off. The default is 3.

The **mask** parameter specifies the address mask that extends the range of IP addresses. If the mask parameter is not present, a mask of 255.255.255.255 is used. The address and mask must be internally consistent in that the result of ANDing the address and mask should be the address.

The **securedelay** parameter sets the number of seconds that may elapse between the entry of one security command and the next without the user being required to re-enter the Security Officer password to validate the command. The default is 60 seconds.

The **minpwdlen** parameter sets the minimum password length that is enforced for the **add user** and **set password** commands. The default is 6 characters.

The **tacretries** parameter sets the number of times a TACACS request is resent when a response is not received within the timeout period. The default is 3.

The **tactimeout** parameter sets the number of seconds the router waits for a TACACS response before retransmitting the request or giving up after the number of retries is reached. The default is 5 seconds.

The **login** parameter specifies whether users with a privilege of user are able to log into the router. If used without a login-name, it changes all login values for user privileged users currently in the User Database. If a valid login-name is used, the login value of that specific user is changed. If **false**, the user is authenticated by the User Database but not allowed to log into the router. If **true**, the user logs into the router and enters commands. The default is **false**.

Examples

To change the password to “BZ4gal” and the privilege level to Manager for user Bruce, use the command:

```
set use=bruce pa=BZ4gal pr=ma
```

To change the minimum password length to eight characters for all users, use the command:

```
set use mi=8
```

Related Commands

add user
delete user
disable system security_mode
disable user
enable system security_mode
enable user rso
purge user
reset user
show user

show alias

Syntax `SHow ALIas`

Description This command displays any aliases currently defined on the router (Figure 1-14, Table 1-15).

Figure 1-14: Example output from the **show alias** command

```
Alias ....... df
  String .... delete file=1-190.rez

Alias ....... ii
  String .... ip interface
```

Table 1-15: Parameters in the output of the **show alias** command

Parameter	Meaning
Alias	The name of the alias.
String	The string substituted for the alias when it appears in a command line.

Related Commands **add alias**
delete alias

show buffer

Syntax `SHow BUFfer [SCAn[=address [QUEuepointers]]]`

where *address* is the memory address of a section of router code expressed in hexadecimal

Description This command displays information about the memory buffers in use by router modules. If no optional parameters are specified, a summary of the buffers in use is displayed (Figure 1-15 on page 1-131, Table 1-16 on page 1-131).

The **scan** and **queuepointers** parameters display low-level debugging information. Use them only when requested by technical support personnel.

The **scan** parameter displays detailed information about buffers usage. If an address is not specified, the memory addresses of sections of router code and the number of buffers in use by that section are displayed (Figure 1-16 on page 1-132). If an address is specified, the addresses of the buffers in use by that section of router code are displayed (Figure 1-17 on page 1-133). The value for **address** is obtained from the output of the **show buffer scan** command.

The **queuepointers** parameter displays additional information about the contents of the buffers used by the router code section at the specified address (Figure 1-18 on page 1-133), and is valid when the **scan** parameter is specified with a valid address.

Figure 1-15: Example output from the **show buffer** command

```
Memory ( DRAM ) .......... 16384 kB
Free Memory .............. 48 %
Free fast buffers ........ 1799
Total fast buffers ....... 1802
Free buffers ............. 4013
Total buffers ............ 4096
Buffer level 3 ........... 125  (don't process input frames)
Buffer level 2 ........... 250  (don't do monitor or command output)
Buffer level 1 ........... 500  (don't buffer up log messages)
```

Table 1-16: Parameters in the output of the **show buffer** command

Parameter	Meaning
Memory (DRAM)	The total amount of DRAM installed in the router.
Free memory	The amount of free (unused) memory, as a percentage of total available memory.
Free fast buffers	[Power PC based router only] The number of free (unused) fast memory buffers. Fast buffer memory is cached by the CPU and is available for program variable storage. It cannot be used for packet buffers.
Total fast buffers	[Power PC based routers only] The total number of fast memory buffers.
Free buffers	The number of free (unused) memory buffers.
Total buffers	The total number of memory buffers.
Buffer level n	Levels where certain processes are halted when the value of "Free buffers" drops below that level.

Figure 1-16: Example output from the **show buffer scan** command

```
Scan of buffers in use

00093d62    2  001338a2    1  0013d27c    1  000cd26a    1  000ccfc2    7
000cd326    5  000cd542    1  0006d1f0    1  000a03e4    1  000a4256    1
001f544e    1  001f5484    1  001f54c0    1  000a50da    1  00082e52    1
0013fe40    2  0008c8b0    1  0008c8f0    1  0008c92c    1  0008f7f6    1
000ebd32    1  000ec0a2    2  000ec364    3  00080048    8  00081352    1
0016ef96    1  0012fd76    1  0012f64a    1  00086e3c    1  0008871a    1
000b6866    1  001f5338   10  001526e0    1  0011e892    2  00099486    1
001194d4    1  0011deb0   17  0011fd6a    2  0011d278    1  001139a4    1
0011b354    1  0011d7e8    1  001fe0ca    1  001fb446    1  001fb48c    2
001fb4e8    2  001fb52a    1  0005e95c    1  0005e9f8    1  000d3976    1
00161596    1  00153b60    1  000994ae    1  000d133e    1  000bbc3a    1
00163154    1  001069fc    1  000a4916    1  000a5298    1  00141e26    1
00157156    1  000f4028    1  00169bd8    1  000a9654    1  001352a4   16
000892ae    1  001524fa    1  00087014    1  00089666    1  0008625c    1
0012f6d2    1  00141e30    1  00141e3a    1  0014190e    1  00141940    1
000c512a   15  00087624    1

Total buffers in use - 84


Scan of fast buffers in use

002e3644    1 002f2170     2

Total fast buffers in use -3

Scan of additional fast buffers in use

003c43d8   40 003d2690    56

Total additional fast buffers in use -96


  Memory ( DRAM ) .......... 16384 kB
  Free Memory .............. 48 %
  Free fast buffers ........ 1799
  Total fast buffers ....... 1802
  Free buffers ............. 4013
  Total buffers ............ 4096
  Buffer level 3 ........... 125  (don't process input frames)
  Buffer level 2 ........... 250  (don't do monitor or command output)
  Buffer level 1 ........... 500  (don't buffer up log messages)
```

Figure 1-17: Example output from the **show buffer scan** command for a specified address

```
002c93bc  002ce7bc  002d42bc  002d49bc  002d57bc  002d5ebc
002d65bc  002df8bc  002dffbc  002e0dbc  002e14bc  002eaebc
002eb5bc  002ec3bc  002ecabc


  Memory ( DRAM ) .......... 16384 kB
  Free Memory .............. 48 %
  Free fast buffers ........ 1799
  Total fast buffers ....... 1802
  Free buffers ............. 4013
  Total buffers ............ 4096
  Buffer level 3 ........... 125  (don't process input frames)
  Buffer level 2 ........... 250  (don't do monitor or command output)
  Buffer level 1 ........... 500  (don't buffer up log messages)
```

Figure 1-18: Example output from the **show buffer scan queuepointers** command

```
002c93bc  002df8bc  002d5ebc  002c9434    002ce7bc  002e0dbc  002dffbc  002ce834
002d42bc  002d49bc  002569f0  002d4334    002d49bc  002d57bc  002d42bc  002d4a34
002d57bc  002d5ebc  002d49bc  002d5834    002d5ebc  002c93bc  002d57bc  002d5f34
002d65bc  002ec3bc  002eb5bc  002d6634    002df8bc  002dffbc  002c93bc  002df934
002dffbc  002ce7bc  002df8bc  002e0034    002e0dbc  002e14bc  002ce7bc  002e0e34
002e14bc  002eaebc  002e0dbc  002e1534    002eaebc  002eb5bc  002e14bc  002eaf34
002eb5bc  002d65bc  002eaebc  002eb634    002ec3bc  002ecabc  002d65bc  002ec434
002ecabc  002569f0  002ec3bc  002ecb34


  Memory ( DRAM ) .......... 16384 kB
  Free Memory .............. 48 %
  Free fast buffers ........ 1799
  Total fast buffers ....... 1802
  Free buffers ............. 4013
  Total buffers ............ 4096
  Buffer level 3 ........... 125  (don't process input frames)
  Buffer level 2 ........... 250  (don't do monitor or command output)
  Buffer level 1 ........... 500  (don't buffer up log messages)
```

show config

Syntax `SHow CONfig [DYNamic[=`*`module-id`*`]]`

where *module-id* is the name of a router module. See "Module Identifiers and Names" on page C-2 of Appendix C, Reference Tables for a complete list.

Description This command displays the current configuration file for the router, or the current dynamic configuration for the router or specified software module.

If no optional parameters are specified, the current default configuration file (set with the **set config** command on page 1-110) is displayed, along with information about how the current configuration was obtained (Figure 1-19 on page 1-134, Table 1-17 on page 1-134).

The **dynamic** parameter displays the current dynamic configuration of the router or of a specific software module. The information displayed is the sequence of router commands required to recreate the current dynamic configuration.

