

Chapter 5

Managing Configuration Files and Software Versions

Managing Configuration Files	5-3
Loading Files onto the Router	5-4
Loading from a Trivial File Transfer Protocol (TFTP) Server	5-5
Loading from a Web Server	5-5
Loading from a Lightweight Directory Access Protocol (LDAP) Server	5-7
Loading from a Remote Device using Secure Copy	5-8
Additional Loader Commands	5-9
Uploading Files from the Router	5-9
Using HTTP	5-9
Using TFTP and ZMODEM	5-11
Using Secure Copy	5-12
Software Upgrades	5-13
Upgrade Overview	5-13
Install Process	5-14
Filenames	5-15
Licencing	5-15
Patches	5-17
Example: Upgrading to new software	5-17
Example: Upgrading to a new patch file	5-19
Upgrading the GUI	5-20
Command Reference	5-22
create config	5-22
delete install	5-23
disable feature	5-24
disable http debug	5-25
disable http server	5-25
disable ldap debug	5-26
disable release	5-26
enable feature	5-27
enable http debug	5-28
enable http server	5-28
enable ldap debug	5-29
enable release	5-30
load	5-31
purge ldap	5-36
reset http server	5-36
reset loader	5-36
restart	5-37
set config	5-38
set http server	5-39
set install	5-40

set loader	5-42
show config	5-46
show feature	5-48
show http client	5-50
show http debug	5-51
show http server	5-52
show http server session	5-53
show install	5-54
show ldap	5-55
show ldap request	5-56
show loader	5-58
show patch	5-60
show release	5-61
upload	5-62

Managing Configuration Files

Managing configuration files consists of the following:

- [Creating a configuration file](#)
- [Specifying a configuration at startup](#)
- [Working with configuration files](#)
- [Configuring multiple routers](#)

Creating a configuration file

A *configuration file* is a text file that contains a sequence of standard commands for a specific purpose. Configuration files should have an `.scp` or `.cfg` extension. Create a file with one of the following methods:

- Use the `create config` command to save the current configuration.
- Save the current configuration and use the `edit` command to modify it.
- Use the `edit` command to create a file on the router and then type commands into it.
- Create a file on a PC, type commands into it, and `load` it onto the router.
- Use the [show config command on page 5-46](#) to display part of the configuration, capture the output, and save it to a file.

Specifying a configuration at startup

When you start or restart a router, or when it automatically restarts, it executes preconfigured commands in a configuration file. The default script is called `boot.cfg`.

To set a configuration file as a boot script to execute when the router starts, use one of these command:

```
create config set
set config=filename
```

The convention is to use `.cfg` for configuration files. You may want to save the configuration as `boot.cfg`. However, we recommend that you do **not** because it removes the possibility of going back to the original configuration.

To display the name of the configuration file that is set to execute when the router restarts, enter the command:

```
show config
```

To start without a configuration in order to configure it completely from a blank one, use the command:

```
set config=none
```

Working with configuration files

When you use the CLI or GUI to configure the router, it stores this dynamic configuration as a list of commands. To view the router's current dynamic configuration, use the `dynamic` parameter in the [show config command on page 5-46](#).

If you turn off the router or restart it, any unsaved changes to the dynamic configuration are lost. To save changes, use the [create config command on page 5-22](#). Once saved, you have a configuration file, or *script*, that you can use for various purposes such as startup.

You will have many configuration files. Storing them on a router allows you to keep a backup router with configuration scripts for every router in the network to speed up network recovery time. Multiple scripts also let you test new configuration scripts before setting them as the default. For example, to test a new script named `test.cfg`, enter the command:

```
restart router config=test.cfg
```

You can run a configuration file any time without restarting the router by using the command:

```
activate script=filename.cfg
```

This command adds the configuration in the script to the dynamic configuration. For more information about how to create and run scripts, see [Chapter 58, Scripting](#).

You can also set a trigger to automatically execute a configuration script when a predetermined event occurs. For information about creating triggers, see [Chapter 59, Trigger Facility](#).

Configuring multiple routers

Follow these steps when configuring a number of routers with similar requirements:

1. Configure one router by using either the CLI or GUI, if supported.
2. Save the configuration. This creates a configuration file that is stored in the router's flash memory. The file consists of a sorted list of the CLI commands that make up the configuration.
3. Upload the file to a PC by using either the CLI or GUI, if supported.
4. Open the file in a text editor, make necessary changes, and download the file onto each router that you want to configure with it.