Figure 1-19: Example output from the **show config** command

```
Boot configuration file: boot.cfg (exists)
Current configuration: boot.cfg
```

Table 1-17: Parameters in the output of the **show config** command

Parameter	Meaning
Boot configuration file	The current boot configuration file set with the **set config** command on page 1-110, and whether the file exists: Not set - The boot configuration file has not been set
	<*filename*> (exists) - The boot configuration file has been set to <*filename*> and <*filename*> exists.
	<*filename*> (doesn't exist) -The boot configuration file has been set to <*filename*> but <*filename*> does not exist.
Current Configuration	The source of the current configuration:
	None - The router booted up with no configuration, because there was no configuration file set, the file `boot.cfg` was not found, the DIP switches were not set for a special configuration and there is no NVS in the router to upgrade from (or the router release is for a newer model that does not have NVS); or the user entered "s" or "S" in response to the "Force EPROM download" message.
	<*filename*> (warm restart) - The router booted up using <*filename*>, but this was a warm restart (RESTART ROUTER CONF=<*filename*>).
	None (file not found) - The router booted up with no configuration because the required configuration file was not found. The commands **restart router conf=<*filename*>** and **set conf=<*filename*>** check that the file exists, but it is possible to execute a **set config** command and then delete the file.
	<*filename*> - The router booted from the <*filename*> configuration file. This is the normal case.
	Receiver sensitivity test script (DIP switch) - The router's DIP switches are set to force the router to execute a configuration for factory testing. This case should never be seen.

Table 1-17: Parameters in the output of the **show config** command (continued)

Parameter	Meaning
	Remote configuration script (DIP switch) - The router's DIP switches are set to execute a special configuration designed to allow a manager to dial in and configure the router. There are two DIP switch settings that can cause this message. One always forces this configuration; the other runs the special configuration when a valid configuration file is not found (either one set or `boot.cfg`).
	<file> (default) - The router booted from the default configuration file, `boot.cfg`, because a configuration file has not been set. The router looks for `boot.cfg` in flash.

Examples To display the default configuration file, use the command:

```
sh con
```

To display the current dynamic configuration of the router, use the command:

```
sh con dyn
```

To display the current dynamic configuration of just the IPX routing software, use the command:

```
sh con dyn=ipx
```

Related Commands

restart
create config
set config

show cpu

Syntax `SHow CPU`

Description This command displays information about CPU utilisation router (Figure 1-20 on page 1-136, Table 1-18 on page 1-136).

Figure 1-20: Example output from the **show cpu** command

```
CPU Utilisation ( as a percentage )
----------------------------------------
Maximum since router restarted ..... 80
Maximum over last 5 minutes ........ 80
Average since router restarted ..... 37
Average over last 5 minutes ........ 37
Average over last minute ........... 32
Average over last 10 seconds ....... 31
Average over last second ........... 32
----------------------------------------
```

Table 1-18: Parameters in the output of the **show cpu** command

Parameter	Meaning
Maximum since router restarted	Maximum CPU utilisation recorded since the router restarted.
Maximum over last 5 minutes	Maximum CPU utilisation achieved over the last five minutes.
Average since router restarted	Average CPU utilisation recorded since the router restarted, as a percentage of total CPU capacity.
Average over last 5 minutes	Average CPU utilisation over the last five minutes, as a percentage of total CPU capacity.
Average over last minute	Average CPU utilisation over the last minute, as a percentage of total CPU capacity.
Average over last 10 seconds	Average CPU utilisation over the last 10 seconds, as a percentage of total CPU capacity.
Average over last second	Average CPU utilisation over the last second, as a percentage of total CPU capacity.

Related Commands show buffer

show debug

Syntax `SHow DEBug [STAck|FULl]`

Description This command displays a snapshot of the state of the router immediately before the last fatal condition, and is used for debugging. The output depends on the router’s security mode and the user’s privileges. Possible command variations are in Table 1-19 on page 1-137.

The **stack** parameter limits the output to a stack dump, if one is available. The output depends on whether the last fatal condition was a hardware reset or a software reboot. After a software reboot, the output is a stack dump (Figure 1-21 on page 1-142). After a hardware reset, no stack dump information is available and a message to this effect is displayed (Figure 1-22 on page 1-142). If the **stack** parameter is not specified, both a stack dump if available and the output of a list of **show** commands is generated.

The **full** parameter extends the list of **show** commands as described in the second half of Table 1-19.

Table 1-19: The list of **show** commands that are executed by the **show debug** command, when the **full** parameter is or is not specified, under different combinations of security mode and privilege level

full parameter specified?	security mode	privilege level	list of commands executed
No	normal	manager	show system show files show install show feature show release show config dynamic show buffer scan show cpu show log show exception show ffile check
No	secure	security officer	
No	secure	manager	show system (without current configuration file) show files show install show release show buffer scan show cpu show log show exception show ffile check

Table 1-19: The list of **show** commands that are executed by the **show debug** command, when the **full** parameter is or is not specified, under different combinations of security mode and privilege level (continued)

full parameter specified?	security mode	privilege level	list of commands executed
Yes	normal	manager	show system show files show install show feature show release show config dynamic show interface show ip interface show ip arp show ip route full show ip count show switch show switch counter show switch fdb (not on AR410 or AR410S) show startup show flash show switch port=all show switch port=all counter (not on AR410 or AR410S) show buffer scan show cpu show log show exception show ffile check
Yes	secure	security officer	
Yes	secure	manager	show system (without current configuration file) show files show install show release show interface show ip interface show ip arp show ip route full show ip count show switch show switch counter show switch fdb (not on AR410 or AR410S) show startup show flash show switch port=all show switch port=all counter (not on AR410 or AR410S) show buffer scan show cpu show log show exception show ffile check

Figure 1-21: Sample output from the **show debug stack** command after a software reboot

```
This is a production version of code
------------------------------------------------------

Router RESTART occurred
Check exception table for restart cause

STACK DUMP
------------------------------------------------------

00012830: 00000001 00000001 00000001 00000001
00012840: 00010000 00000001 00000010 00000000
00012850: 0004c300 004c29f0 0001289c 0000e9a8
00012860: 0000e990 004bea9c 0001287c 00012004
00012870: 20040005 19c20084 00000000 000128d8
00012880: 00090c58 00000000 00090c2c 00000010
00012890: 0000e990 00000000 002aa284 004b0318
000128a0: 00000000 00000000 00000000 00000001
000128b0: 00000001 002b2660 0001294c 000128d4
000128c0: 004bea9c 0027c164 004bea9c 002b2850
000128d0: 000128d8 002b2850 004b030a 00000007
000128e0: 00000000 00000000 00000000 00000010
000128f0: 00000001 00000483 004b029c 00000000
00012900: 004bea9c 07400000 0009bcd6 004bea9c
00012910: 002b2660 0001294c 004b030a 0000003f
00012920: 00317567 00000fd5 00000023 00000014
00012930: 00000001 00000022 00317571 00000010
00012940: 00000000 00317572 004b030a 00287170
00012950: 0047c29c 0000030c 00000000 00000010
00012960: 00400100 00000006 00000000 000115a4
00012970: 000115a8 004b029c 0009bb78 00000010
00012980: 004b029c 00000000
```

Figure 1-22: Sample output from the **show debug stack** command after a hardware reset.

```
This is a production version of code
------------------------------------------------------

Router hardware reset occurred - no debug info
```

Related Commands

show exception
show log
show startup
show system

show exception

Syntax `SHow EXception`

Description This command displays the router exception list (Figure 1-23 on page 1-143).

There may be up to ten entries in the list, ordered from most recent (event 01) to least recent (event 10). The explicit format of each entry depends on the exception type and hence what information was stored for that event.

The Spurious Interrupts field is the number of spurious interrupts handled by the router since startup. Under normal operating conditions this field should always be zero (0).

The fatal trap with error code of $001e is a CPU software trap that is invoked in response to the **restart** command on page 1-109 and hence should not be considered an error.

Figure 1-23: Example output from the **show exception** command

```
Spurious interrupts = 0

Router exception list
------------------------------------------------------------------------------
No: 01
  Offset/Type : $008/Bus error            Address    : $0019aaee
  Time        : 09:17:19 on 10-May-1997   Clock Log  : 09:16:42 on 10-May-1997
  SSW         : $0225                     Fault Addr : $0d0a0044

No: 02
  Offset/Type : $008/Bus error            Address    : $0019aaee
  Time        : 09:15:26 on 10-May-1997   Clock Log  : 09:14:29 on 10-May-1997
  SSW         : $0225                     Fault Addr : $0d0a0044

No: 03
  Offset/Type : $028/Line A emulator      Address    : $0009624c
  Time        : 10:42:59 on 01-May-1997   Clock Log  : 10:41:22 on 01-May-1997

No: 04
  Offset/Type : $028/Line A emulator      Address    : $0009624c
  Time        : 10:42:59 on 01-May-1997   Clock Log  : 10:41:22 on 01-May-1997

No: 05
  Offset/Type : $028/Line A emulator      Address    : $0009624c
  Time        : 10:42:59 on 01-May-1997   Clock Log  : 10:41:22 on 01-May-1997

No: 06
  Offset/Type : $028/Line A emulator      Address    : $0009624c
  Time        : 10:42:59 on 01-May-1997   Clock Log  : 10:41:22 on 01-May-1997

------------------------------------------------------------------------------
```

show feature

Syntax `SHow FEAture[={`*`featurename`*`|`*`index`*`}]`

where:

- *featurename* is a character string 1 to 12 characters long. Valid characters are any printable character.
- *index* is a decimal number in the range 1 to the number of special feature licences.

Description This command displays information about the special feature licences in the router. If a specific feature or index is not entered, summary information about all special feature licences is displayed (Figure 1-24 on page 1-144, Table 1-20 on page 1-144). If a special feature licence name or index is specified, detailed information about it is displayed (Figure 1-25 on page 1-145, Table 1-21 on page 1-145). This command can be issued only by a user with Security Officer privilege.

Figure 1-24: Example output from the **show feature** command

```
The Special Feature licences

Index  FeatureName   Licence       Period
-----------------------------------------------------------
1      ENCO          Full          -
2      Test          30 day Trial  16 aug 1998- 16 sep 1998
3       est2         password incorrect

The current valid features:

Triple DES Encryption
SW Compression
```

Table 1-20: Parameters in the output of the **show feature** command

Parameter	Meaning
Index	Index number for this special feature licence.
FeatureName	Name assigned to the special feature licence with the **enable feature** command on page 1-88.
Licence	Type of licence; either Full, 30 day Trial, or password incorrect if an invalid password has been specified with the **enable feature** command on page 1-88.
Period	Period when the licence is valid; either a date range for a 30-day trial licence or “ - ” for a full licence.
The current valid features	List of the special features enabled by this licence.

Figure 1-25: Example output from the **show feature** command for a specified special feature licence

```
The special feature licence : ENCO
Licence Type                : full
Period                      : -


The included features       : 3des Encryption
```

Table 1-21: Parameters in the output of the **show feature** command for a specified special feature licence

Parameter	Meaning
The special feature licence	Name assigned to the special feature licence with the **enable feature** command on page 1-88.
Licence Type	Type of licence; either Full, 30 day Trial, or Password Incorrect if an invalid password has been specified with the **enable feature** command on page 1-88.
Period	Period when the licence is valid; either a date range for a 30-day trial licence or " - " for a full licence.
The included features	List of the special features enabled by this licence.

Examples To display a list of all special feature licences, use the command:

```
sh fea
```

To display detailed information about special feature licence "Triple DES", use the command:

```
sh fea="Triple DES"
```

Related Commands **disable feature**
enable feature

show ffile

Syntax `SHow FFile[=`*`file-identifier`*`] [CHECK]`

where *file-identifier* is a valid FFS file identifier in the format `module\filename.ext`. Invalid characters are * + = “| \ [ ] ; : ? / , < >, and wildcards are allowed in any of the elements. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ’ ! @ # $ % ^ & ( ) _ - { }

Description This command displays a list of the files in the Flash File System (FFS) that match the specified file identifier (Figure 1-26 on page 1-146, Table 1-22 on page 1-146). If a file identifier is not specified, all files are displayed. Wildcards can be used to replace any part of the file identifier to allow a more selective display.

The **check** parameter specifies that the file data checksums are to be verified. Output with this parameter may take a number of seconds to complete for larger files.

Figure 1-26: Example output from the **show ffile** command

```
module   name      type    size     file date & time        address        check
---------------------------------------------------------------------------------
        ops        cfg     2610     18-Feb-2003 03:50:12    FECD734C         -
        help       hlp     94790    21-Jan-2003 07:57:41    FECC005C         -
        config     ins     32       03-Mar-2003 10:24:43    FEB05DC0         -
        gui        ins     64       19-Feb-2003 05:41:52    FECD7EDC         -
        prefer     ins     64       28-Feb-2003 06:08:59    FEADD1B4         -
        longname   lfn     60       18-Feb-2003 03:54:54    FECD7E60         -
        feature    lic     39       21-Jan-2003 07:57:59    FECD72E4         -
        random     rnd     3904     03-Mar-2003 10:44:43    FEB05E20         -
        d_410e00   rsc     2449712  19-Feb-2003 09:09:09    FECD7F5C         -
inst    release    lic     96       18-Feb-2003 03:54:09    FECD7DC0         -
load    melistst   paz     6108     03-Mar-2003 10:24:09    FEB045A4         -
load    52-251     rez     2795756  28-Feb-2003 05:59:36    FE82F12C         -

-----------------------------------------------------------------------------
flash use:
    files ......   5354100 bytes  (12 files)
    garbage ....    178988 bytes
    free .......   1675872 bytes
    block size .    131072 bytes
    total ......   7340032 bytes
-----------------------------------------------------------------------------
```

Table 1-22: Parameters in the output of the **show ffile** command

Parameter	Meaning
module	Module that created the file.
name	File name.
type	File type.
size	Size of the file in bytes, as a decimal number.
file date & time	Date and time the file was created.

Table 1-22: Parameters in the output of the **show ffile** command (continued)

Parameter	Meaning
address	Base address of the file, in hexadecimal.
check	Result of the file data check (if CHECK was specified).
files	Number of bytes of flash memory used by valid files.
garbage	Number of bytes of flash memory used by deleted files.
free	Number of bytes of flash memory free.
total	Total size of flash memory.

Examples To display all the release files created by the Loader module, use the command:

```
sh ff=load\*.rez
```

Related Commands show file

show file

Syntax

```
SHow FIle[=filename] [DEvice={ALl|FLash|NVs}]
```

where *filename* is a file identifier in the format `[device:]name.ext`. Invalid characters are " \ ; ? / , <. Valid characters are:

- uppercase and lowercase letters
- digits (0–9)
- the characters ~ ' ! @ # $ % ^ & ( ) _ - { } * > [ ] | :

Wildcard characters * may appear anywhere in the filename when displaying them, (not when creating them). The wildcard character matches any string.

Character ranges may be specified using the > character, for example a>z matches any letter in the alphabet. The + character may be used to specify a list of options, for example a*.scp+b*.scp would specify files that match a*.scp or b*.scp.

Square brackets may be used, for example ppp*.[scp+cfg] matches scripts and configuration files whose names start with "ppp".

The vertical bar | character matches any single character. For example, |||.scp matches script files with names three characters long (excluding extension and device name).

If a colon is seen anywhere in the filename, the device parameter is ignored and it is assumed that the filename includes the device name.

Description

This command displays a list of the files in the file subsystem that match the specified file name (Figure 1-27 on page 1-149, Table 1-23 on page 1-149). Wildcards can be used to replace any part of the file identifier to allow a more selective display. If the file name matches an explicit file and the file is an ASCII text file, the contents of the file are displayed.

The **device** parameter specifies the physical memory device where the file is stored.

To display the contents of the translation table, which converts filenames between DOS 28.3 format and DOS 8.3 format, use the **show file=longfile.lfn** command (Figure 1-28 on page 1-149 and Table 1-24 on page 1-149).

Figure 1-27: Example output from the **show file** command

```
Filename              Device        Size     Created               Locks
-------------------------------------------------------------------------
12345678901234567890.scp
                        flash         24       29-Mar-2004 15:34:21    0
13gggggg.scp            flash         8        29-Mar-2004 15:34:03    0
16a.scp                 flash         7        17-Mar-2004 10:50:33    0
16abcd.scp              flash         14       17-Mar-2004 10:21:24    0
16ffff.scp              flash         32       16-Mar-2004 13:41:26    0
16ffffff.scp            flash         8        16-Mar-2004 14:17:19    0
409275.scp              flash         507      03-Nov-2003 12:07:37    0
409275a.scp             flash         441      24-Oct-2003 12:23:04    0
409451.scp              flash         588      10-Nov-2003 10:17:18    0
86263aka.rez            flash         3604528  16-Apr-2004 14:20:46    0
atobrsa.key             flash         321      04-Feb-2004 14:32:51    0
basic.cfg               flash         119      01-Dec-2003 15:35:56    0
bgp.cfg                 flash         2811     15-Apr-2004 10:22:40    0
bgppeer.scp             flash         35       16-Apr-2004 09:59:20    0
cck.scp                 flash         1018     14-Oct-2003 15:27:57    0
client.cfg              flash         2679     06-Nov-2003 13:38:48    0
config.ins              flash         32       19-Apr-2004 12:07:50    0
--------------------------------------------------------------------------
```

Table 1-23: Parameters in the output of the **show file** command

Parameter	Meaning
Filename	Name of the file.
Device	Device where the file is physically stored; flash.
Size	Size of the file in bytes, as a decimal number.
Created	Date and time the file was created.
Locks	Number of concurrent processes using the file.

Figure 1-28: Example output from the **show file=longfile.lfn** command

```
short filename  device  long filename               created      size   check
--------------------------------------------------------------------------------
123456~0.scp    flash   12345678901234567890.scp    15:34:21     24          0
--------------------------------------------------------------------------------
```

Table 1-24: Parameters in the output of the **show file=longfile.lfn** command

Parameter	Meaning
Short filename	Name of the file in DOS 8.3 format.
Device	Device where the file is physically stored; flash.
Long filename	Name of the file in DOS 28.3 format.
Created	Date and time the file was created.
Size	Size of the file in bytes, as a decimal number.
Check	For flash files this value is 0 and not used.

Examples To display all the patch files on the router, use the command:

```
sh fi=*:*.paz
```

To display the contents of the config.scp script file, use the command:

```
sh fi=config.scp
```

To display the contents of the longfile.lfn long filename table, use the command:

```
sh fi=longfile.lfn
```

Related Commands delete file
purge file translationtable

show flash

Syntax `SHow FLash [FFs]`

Description This command displays general status information about the Flash File System (FFS). The FFS provides a consistent file-based interface to the physical flash memory structure, and housekeeping and management functions (Figure 1-29 on page 1-150, Table 1-25 on page 1-151).

Figure 1-29: Example output from the **show flash** command

```
FFS info:
global operation ...... none
compaction count ...... 256
est compaction time ... 88 seconds
files ................   1420044 bytes  (4 files)
garbage ..............     19652 bytes
free .................    526384 bytes
required free block ...   131072 bytes
total ................   2097152 bytes

diagnostic counters:
event      successes        failures
--------------------------------------
get                0              0
open               0              1
read               0              0
close              0              0
complete           0              0
write              0              0
create             0              0
put                0              0
delete             0              0
check              0              0
erase              0              0
compact            0              0
verify             0              0
--------------------------------------
```

Table 1-25: Parameters in the output of the **show flash** command

Parameter	Meaning
global operation	Global operation currently running; either none, restarting, erasing, compacting, or verifying.
compaction count	Number of times the flash has been compacted since the last total erasure.
est compaction time	Estimate of how long compaction would take if it was started now.
files	Amount of space used by valid files.
garbage	Amount of space used by deleted files.
free	Amount of free space.
required free block	Minimum contiguous working space. This amount of flash memory must remain available. Therefore, it is not included in the "free" entry.
total	Total flash size.
diagnostic counters	Counts of the successes and failures for each type of FFS operation.

FFS failure counts do not necessarily mean that an error has occurred, but are also incremented if the file specified could not be found. For example, attempting to delete a file that does not exist results in the delete failures count being incremented.

Related Commands

activate flash compaction
show flash physical

show flash physical

Syntax `SHow FLash PHYSICAL`

Description This command displays physical information about the specific type of flash installed in the router (Figure 1-30, Table 1-26).

Figure 1-30: Example output from the **show flash physical** command

```
total size ........... 16 MBytes
   available to FFS ... 15 MBytes
   available to boot .. 1 MBytes
device type .......... 28F128
devices .............. 1
location ............. built in
programming power ..... off
block erase time ...... 1000 milliseconds
total erase blocks .... 128
   FFS erase blocks ... 120
   Boot erase blocks .. 8
erase block size ...... 128 kBytes
erase bit state ....... 1
page buffers .......... 1
size of page buffer ... 32 bytes
```

Table 1-26: Parameters in the output of the **show flash physical** command

Parameter	Meaning
total size	Amount of flash memory installed.
available to FFS	Amount of flash memory available to the Flash Filing System.
available to boot	Amount of flash memory available to the boot flash.
device type	Type of flash device installed.
devices	Number of flash devices installed.
location	Whether flash memory is built in or a SIMM stick.
programming power	Whether programming power is on or off.
block erase time	Time taken to erase an erase block.
total erase blocks	Number of erase blocks.
FFS erase blocks	Number of erase blocks available to the Flash Filing System.
Boot erase blocks	Number of erase blocks available to the Boot system.
erase block size	Size of each erase block, in bytes.
erase bit state	State of an erased bit.
page buffers	Number of page buffers.
size of page buffer	Byte size of each page buffer.

Related Commands show flash

show gui

Syntax SHow GUI

Description This command displays information about the GUI status and the GUI resource file. The resource file contains the HTML pages that make up the GUI (Figure 1-31, Table 1-27 on page 1-153).

Figure 1-31: Example output from the **show gui** command

```
GUI Configuration
-------------------
Module Status         : Enabled

Resource File
-----------------------
Name                  : s_sb8e01.rsc
Status                : Good

Header Info
-----------------------
Type                  : Switch
Model                 : Switchblade 4000
Gui Builder Version   : 2.1
Language              : English
Version               : 01
File Creation Date    : 5/4/2002
Build Type            : PRODUCTION
File Size             : 1260309
```

Table 1-27: Parameters in the output of the **show gui** command

Parameter	Meaning
Module Status	Whether the GUI is enabled or disabled.
Name	Filename of the GUI resource file.
Status	State of the resource file; either Good (no errors in the file) or Error. If the state is Error, a line is displayed below the status indicating the nature of the error.
Type	Type of GUI.
Model	The model the resource file has been produced to run on. Resource files are model-dependent, so this must be the same model as the router.
GUI Builder Version	Version of the Allied Telesyn GUI creation program that this resource file was built with.
Language	Language in which the GUI is displayed.
Version	Version of the GUI.
File Creation Date	Date in day/month/year format when the resource file was created.
Build Type	The status of this build. "Production" indicates a build that has been released for use.
File size	Byte size of the resource file.

Example To display information about the GUI, use the command:

```
sh gui
```

Related Commands disable gui
enable gui
reset gui

show http client

Syntax

```
SHow HTTP CLIent
```

Description This command displays the current state of the HTTP client (Figure 1-32 on page 1-154, Table 1-28 on page 1-154).

Figure 1-32: Example output from the **show http client** command

```
HTTP Client
------------------------------------------------------------
  Sessions opened .............. 1
  Sessions closed .............. 1
  Transmitted requests ......... 1
  Received replies ............. 1
------------------------------------------------------------
```

Table 1-28: Parameters in the output of the **show http client** command

Parameter	Meaning
Sessions opened	Number of HTTP client sessions that have been started.
Sessions closed	Number of HTTP client sessions that have been closed.
Transmitted requests	Number of HTTP GET and POST requests transmitted by the client.
Received replies	Number of HTTP responses received by the client.

Examples To display the current status of the HTTP client, use the command:

```
sh http cli
```

Related Commands set http server
show http client
show http debug
show http server
show http server session

show http debug

Syntax `SHow HTTP DEBug`

Description This command displays the debugging options currently enabled for the HTTP server (Figure 1-33 on page 1-155, Table 1-29 on page 1-155).

Figure 1-33: Example output from the **show http debug** command

```
Enabled Debug Modes
-------------------------------------------------------
AUTH,MSG
-------------------------------------------------------
Enabled Debug Modes
-------------------------------------------------------
  AUTH,MSG
-------------------------------------------------------
```

Table 1-29: Parameter in the output of the **show http debug** command

Parameter	Meaning
Enabled Debug Modes	Debugging modes currently enabled for the HTTP server: NONE, AUTH, MSG, SESSION, or ALL.

Examples To display the currently enabled debugging modes for the HTTP server, use the command:

```
sh http deb
```

Related Commands

disable http debug
enable http debug
show http client
show http server
show http server session

show http server

Syntax `SHow HTTP SERVer`

Description This command displays configuration and status information for the HTTP server (Figure 1-34 on page 1-156, Table 1-30 on page 1-156).

Figure 1-34: Example output from the **show http server** command

```
HTTP Server
------------------------------------------------------------
  Status ................................ Enabled
  SSL Security ................. OFF
  SSL Key ID ................... -
  Port .................................. 80
  Listen port ........................... Open

  Sessions opened ....................... 12
  Sessions closed ....................... 12
  Received requests ..................... 205
  Unknown requests ...................... 0
  Transmitted replies ................... 205
  Aborted replies ....................... 0
  Transmitted replies on bad session .... 0
  Authorisation successes ............... 202
  Authorisation failures ................ 3
------------------------------------------------------------
```

Table 1-30: Parameters in the output of the **show http server** command

Parameter	Meaning
Status	Whether the HTTP server is enabled.
SSL Security	Whether the HTTP server is enabled for SSL secured connections. If ON, the HTTP server accepts SSL secured connections; and if OFF, the HTTP server accepts connections not secured with SSL.
SSL Key ID	Identification number for the private key used for encryption.
Port	TCP port that the HTTP server is listening on.
Listen port	Whether the HTTP server’s TCP listen port is open or closed.
Sessions opened	Number of HTTP server sessions that have been started.
Sessions closed	Number of HTTP server sessions that have been closed.
Received requests	Number of HTTP GET and POST requests received by the server.
Unknown requests	Number of unrecognised HTTP requests received by the server
Transmitted replies	Number of HTTP responses transmitted by the server.
Aborted replies	Number of HTTP replies aborted by the server.
Transmitted replies on bad session	Number of HTTP replies transmitted by the server for bad sessions.
Authorisation successes	Number of successful HTTP authorisations.
Authorisation failures	Number of failed HTTP authorisations.

Examples To display the current status of the HTTP server, use the command:

```
sh http serv
```

Related Commands

disable http server
enable http server
set http server
show http client
show http server session

show http server session

Syntax `SHow HTTP SERVer SESSion`

Description This command displays TCP session information for the HTTP server (Figure 1-35 on page 1-157, Table 1-31 on page 1-157).

Figure 1-35: Example output from the **show http session** command

```
Client IP       Interface  Current User       State
------------------------------------------------------------
127.0.0.1      eth0       manager           RECEIVING_REQ
127.0.0.1      eth0       manager           RECEIVING_REQ
------------------------------------------------------------
```

Table 1-31: Parameters in the output of the **show http server session** command

Parameter	Meaning
Client IP	IP address of the client using the session.
Interface	IP interface through which the client session is running.
Current User	User name used to authenticate the session.
State	Current state of the HTTP server session: AWAITING_REQ PROC_KEEPUP_REQ PROC_CLOSE_REQ RECEIVING_REQ CLOSING

Examples To display TCP session information for the HTTP server, use the command:

```
sh http sess
```

Related Commands

set http server
show http client
show http debug
show http server

show install

Syntax `SHow INSTall`

Description This command shows install information, the install that the router is currently running, and the history of checking install information at boot. This information includes the release file, GUI resource file and patch file used (Figure 1-36 on page 1-158, Table 1-32 on page 1-158).

If the selected GUI resource file fails to pass validation checks on boot-up, described under the **set install** command on page 1-113, the given install does not fail. Instead, the release and patch files are installed, but the GUI resource file is not installed. The success or failure of this validation is recorded in the "install history" section of the command output.

Figure 1-36: Example output from the **show install** command after a new release file is installed

```
Install     Release              Patch        GUI
-----------------------------------------------------------
Temporary   -                    -            -
Preferred   flash:52-240g.rez    -            d_sb8e00.rsc
Default     EPROM (PR1-1.1.0)    -            -
-----------------------------------------------------------

Current install
------------------------------------------------------------
---
Preferred   flash:52-240g.rez    -            d_sb8e00.rsc
-----------------------------------------------------------

Install history
-----------------------------------------------------------
No Temporary release selected
Preferred release selected
Preferred release successfully installed
Preferred GUI successfully installed
-----------------------------------------------------------
```

Table 1-32: Parameters in the output of the **show install** command

Parameter	Meaning
Install	Type of install: Temporary, Preferred, or Default.
Release	Release file for the install.
Patch	The patch file for the install.
GUI	The resource file installed on the router. The filename is displayed independently of whether the GUI is enabled.
Dmp	The third party Data Manipulation Program for the install. This is not present on most models and software releases.
Current install	The install currently running in the router.
Install history	List of checks that the Install module carried out during the install boot. This list shows how the current install came to be selected and loaded.

Related Commands delete install
set install

show ldap

Syntax `SHow LDAP [DEBug]`

Description This command summarises information about the LDAP module (Figure 1-37 on page 1-159, Table 1-33 on page 1-159).

If **debug** is specified, debug status for the LDAP module is displayed.

Figure 1-37: Example output from the **show ldap** command

```
LDAP Module Information:
  Number of outstanding requests: 2

  Open Request Summary:
    Request ID ...... 2
      Level ......... Top Level
      Status ........ BINDING TO SERVER
    Request ID ...... 1
      Level ......... Top Level
      Status ........ BINDING TO SERVER

LDAP module trace debugging:
  Current Status .... DISABLED
  Debug Device ...... 16
```

Table 1-33: Parameters in the output of the **show ldap** command

Parameter	Meaning
Number of outstanding requests	Number of currently active requests in the LDAP module database.
Request ID	ID allocated to the request by the LDAP module.
Level	Level where the request was initiated: Top Level - the request was initiated from outside of the module (by the user or another module) Subordinate - the request was generated internally by the LDAP module
Status	Current status of the request in progress: BINDING TO SERVER - attempting to establish a connection to the LDAP server WAITING FOR RESULT - waiting for the server to send the results of the requested operation ABANDONED - the operation has been abandoned by the original requester
Debugging Current Status	Status of module trace debugging; either Enabled or Disabled.
Debug Device	Device last or currently receiving debug information.

Examples To show the current state of the LDAP module, use the command:

```
sh ldap
```

Related Commands show ldap request

show ldap request

Syntax `SHow LDAP REQuest[={ALL|number}]`

where *number* is the request identification number of an open request

Description This command displays information about LDAP requests (Figure 1-38 on page 1-160, Table 1-34 on page 1-160). If the **request** parameter is specified with the identification number of an open request, information is displayed for the specific request.

Figure 1-38: Example output from the **show ldap request** command

```
Show all LDAP Requests:
Info for Request ID 1:
  Schema ............ PKI
  Operation ......... Read
  Request Level ..... Top Level
  Request Status .... BINDING TO SERVER
  Host IP/Port ...... 192.168.3.4:389
  BindDN/User .......
  Password ..........
  Base Object DN .... cn=Joe Blobbs,dc=blobby,dc=com
  Scope ............. Base Object Only
  Return Objects .... userCertificate
  Get Names Only .... False
  Search Filter ..... (objectclass=*)
```

Table 1-34: Parameters in the output of the **show ldap request** command

Parameter	Meaning
Schema	LDAP Schema under which the request was made.
Operation	Type of operation requested under the schema; either Read or Search.
Request Level	Level where the request was initiated: Top Level - the request was initiated from outside of the module (by the user or another module) Subordinate - the request was generated internally by the LDAP module

Table 1-34: Parameters in the output of the **show ldap request** command (continued)

Parameter	Meaning
Request Status	Current status of the request in progress: BINDING TO SERVER - attempting to establish a connection to the LDAP server WAITING FOR RESULT - waiting for the server to send the results of the requested operation) ABANDONED - the operation has been abandoned by the original requester)
Host IP/Port	IP address and port of the LDAP server.
BindDN/User	Server authentication username.
Password	Server authentication password.
Base Object DN	Base object for the requested LDAP operation; a distinguished name in the format shown in *"Distinguished Names (DN)" on page 1-50*.
Scope	Scope of objects in the X.500-like directory to which the operation should apply: Base Object Only Single Level Whole Subtree
Return Objects	Type of object/s to be returned as a result of a read or search operation.
Get Names Only	Whether the objects' names are returned (True) or their values (False) also.
Search Filter	LDAP filter for the operation.

Examples To show LDAP requests in detail, use the command:

```
sh ldap req
```

Related Commands **show ldap**

show loader

Syntax `SHow LOAder`

Description This command displays the defaults for the Loader module and the progress of the current load (Figure 1-39 on page 1-162, Table 1-35 on page 1-162).

Figure 1-39: Example output from the **show loader** command

```
Loader Information
------------------------------------------------------------------------------
Defaults:
Method.............. TFTP
File ............... /netupgrades/new.cfg
Server ............. tftp.company.com (192.168.1.1)
HTTP Proxy ......... -
Proxy Port ......... Default ( 80 )
Asyn ............... -
Destination ........ Flash
Delay (sec) ........ 0

Current Load:
Method.............. HTTP
File ............... myserver/newreleasefiles/releaseupgrades/mycurrentproducts
                     /netupgrades/8-200gui.rez
Server ............. www.company.com (192.168.163.22)
TCP Port ........... 80
Destination .......  Flash
Delay (sec) ........ 0
Status ............. Loading
Load Level ......... 0%
------------------------------------------------------------------------------
```

Table 1-35: Parameters in the output of the **show loader** command

Parameter	Meaning
Defaults	This section lists the defaults used for parameters not specified in the **load** and **upload** commands.
Current Load	This section lists the values currently being used to load a file to or from the router.
Last Load	This section lists the values last used to load a file to or from the router.
Method	Method used to load files: TFTP HTTP WEB WWW ZMODEM NONE
File	Name of the file to be loaded.
Server	IP address or host name of the server. Used when METHOD is set to TFTP or HTTP.

Table 1-35: Parameters in the output of the **show loader** command

Parameter	**Meaning**
HTTP Proxy	IP address or host name of the proxy server when METHOD is set to HTTP and access is via a proxy server.
Last Message	Last error or informational message sent to the device where the last **load** command on page 1-96 was issued. At router boot, the Last Message is undefined and shows as " - ". This is not displayed when the Loader status is "Loading".

Related Commands

load
set loader
upload

show mail

Syntax `SHow MAIL`

Description This command displays the current configuration of the email system, and any email messages that are currently queued for transmission (Figure 1-40 on page 1-163, Table 1-36 on page 1-163).

Figure 1-40: Example output from the **show mail** command

```
MAIL
  Host Name ............ router2.company.com
  SMTP Server .......... 192.68.6.100
  State ................ alive
  Debug ................ disabled
  Mails Sent ........... 0

Date/Time   Id    To                   Subject              State      Retries
------------------------------------------------------------------------------
29 15:00:05 0002  jb@it.company.com    Test Message         Open       0
------------------------------------------------------------------------------
```

Table 1-36: Parameters in the output of the **show mail** command

Parameter	**Meaning**
Host Name	Host name used by the mail system.
SMTP Server	IP address of the SMTP mail server where mail is sent or "Not Set".
State	State of the mail system: Alive DEAD - name server not set DEAD - hostname not set
Debug	Whether debugging is enabled for the mail system.
Mails Sent	Number of mail messages transmitted since the last router restart.

Table 1-36: Parameters in the output of the **show mail** command

Parameter	Meaning
Date/Time	Date and time the message was queued for transmission.
Id	Unique message id for the message.
To	Email address where the message is to be sent.
Subject	Contents of the subject field in the message header.
State	State of the transmission process: initial - Starting get MX-IP - Performing DNS lookup on MX record get IP - Performing DNS lookup Connect - TCP connection established S-helo - Sending HELO command S-from - Sending MAIL FROM command S-rcpt - Sending RCPT TO command S-data - Sending DATA command S-header - Sending headers S-file - Sending file S-buffer - Sending message text S-last - Sending dot to terminate message S-done - Sending message transmission S-quit - Sending QUIT command
Retries	Number of times the mail system re-transmitted the message because an acknowledgement was not received from the remote mail system.

Examples

To show the state of the email system, use the command:

```
sh mail
```

Related Commands

delete mail
disable mail debug
enable mail debug
mail
show mail

show manager asyn

Syntax

```
SHow MAnager ASYn
```

Description

This command displays the port number of the current semipermanent manager port. There can be only one semipermanent manager port at a time. When a semipermanent manager port is defined, the following message is displayed:

```
The manager port is ASYN 0
```

When no semipermanent manager port is defined, the following message is displayed:

```
No manager port is defined.
```

Related Commands

login
set manager asyn
set asyn

show patch

Syntax

```
SHow PATch
```

Description

This command displays all patch files stored in NVS and flash (Figure 1-41, Table 1-37 on page 1-165).

Figure 1-41: Example output from the **show patch** command

```
Patch files
Name             Device     Size      Version
---------------------------------------------
28-74.pat        flash      376032    7.4.0-11
28760-02.paz     flash      109644    7.6.0-02
---------------------------------------------
```

Table 1-37: Parameters in the output of the **show patch** command.

Parameter	Meaning
Name	Name of the patch file.
Device	Whether the device where the patch is physically stored is flash or NVS.
Size	Size of the patch file in bytes expressed as a decimal number.
Version	Version number of the patch, consisting of the version number of the release to which the patch applies, followed by a hyphen, and the generation number of the patch itself.

Related Commands

load

show radius

Syntax `SHow RADius`

Description This command displays the list of known RADIUS servers (Figure 1-42, Table 1-38 on page 1-166). RADIUS servers are used for user authentication.

Figure 1-42: Example output from the **show radius** command

```
Server           Port  AccPort  Secret LocalInterface
------------------------------------------------------
192.168.17.11    1645     1646  ****** local4
172.31.253.9     1645        0  ****** Not set
------------------------------------------------------
```

Table 1-38: Parameters in the output of the **show radius** command

Parameter	Meaning
Server	IP address of this RADIUS server.
Port	Port number used to communicate with the RADIUS authentication server.
AccPort	Port number used to communicate with the RADIUS accounting server.
Secret	Shared secret used in communications between the router and the RADIUS server. Asterisks are displayed to prevent accidental discovery by unauthorised users.
Passcode prompt	Status of the passcode prompt generation; either On or Off.
Local Interface	Interface used as the source in outgoing messages to the RADIUS server.

Examples To displays the list of known RADIUS servers, use the command:

```
sh rad
```

Related Commands **add radius server**
delete radius server

show radius debug

Syntax `SHow RADius DEBug`

Description This command shows the debugging options for all RADIUS servers (Figure 1-43, Table 1-39 on page 1-167).

Examples To display the debugging options enabled for RADIUS, use the command:

```
sh rad deb
```

Figure 1-43: Example output from the **show radius debug** command

```
RADIUS Server        Enabled Debug Modes
------------------------------------------------------------
All Servers          PKT, DECODE, ERROR
```

Table 1-39: Parameters in the output of the **show radius debug** command

Parameter	Meaning
RADIUS Server	Servers where debugging is enabled.
Enabled Debug Modes	Debugging modes enabled.

Related Commands **enable radius debug**
disable radius debug

show release

Syntax `SHow RELease`

Description This command shows the release licence information in the router (Figure 1-44, Table 1-40). All releases that have a licence are displayed, along with the status of the licence.

Figure 1-44: Example output from the **show release** command

```
Release                      Licence          Period
---------------------------------------------------------------------------------
flash:load\28-74ang.rel      full             -
flash:load\28-761.rel        30 day trial     10-May-1998 to 10-Jun-1998
---------------------------------------------------------------------------------
```

Table 1-40: Parameters in the output of the **show release** command

Parameter	Meaning
Release	Full name of the release file.
Licence	Licence type: Full or 30-day trial.
Period	Period of the licence when it is a 30-day trial.

Related Commands disable release
enable release

show skey

Syntax SHow SKEY [SEQuence=*seq_no* SEED=*seed_name* [NUMber=*value*]]

where:

- *seq_no* is an integer from 1 to 9999 representing the sequence number of the last S/Key or OTP password to be generated.
- *seed_name* is the 1-16 alphanumeric user-defined string that initialises the one-time password system on the authentication server.
- *value* is an integer from 1 to 99 representing the number of consecutive S/Key or OTP passwords to generate, finishing at *seq_no*.

Description This command shows the current S/Key configuration on the router (Figure 1-45 on page 1-169, Table 1-41 on page 1-169).

If the **sequence** and **seed** parameters are specified, the router calculates and displays one-time passwords for use during authentication when a user logs into the router using the S/Key or OTP system (Figure 1-46 on page 1-170, Table 1-42 on page 1-170).

To display the correct one-time passwords, the user must supply their current sequence number and seed. They are then asked to enter the password, which was used when initialising their current sequence of one-time passwords on the authentication server. The entered password is not echoed to the screen. The output shows the sequence of S/Key or OTP one-time passwords to be used for a user's subsequent login attempts.

Figure 1-45: Example output from the **show skey** command

```
Current S/Key Configuration
------------------------------------------------------------
   Password Calculation Method ..... SKEY
   Encryption Algorithm ............ MD4
------------------------------------------------------------
```

Table 1-41: Parameters in the output of the **show skey** command

Parameter	Meaning
Password Calculation Method	Method to calculate the password: SKEY or OTP.
Encryption Algorithm	Encryption method: MD4 or MD5.

Figure 1-46: Example output from the **show skey seq=n seed=seed** command

```
Enter S/KEY initialisation password :
Computing SKEY passwords using MD4....
-----------------------------------------------------------
Seq No        One-Time Password
95            IT DOLT ROOM NET GLUT ROWE
96            DARE MOS SARA GOAD MAO LEO
97            GUN TAIL MEND EAT INCH JOHN
98            EARN KID CARE HELD GIRD WINE
99            ADAM WARD DECK PLY EGAN WEED
-----------------------------------------------------------
```

Table 1-42: Parameters in the output of the **show skey seq=n seed=seed** command

Parameter	Meaning
Seq No	Sequence number of the S/Key password to be generated.
One-Time Password	S/Key passwords to be used when the user next logs in.

Examples To show the next five passwords to be used when logging into a router under S/key or OTP authentication control, where the current sequence number is 99 and the seed used to generate the sequence was hs12345, use the command:

```
sh skey seq=99 seed=hs12345 num=5
```

Related Commands set skey
login

show startup

Syntax `SHow STARTup`

Description This command prints the state of the bits in the router Startup Status Flag (Figure 1-47). This command can be used to check the state of the router when it last started up. When a bit signals an error, its message has an > appended to the front of it.

Figure 1-47: Example output from the **show startup** command

```
Router Startup Status Flag is 00600040, which means:
----------------------------------------------------
   4096k of RAM found
> Router CRASHED prior to this startup
  Battery backed RAM battery OK
  Battery backed RAM not corrupted
  Real time clock not corrupted
  Real time clock, time set
  Router software download OK
  Router vector download OK
----------------------------------------------------
```

show system

Syntax `SHow SYStem`

Description This command displays general system information about the router, including the hardware installed, memory, software release, and patches loaded (Figure 1-48 on page 1-171, Table 1-43 on page 1-172). It also displays location and contact details when these have been set with the appropriate **set system** commands.

Figure 1-48: Example output from the **show system** command

```
Router System Status                               Time 16:01:16 Date 19-Jun-2002.
Board      ID  Bay Board Name                      Rev   Serial number
--------------------------------------------------------------------------------
Base      176      AR410                           M3-0  42858945
PIC        75   0  AT-AR020-00 PIC E1/T1 PRI       M1-0  42197228
--------------------------------------------------------------------------------
Memory -   DRAM : 16384 kB    FLASH :  7168 kB
--------------------------------------------------------------------------------
SysDescription
Allied Telesyn AR410 version 2.5.1-00 10-Nov-2002
SysContact

SysLocation

SysName

SysDistName

SysUpTime
1252 ( 00:00:12 )
Boot Image      : 410-101.fbr size 903404 20-Aug-2002
Software Version: 2.5.1-00 10-Nov-2002
Release Version : 2.5.1-00 10-Nov-2002
Patch Installed : NONE
Territory       : usa
Country         : Canada
Help File       : help.hlp

Configuration
Boot configuration file: redeye.cfg (exists)
Current configuration: redeye.cfg

Security Mode   : Disabled

Patch files
Name            Device     Size      Version
--------------------------------------------
52251tst.paz    flash      87240     2.5-2
--------------------------------------------
```

Table 1-43: Parameters in the output of the **show system** command

Parameter	**Meaning**
Board	Possible board types: Base Expansion PAC NSM PIC MAC Uplink
ID	Identification number of the board. Board IDs can be larger than 99.
Bay	Bay number where the expansion board is installed
Board Name	Descriptive name of the board.
Rev	Revision number and hardware modification level of the board.
Serial number	Serial number of the board.
DRAM	Amount of DRAM memory installed.
FLASH	Amount of flash memory installed.
SysDescription	Description of the product and software release.
SysContact	A string specifying a contact name or address to call for the router. This is set with the **set system contact** command on page 1-121.
SysLocation	A string specifying the location of the router. This is set with the **set system location** command on page 1-123.
SysName	A string specifying the name (usually the complete IP domain name) of the router. This is set with the **set system name** command on page 1-124.
SysUpTime	Elapsed time in 100ths of a second since the last router restart.
Boot Image	Flash boot image file name, size, and when it was loaded onto the flash boot area, for devices that boot from flash.
Software Version	Patch version running on the router.
Release Version	Software release running on the router.
Patch Installed	Description of the patch currently installed, or “NONE” when no patch is installed.
Territory	Territory where the router is being used; either Australia, China, Europe, Japan, Korea, Newzealand, or USA. This can be set with the **set system territory** command on page 1-124.
Country	Country where the router is being used, for ATM default values. This can be set using the **set system country** command on page 1-122 of Chapter 1, Operation.
Help File	System help file used by the **help** command on page 1-95 for online help. This can be set with the **set help** command on page 1-111.
Boot configuration file	Current boot configuration file set with the **set config** command on page 1-110 and whether the file exists (Table 1-17 on page 1-134).

Table 1-43: Parameters in the output of the **show system** command (continued)

Parameter	Meaning
Current configuration	Source of the current router configuration. This can be one of a number of items, including a configuration file name, no configuration, or configuration set by DIP switches (Table 1-17 on page 1-134).
Security Mode	Whether security mode is enabled.
Patch files	Information about the patch files installed on the router, or the message "Warning (248283): No patches found.".
Name	Name of a patch file.
Device	Memory device where the patch file is stored; either NVS or flash.
Size	Size of the patch file in bytes.
Version	Version number of the patch, consisting of the version number of the release to which the patch applies, followed by a hyphen and the generation number of the patch itself.

Related Commands
disable system security_mode
enable system security_mode
set help
set system contact
set system location
set system name
set system territory

show system serialnumber

Syntax

```
SHow SYStem SErialnumber
```

Description This command displays the base hardware serial number of the router (Figure 1-49 on page 1-173, Table 1-44 on page 1-173).

Figure 1-49: Example output from the **show system serialnumber** command

```
56845218
```

Table 1-44: Parameters in the output of the **show system serialnumber** command

Parameter	Meaning
Base hardware serial number	Serial number of the router's base hardware.

Examples To display the router's base hardware serial number, use the command:

```
sh sys se
```

Related Commands **show system**

show tacacs debug

Syntax `SHow TACacs DEBug`

Description This command shows the debugging options for all TACACS servers (Figure 1-50, Table 1-45).

Examples To display the debugging options enabled for TACACS, use the command:

```
sh tac deb
```

Figure 1-50: Example output from the **show tacacs debug** command

```
TACACS Server        Enabled Debug Modes
------------------------------------------------------------
All Servers          PKT, DECODE, ERROR
```

Table 1-45: Parameters in the output of the **show tacacs debug** command

Parameter	Meaning
TACACS Server	Servers where debugging is enabled.
Enabled Debug Modes	Debugging modes enabled.

Related Commands **enable tacacs debug**
disable tacacs debug

show tacacs server

Syntax `SHow TACacs SERVER`

Description This command displays the list of TACACS servers used for authenticating login names (Figure 1-51, Table 1-46).

Figure 1-51: Example output from the **show tacacs server** command

```
TACACS server addresses  Passcode prompt
----------------------------------------
192.168.35.17            On
192.168.163.30           Off
----------------------------------------
```

Table 1-46: Parameters in the output of the **show tacacs server** command

Parameter	Meaning
TACACS server address	IP address of this TACACS server.
Passcode prompt	Status of the passcode prompt generation.

Related Commands add tacacs server
delete tacacs server

show tacplus key

Syntax `SHow TACPlus Key`

Description This command displays the global key for TACACS+.

Examples To show the TACACS+ global key, use the command:

```
sh tacp k
```

Figure 1-52: Example output from the **show tacplus key** command

```
Tacplus global key: thisIsTheCurrentGlobalTacplusKey
```

Related Commands set tacplus key
show tacplus server

show tacplus server

Syntax `SHow TACPlus SERVER`

Description This command displays the configured TACACS+ servers.

Example To show the TACACS+ servers currently configured, used the command:

```
sh tacp server
```

Figure 1-53: Example output from the **show tacplus server** command

```
Tacacs Plus Server Information
IP Address       Port    Timeout Value  Sessions  Single connection Local Interface
----------------------------------------------------------------------------------
172.168.198.254    49        5              1          Yes               local7
192.168.196.254    49        8              2          No                Not set
----------------------------------------------------------------------------------
```

Table 1-47: Parameters in the output of the **show tacplus server** command

Parameter	Meaning
IP Address	IP address of the TACACS+ server.
Port	TCP port being used.
Timeout Value	Length of time the router waits for a response from the TACACS+ server.
Sessions	Number of TACACS+ sessions for each server.
Single connection	Whether multiple TACACS+ sessions are supported.
Local Interface	Interface used as the source in outgoing TACACS + messages sent to the TACACS+ server.

Related Commands

add tacplus server
delete tacplus server
set tacplus server

show tacplus telnet

Syntax `SHow TACPlus TELnet`

Description This command displays the level of TACACS+ privilege that is currently required for using telnet on the router.

Figure 1-54: Example output from the **show tacplus telnet** command

```
TACACS+ telnet privilege level: NONE
```

Table 1-48: Parameters in the output of the **show tacplus telnet** command

Parameter	Meaning
TACACS+ telnet privilege level	Level of TACACS+ privilege required for using telnet on the router—a number from 0 to 15 or **none**. **None** indicates that no TACACS+ authenticated user can use telnet.

Related Commands set tacplus telnet

show tacplus user

Syntax `SHow TACPlus USer`

Description This command displays users who are currently being authenticated by TACACS+, or those who have been authenticated very recently. It does **not**

maintain a list of all currently logged-in users who were authenticated by a TACACS+ server.

Example

```
sh tacp us
```

Figure 1-55: Example output from the **show tacplus user** command

```
Tacacs Plus User Information
------------------------------------------------------------
User Name: admin
Privilege: manager        Login: 12:03:08

User Name: user1
Privilege: unknown        Login: not logged in
```

Table 1-49: Parameters in the output of the **show tacplus user** command

Parameter	Meaning
User Name	User’s login name.
Privilege	User’s privilege level.
Login	Time the user logged in, or “not logged in”.

Related Commands show tacplus server

show time

Syntax

```
SHow TIme
```

Description This command displays the current router time as maintained by the real-time clock. An example of the message is:

```
System time is 09:18:05 on 10-Jun-1997
```

Related Commands set time

show user

Syntax

```
SHow USEr[=login-name] [Configuration]
```

where *login-name* is a character string 1 to 64 characters long. Valid characters are uppercase and lowercase letters and decimal digits (0–9). The string cannot contain spaces.

Description

This command displays the contents of the User Authentication Database or global configuration parameters and counters for the User Authentication Facility.

For a user with Manager or Security Officer privilege, the command displays the contents of the User Authentication Database. When the router is in security mode, the command also displays the number of users currently logged in with Security Officer privilege. If a login name is specified, information for the specific user is displayed. If a login name is not specified, the entire database is displayed (Figure 1-56 on page 1-180, Table 1-50 on page 1-180). For someone with User privilege, parameters are not allowed and the their own database record is displayed.

The **configuration** parameter displays global configuration parameters and counters for the User Authentication Facility (Figure 1-57 on page 1-181, Table 1-51 on page 1-182). A login name is not valid with this parameter.

Figure 1-56: Example output from the **show user** command

```
Number of logged in Security Officers currently active ...1

User Authentication Database
--------------------------------------------------------------------------

Username: dave ()
   Status: enabled    Privilege: Sec Off   Telnet: yes   Login: yes
   Callback number: 0061393546786    Calling number: 5554491
   Logins: 2          Fails: 0        Sent: 0            Rcvd: 0
   Authentications: 0 Fails: 0
Username: manager (Manager Account)
   Status: enabled    Privilege: manager   Telnet: yes   Login: yes
   Logins: 4          Fails: 0        Sent: 0            Rcvd: 0
   Authentications: 0 Fails: 0
Username: tony ()
   Status: enabled    Privilege: user      Telnet: no    Login: no
   Ip address: 192.168.1.5      Netmask: 255.255.255.0   Mtu: 1500
   Logins: 0          Fails: 2        Sent: 0            Rcvd: 0
   Authentications: 0 Fails: 0
--------------------------------------------------------------------------


Active (logged in) Users
------------------------

User              Port/Device
    Login Time               Location
---------------------------------------------------------------------
manager           Asyn 0
    14:33:22 18-Apr-2002     local
manager           Telnet 1
    14:33:22 18-Apr-2002     10.1.1.1
---------------------------------------------------------------------
```

Table 1-50: Parameters in the output of the **show user** command

Parameter	Meaning
User Authentication Database	This section shows the contents of the User Authentication Database
Number of logged in Security Officers currently active	The number of users currently logged in with Security Officer privilege. This counter does not include users whose Security Officer privilege is disabled because they have not entered a security command within the secure delay period.
Username	Login name.
Status	Whether the entry is enabled.
Privilege	Privilege level for this user; either Sec Off, Manager, or User.
Telnet	Whether the user is permitted to use the **telnet** command to telnet to a host.
Login	Whether the user can log into the router.
Authentications	Number of authentications.
IP address	IP address for this user.
Netmask	Network mask for this user.
Mtu	MTU for this user.

Table 1-50: Parameters in the output of the **show user** command (continued)

Parameter	Meaning
IPX network	Novell network number assigned to the user. This field is not present if a network number has not been assigned.
Callback number	ISDN phone number for this user when making a call back to a remote user.
Calling number	Number to check against the incoming calling number of an L2TP or ISDN call when the call provides caller ID information.
Logins	Number of times a successful login has been made using this login name.
Fails	Number of times an incorrect password was given for this login name.
Sent	Number of octets sent by the user to the router.
Rcvd	Number of octets set to the user from the router.
Active (logged in) Users	This section summarises the users currently logged in.
user	Login name of the user.
Port/Device	Port or device on the router that the user is logged in to; either Port x, Telnet x, or SSH x, where *x* is the device instance.
Location	Location of the user. It is local if the user is attached to an asynchronous port or the IP address of the remote device.
Login Time	Time the user most recently logged in.

Figure 1-57: Example output from the **show user configuration** command

```
User Authentication Facility configuration and counters
--------------------------------------------------------------------------------
Security parameters
  login failures before lockout ............     4            (LOGINFAIL)
  lockout period ...........................    20 seconds    (LOCKOUTPD)
  manager password failures before logoff ..     3            (MANPWDFAIL)
  maximum security command interval ........    30 seconds    (SECUREDELAY)
  minimum password length ..................     6 characters (MINPWDLEN)
  TACACS retries ...........................     3            (TACRETRIES)
  TACACS timeout period ....................     5 seconds    (TACTIMEOUT)
  semi-permanent manager port ..............     0
  User Login ...............................     True
Security counters
  logins                         7        authentications               23
  databaseClearTotallys          0        managerPwdChanges             0
  defaultAcctRecoveries          0        unknownLoginNames             1
  tacacsLoginReqs                1        totalPwdFails                 5
  tacacsLoginRejs                1        managerPwdFails               3
  tacacsReqTimeouts              0        securityCmdLogoffs            1
  tacacsReqFails                 0        loginLockouts                 1
--------------------------------------------------------------------------------
```

Table 1-51: Parameters in the output of the **show user configuration** command

Parameter	Meaning
login failures before lockout	Default number of login failures allowed by a user before the login prompt is withheld for the lockout period.
lockout period	Default period in seconds that the login prompt is withheld from a user after a number of consecutive login failures.
manager password failures before logoff	Default number of successive failures a manager may make entering the login password before the session is logged off.
maximum security command interval	Default interval in seconds that may elapse between successive commands without the security officer being prompted to re-enter the login password.
minimum password length	Default for the minimum password length.
TACACS retries	Default number of times a TACACS request is retransmitted when a response is not received within the timeout period.
TACACS timeout period	Default in seconds that the router waits for a TACACS response before retransmitting the request.
semi-permanent manager port	Port number of the semipermanent manager port.
logins	Total number of logins by any user to the router.
authentications	Total number of authentications by a user, by the router.
managerPwdChanges	Number of times a manager privilege level password has been changed.
unknownLoginNames	Number of attempted logins with a login name that did not exist in the database and was not validated by a TACACS server.
totalPwdFails	Total number of times an incorrect password was given for a login name that exists in the database.
managerPwdFails	Number of times a manager was challenged to give their password for a security command and they entered the incorrect password.
securityCmdLogoffs	Number of times a manager was logged off because a correct password was not entered when required to validate a security command.
loginLockouts	Number of times the login lockout period was instigated because too many unsuccessful login attempts were made.
databaseClearTotallys	Number of times the database has been cleared.
defaultAcctRecoveries	Number of times the router was rebooted with DIP switch 3 set to restore the default account passwords.
tacacsLoginReqs	Number of login requests made to a TACACS server.
tacacsLoginRejs	Number of rejects received from a TACACS server in response to a login request.
tacacsReqTimeouts	Number of login requests to a TACACS server that terminated in a timeout.
tacacsReqFails	Number of login attempts terminated because of TACACS server timeouts.

Related Commands

- add user
- delete user
- disable system security_mode
- disable user
- enable system security_mode
- enable user rso
- purge user
- reset user
- set user

show user rso

Syntax

```
SHow USEr RSO
```

Description

This command displays information about the current state of Remote Security Officer (RSO) access and the log of access events (Figure 1-58 on page 1-184, Table 1-52 on page 1-184).

For security reasons, this command is accepted only if the user has Security Officer privilege.

Figure 1-58: Example output from the **show user rso** command

```
Remote Security Officer Access is enabled.

Remote Security Officer Log
-------------------------------------------------------------------------------

Remote Security Officer range from: 3ffe::1:6
                                to: 3ffe::1:10
Failed logins ..................... 1
Last failed login ................. 23-Feb-2004 03:28:29
Successful logins ................. 2
Last successful login ............. 23-Feb-2004 03:28:05
-------------------------------------------------------------------------------

Remote Security Officer ........... 3ffe::1:2/128
Failed logins ..................... 0
Last failed login ................. **-***-**** **:**:**
Successful logins ................. 2
Last successful login ............. 23-Feb-2004 05:04:27
-------------------------------------------------------------------------------

Remote Security Officer ........... 192.168.100.200/255.255.255.255
Failed logins ..................... 1
Last failed login ................. 23-Feb-2004 03:31:17
Successful logins ................. 1
Last successful login ............. 23-Feb-2004 04:04:27
-------------------------------------------------------------------------------

Remote Security Officer ........... 192.168.5.0/255.255.255.0
Failed Logins ..................... 1
Last failed login ................. 18-Mar-2004 23:33:50
Successful Logins ................. 0
Last successful login ............. **-***-**** **:**:**
-------------------------------------------------------------------------------

Illegal Login Attempts
-------------------------------------------------------------------------------
IP Address                                 Date/Time                   Attempts
-------------------------------------------------------------------------------
202.175.36.132                             23-Feb-2004 04:03:48               1
172.20.1.3                                 23-Feb-2004 03:27:17               3
2ffe::1:3                                  23-Feb-2004 03:26:34               6
-------------------------------------------------------------------------------
```

Table 1-52: Parameters in the output of the **show user rso** command

Parameter	Meaning
Remote Security Officer Access is...	Whether the Remote Security Officer access is enabled.
Remote Security Officer Log	The list of Remote Security Officers and a log of access events for those Remote Security Officers.
Remote Security Officer	IP address and mask, or IP address range, of a Remote Security Officer. A mask other than 255.255.255.255 defines a range of Remote Security Officer addresses.
Failed logins	Number of failed login attempts by users in the Remote Security Officer address range.

Table 1-52: Parameters in the output of the **show user rso** command (continued)

Parameter	**Meaning**
Last failed login	Date and time of the last failed login attempt, or "`**-***-**** **:**:**`" when there have been no failed attempts.
Successful logins	Number of successful login attempts by users in the Remote Security Officer address range.
Last successful login	Date and time of the last successful login attempt, or "`**-***-**** **:**:**`" when there have been no successful attempts.
Illegal login attempts	A log of illegal login attempts from IP addresses not in one of the defined Remote Security Officer address ranges.
IP address	IP address where the Telnet session originated.
Date/time	Date and time of the login attempt.
Attempts	Number of attempts made from this IP address.

Examples To display the log of Remote Security Officer access events, use the command:

```
sh use rso
```

Related Commands

add user rso
delete user rso
disable user rso
enable user rso

upload

Syntax

```
UPLoad [METhod=TFtp] [DESTFile=destfilename]
    [FIle=filename] [SErver={hostname|ipadd|ipv6add}]

UPLOAD [METhod=ZModem] [ASYn=port] [DESTFile=destfilename]
    [FIle=filename]
```

where:

- *filename* is the name of the file to upload. This may be a full path name for the file in the syntax of the TFTP server.
- *ipadd* is an IP address in dotted decimal notation.
- *ipv6add* is a valid IP address.
- *hostname* is a character string up to 40 characters long.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with asyn 0.
- *destfilename* is a character string 5 to 20 characters long specifying the name of the destination file in the TFTP server file system.

Description

This command uploads a file from the router using TFTP or ZMODEM. It can be issued only by a user with Security Officer privilege.

Any parameters that are not specified use the defaults set with the **set loader** command on page 1-114. Some parameters are invalid or have different meanings depending on the method used to download the file.

The **asyn** parameter specifies the asynchronous port where the file is uploaded if the **method** parameter is set to **zmodem**. If **method** is set to **zmodem**, the **asyn** parameter is required unless it was set with the **set loader** command on page 1-114.

The **destfile** parameter specifies the name the file is to be saved under in the TFTP file system.

The **file** parameter specifies the name of the file on the router's file subsystem and should be a fully qualified file name, including the device name. This parameter is required unless it was already set with the **set loader** command on page 1-114.

The **method** parameter specifies the method to use when uploading the file. If **tftp** is specified, TFTP is used to upload the file. If **method** is **tftp**, the **file** and **server** parameters are required, unless they were previously set with the **set loader** command on page 1-114. If **zmodem** is specified, the ZMODEM protocol is used to upload the file. If **zmodem** is specified, the port parameter must also be specified unless it was previously set with the **set loader** command on page 1-114. Only text files can be uploaded with **method** set to **zmodem**. The **asyn** parameter is not used when **method** is set to **tftp**. The default is TFTP.

The **server** parameter specifies the IP address or the host name (a fully qualified domain name) of the TFTP server where the file is uploaded. If a host name is specified, a DNS lookup is used to translate this to an IP address. See the **set ip nameserver** command on page 14-151 of Chapter 14, Internet Protocol (IP) for more information about setting up name servers. The **ping** command on page 14-129 of Chapter 14, Internet Protocol (IP) can be used to verify that the router can communicate with the server via IP. The **server** parameter is required if **method** is **tftp**, unless it was previously set by the **set loader** command on page 1-114. The **server** parameter is not used when **method** is **zmodem**.

Examples To upload show.scp stored in flash memory to a TFTP server with an IP address of 172.16.8.5, use the command:

```
upl fi=show.scp se=172.16.8.5
```

To upload the reallylongfile.scp file from the router to the TFTP server's download directory, with an IP address of 172.16.8.5 so that the server saves the file as 52-240.scp, use the command:

```
upl fi=/downloads/reallylongfile.scp   se=172.16.8.5
   destf=52-240.scp
```

Related Commands

load
set loader
show file
show loader