Loading Files onto the Router

When you want to upgrade your router with new features, you must load new files onto it. Use the router's loader to load the files. The loader uses the following protocols to load and store files into memory:

- [Loading from a Trivial File Transfer Protocol \(TFTP\) Server](#)
- [Loading from a Web Server](#)
- [Loading from a Lightweight Directory Access Protocol \(LDAP\) Server](#)
- [Loading from a Remote Device using Secure Copy](#)
- [Additional Loader Commands](#)

The loader also uses ZMODEM over an asynchronous port to retrieve files from a network host.

Loading from a Trivial File Transfer Protocol (TFTP) Server

TFTP runs over User Datagram Protocol (UDP). It is simpler and faster than FTP but has minimal capability, such as no provisions for user authentication.

Allied Telesis's TFTP server, AT-TFTP, is on the Documentation and Tools CD-ROM along with instructions in a readme file. You can install it on a PC or server running Windows. See the Hardware Reference for the router for more information about AT-TFTP.

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

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

If the file is not in the top level of the TFTP server, include the path as part of the filename. You can optionally rename the file as it is saved to the router memory, but you should not rename software files.

If you rename the file to one that has more than 8 characters, with an extension of 3 characters (DOS 8.3 format), the translation table dynamically allocates a DOS 8.3 formatted filename. The file is saved to memory under this name and an entry is added to the translation table; the file is managed through this translation table.

Loading from a Web Server

The router has a built-in HTTP client. The HTTP client enables the router to act as a browser by sending HTTP "get" or "post" requests to an HTTP server.

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]
```

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

```
show http client
```

Example: Loading a file over the Internet

This example loads a configuration file from a remote router at company.com to a local router across the Internet using HTTP. The file is called company.cfg.

Before starting, make sure that:

- IP is configured on both routers
- both routers can access the Internet

1. Configure the loader.

If desired, set the loader with defaults to make the process of downloading files simpler in the future. Use the command:

```
set loader method=http server=ip-address-of-remote-router
[other-options...]
```

If the HTTP server requires authentication, then specify the username and password in either the **set loader** or **load** command.

2. Download the configuration file.

Download the file onto the router with the defaults set above by using the command:

```
load file=company.cfg
```

When the download has completed, check that the file is in flash memory by using the command:

```
show file=*.cfg
```

Example: Loading a patch file using HTTP

This example loads a patch file onto the router from a HTTP server on the network. Before starting, make sure that:

- the HTTP server is operating on a host with an IP address (for example, 192.168.1.1) on the network, and that the patch file is in the server's HTTP directory
- the router has an IP address (for example, 192.168.1.2) on the interface connecting it to the HTTP server, and that it can communicate with the server
- there is enough space in the router's flash memory for the new patch file

1. Configure the loader.

Set the loader with defaults to make the process of downloading files in the future simpler.

```
set loader method=http server=192.168.1.1
destination=flash
```

2. Download the patch file.

Download the patch file onto the router by using the defaults set above.

```
load file=54264-01.paz
```

When the download has completed, check that the file is in flash memory by using the command:

```
show file=*.paz
```

To activate the patch, see [“Example: Upgrading to a new patch file” on page 5-19](#).

Loading from a Lightweight Directory Access Protocol (LDAP) Server

LDAP is a 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={bootblock|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 information about one or all LDAP requests, use the command:

```
show ldap request
```

To display LDAP data on the terminal as it is received, use the command:

```
enable ldap debug
```

To disable debugging information, use the command:

```
disable ldap debug
```

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 22-135 of Chapter 22, 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 49-4 of Chapter 49, Public Key Infrastructure (PKI).

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

Loading from a Remote Device using Secure Copy

Secure Copy (SCP) provides a secure way to copy files onto the router from a remote device. You can load files onto the router by using one of the following methods:

- locally by using the router's CLI. This uses the SSH client on the router.
- remotely by using a suitable client on a remote device and the SSH server on the router.

Secure Copy connections cannot load to the bootblock.

Loading using the router

The router can load files from a remote server using SCP. To do this, do all of the following:

- Check the server is running SCP and set a username.
- Set either a password or RSA keyid on the server to authenticate the user. If using RSA authentication, set the public key onto the server.

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

```
load [method=scp] [delay=delay] [destfile=destfilename]
      [destination={cflash|flash|nvs}]
      [{file|srcfile}=filename]
      [{keyid=key-id|password=password}]
      [server={hostname|ipadd|ipv6add}] [username=username]
```

Example

In this example, the SCP server has an IP address of 192.168.1.2, with the username "john", and the password "secret" set on it. To download the file /atr-281/86s-281.rez from the server, use this command on the router:

```
load method=scp username=john password=secret
      server=192.168.1.2 file=/atr-281/86s-281.rez
      destination=flash
```

If desired, set the loader with defaults to make the process of downloading files simpler in the future. Use the command:

```
set loader method=scp username=john password=secret
      server=192.168.1.2 destination=flash
```

Loading using the remote device

Secure Copy allows remote client devices to load files onto the router. To do this, do all of the following:

- Check the router is running as a SSH server with SCP enabled.
- Configure the user to allow them to connect using SSH.
- Set either a password or RSA keyid on the router to authenticate the user. If using RSA authentication, set the public key onto the router.

Example

In this example, the username is "Alice" and the client device is running Linux. The router has the IP address 192.168.1.1. To copy the file 86s-281.rez onto the router, use this command on the client device:

```
scp atr-281/86s-281.rez alice@192.168.1.1:86s-281.rez
```

Additional Loader Commands

The router loads one file at a time. Wait for the current transfer to complete before initiating another one. To display the current configuration of the loader, and the progress of the current transfer, enter the [show loader command on page 5-58](#).

You are likely to repeat the process of downloading files onto the router using a similar method each time. You can set defaults for some or all of the loader parameters. You can then use or override some or all of these defaults for each load.

To set loader defaults, enter the [set loader command on page 5-42](#). You can set all parameters except **destfile**, **scrfile** and **file** back to the factory defaults with the option **default**.

To stop a load at any time and leave the loader ready to load again, enter the [reset loader command on page 5-36](#).

You can also use the loader to upload files, such as log files, from the router to a host on the network (see [“Using TFTP and ZMODEM” on page 5-11](#)).

Uploading Files from the Router

This section explains the following popular methods to upload files from a router to another location:

- [Using HTTP](#)
- [Using TFTP and ZMODEM](#)
- [Using Secure Copy](#)

Note that some file types cannot be uploaded from the router for security reasons.

Using HTTP

When you use a web browser to load files off the router onto a PC, you are accessing the router's built-in HTTP server. You also access the server when you use the **load** command to load files from one router to another.

The HTTP server offers an alternative loading and uploading method to TFTP, at the same level of reliability and security as FTP. Unlike HTTP and FTP, TFTP is a connectionless protocol and does not guarantee reliable delivery of files across the Internet. If you are loading a file across the Internet, we recommend that you use HTTP.

You also access the HTTP server when you use a web browser to manage the router using its GUI. [Chapter 3, Using the Graphical User Interface \(GUI\)](#) describes the GUI.

Commands The HTTP server is enabled by default. To disable the HTTP server, use the command:

```
disable http server
```

To enable the HTTP server after it has been disabled, use the command:

```
enable http server
```

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

```
show http server
```

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

```
show http server session
```

The router logs all "get", configure, and monitor requests as well as authorisation failures (see [Chapter 60, Logging Facility](#)). You can also display debug messages by using the command:

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

Debug messages display authorisation attempts, HTTP "get" and "post" requests and responses, and TCP state changes.

To disable debugging, use the command:

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

To list the currently enabled debugging options, use the command:

```
show http debug
```

To restart the HTTP server, disable debugging, and clear all counters, use the command:

```
reset http server
```

Resolving URLs When you enter a URL, the router checks whether it matches one of the following:

1. a page in the GUI, or
2. a file in the router's flash memory

If the URL does not match one of these, the HTTP server returns the HTTP error 404, File Not Found.

If you browse to the router's IP address without specifying a filename and the router has a GUI enabled, the server loads the GUI home page (usually the System Status page). If you browse to the router's IP address without specifying a filename and the router does not have a GUI enabled, the server returns error 404, File Not Found.

HTTPS: Secure access You can secure the HTTP server so that it only accepts HTTPS connections. For a configuration example, see ["Configuration Example" on page 45-7 of Chapter 45, Secure Sockets Layer \(SSL\)](#).

Example: Uploading to a PC This example loads a configuration file from a router to a PC using HTTP. The file is called `company.cfg` and the IP address for the router is 192.168.1.1.

Before starting, make sure that:

- IP is configured on the router
- the PC can access the router's subnet

1. Browse to the file.

Use the router's IP address, followed by a forward slash, and the name of the file as follows:

```
http://192.168.1.1/company.cfg
```

The browser prompts you for a username and password.

2. Enter your username and password.

The username and password must be valid—they must match a user defined in the User Authentication Database or on an external authentication server. For information about user accounts, see [Chapter 41, User Authentication](#).

3. Save the file to your PC.

Follow the browser's prompts.

Using TFTP and ZMODEM

The **upload** command enables you to load files from the router to a network host using TFTP or ZMODEM. Allied Telesis's TFTP server, AT-TFTPD, is provided on the Documentation and Tools CD-ROM. You can install it on a PC or server running Windows. The CD-ROM also includes a readme file describing how to install and use AT-TFTPD.

Upload files by using one of the following commands:

```
upload [method=tftp] [file=path/filename]
      [server={hostname|ipadd}]

upload [method=zmodem] [file=filename] [asyn=port]
```

The **upload** command uses defaults set with the **set loader** command for parameters not specified with the upload command.

Example: Uploading to a TFTP server This example uploads a configuration file from the router to a TFTP server on the network. Before starting, ensure that:

- the TFTP server is operating on a host with an IP address (for example 192.168.1.3) on the network.
- the router has a valid IP address (for example 192.168.1.2) on the interface connecting it to the TFTP server, and that it can communicate with the server.
- the configuration file is present in the router's flash memory.

1. Configure the loader.

Set the loader with defaults to make the process of downloading and uploading files simpler in the future.

```
set loader method=tftp server=192.168.1.3
```

2. Upload the configuration file.

Upload the configuration file from the router into the TFTP directory of the TFTP server on the network by using the defaults set above.

```
upload file=filename.cfg
```

Monitor the load progress with the command:

```
show load
```

When the upload is complete, check that the file is in the TFTP directory on the network host.

Using Secure Copy

The Secure Copy Protocol (SCP) provides a secure way to copy files from the router onto a remote device. You can upload files from the router by using one of the following methods:

- locally by using the router's CLI. This uses the SSH client on the router.
- remotely by using a suitable client on a remote device and the SSH server on the router.

Uploading using the router

The router uses SCP to load files onto a remote server. To do this, do all of the following:

- Check the server is running SCP and set a username.
- Set either a password or RSA keyid on the server to authenticate the user. If using RSA authentication, set the public key onto the server.

To upload a file from the router, use the command:

```
upload [method=scp] [file=filename] [destfile=destfilename]
[ {keyid=key-id | password=password } ]
[server={hostname | ipadd | ipv6add} ] [username=username]
```

Example

In this example, the SCP server has an IP address of 192.168.1.2, with the username "john", and the password "secret" set on it. To upload the file voip.cfg to the server, use this command on the router:

```
upload method=scp server=192.168.1.2 username=john
password=secret file=voip.cfg destfile=voip.cfg
```

If desired, set the loader with defaults to make the process of uploading files simpler in the future. Use the command:

```
set loader method=scp server=192.168.1.2 username=john
password=secret
```

Uploading using a remote device

Secure Copy allows remote client devices to load files from the router. To do this, do all of the following:

- Check the router is running as a SSH server with SCP enabled.
- Configure the user so that they are allowed to use SSH.
- Set either a password or RSA keyid on the router to authenticate the user. If using RSA authentication, set the public key onto the router.

Example

In this example, the username is “Alice” and the client device is running Linux. The router has the IP address 192.168.1.1. To copy the file `voip.cfg` from the router, use this command on the client device:

```
scp alice@192.168.1.1:voip.cfg /root/voip.cfg
```

Software Upgrades

The most current software is on the router when it leaves the factory, and the router automatically loads it from flash memory into RAM when you first start it (see “How the Router Starts” in the Hardware Reference for details). You can upgrade software on your router as it becomes available by downloading the latest version along with CLI help files from www.alliedtelesis.com/support/software.

This section contains the following topics:

- [Upgrade Overview](#)
- [Install Process](#)
- [Filenames](#)
- [Licencing](#)
- [Patches](#)
- [Example: Upgrading to new software](#)
- [Example: Upgrading to a new patch file](#)
- [Upgrading the GUI](#)

Upgrade Overview

Upgrading software is a 2-part procedure: loading the correct files into the router’s file system, and then installing the files. The following is an overview of this procedure:

1. Contact your Allied Telesis representative to obtain the new file.
2. Place the file in a directory on a server that the router can access, such as a TFTP server. See “[Loading Files onto the Router](#)” on page 5-4 for different ways to load files.
3. Load the file into the file system by using the **load** command.
4. Set the router to use the new file as follows:

Enable a file and specify the password if necessary by using the **enable release** command.

Install the release, GUI resource, or patch file by using the **set install** command.

Set CLI help for a help file by using the **set help** command on page 2-15 of [Chapter 2, Using the Command Line Interface \(CLI\)](#).

See “[Install Process](#)” on page 5-14 for details about installing software on the router.

5. Restart the router if you loaded release or patch files.

Install Process

The router keeps information, called *install records*, about the files it installs and uses. The following table describes the types of install:

Install Type	Description
Preferred	Software that the router routinely uses. This install is completely configurable, and the router is set to run it when you receive the router.
Temporary	Allows software and/or patch to be loaded just once for testing in case it causes a problem. Can be completely configured. 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.
Default	Used when only the flash boot release is present. This is a very limited release of the software and is the last resort. It cannot be modified but a manager can set a patch for it.

The router inspects install information in strict order as follows:

1. The router checks for a temporary install. If one is specified, the router loads it into RAM and runs it. The router then deletes the temporary install information so it cannot load again. This information is deleted even if the temporary install triggers a fatal condition causing the router to reboot immediately.
2. If no temporary install is defined or the temporary install information is invalid, the router checks for a preferred install. If present, the preferred install is loaded. The router never deletes the preferred install information.
3. If neither a temporary install nor a preferred install is specified, the router loads the default install. The Install module ensures that the default install is always present in the router.

To change installation information when the router boots up or later, use the command:

```
set install={temporary|preferred|default} [gui=file-name]
[release=release-name] [patch=patch-name]
```

The **set install** command requires a user with Security Officer privilege when the router is in security mode.

When you set a patch file as part of a temporary or permanent install, you must also set the corresponding release file in the same command if it has not already been set as part of that install. You can set the patch, but not the release, for the default install.

To delete a temporary or preferred install, enter the following command and specify the desired option:

```
delete install={temporary|preferred}
```

To delete the patch associated with the default install, use the command:

```
delete install=default
```

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

```
show install
```

Filenames

Software products are available as compressed files with filename formats of *mmm-nnn.ext* where:

- *mmm* identifies the device model, for example, AT-9924s or AR750S
- *nnn* is the version identifier, for example, 273 for version 2.7.3
- *.ext* is the filename extension *.rez*

This software contains code that runs the router, and a number of the files can be stored on the router at once. Software is specific to a particular series of router, and may run on just several models in a series. This means that the software version must be appropriate for your router model. This information—filenames and models—is provided in release notes that routinely accompany our software.

The following table explains software files for the router.

File type	File Extension	Purpose
release	rez	Runs the router and controls features.
patch	paz	Small software updates. These files must be compatible with the release file on your router.
GUI resource	rsc	Graphical user interface and its help file. These files must be compatible with the release on your router.
help	hlp	Command line help. Help files typically support a particular software version but can run under others.



Product software is released as a single compressed ASCII file, and consists of a header followed by a sequence of Motorola S-records that contain code for the release. The header has a standard format that gives the router important information. Do **not** change this header. Doing so might cause the file load or install to fail. The router could be put into a state where it would require field service in order to boot correctly.

Licensing

Release licences Product software sometimes requires a release licence. To determine if you need one, use the command:

```
show release
```

If your router does not have the following output, contact your Allied Telesis representative to request a licence.

Release	Licence	Period
-----	-----	-----
any	full	-
-----	-----	-----

A new release licence is not required when you are updating to a minor or maintenance release.

Special feature licences

Special features are often offered that are separate from standard software products. A special licence and password are required to activate these features. A licence can be full (unlimited time) or a 30-day trial. Passwords must be ordered from your authorised distributor or reseller.

You must specify the special features to be licenced and the serial number of the routers where they are to be enabled. Passwords cannot be transferred from one router to another. Password information is stored in flash memory. You must set the system date and time before you can enable a trial feature licence.

If you are using the CLI, enter the command:

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

If you are using the GUI:

1. Select Management > Software > Licences from the sidebar menu.
2. Select the **Feature Licences** tab.
3. Click **Add**.
4. Enter the feature name and password.
5. Click **Apply**.

To disable a special feature licence, use the command:

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

To display a list of current special feature licences, use the command:

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

Examples: Special features

Features that are available and that require special feature licences depend on region and router model. Software features that may require a special feature licence are:

- Triple DES S/W
- Firewall SW (enabled on the AR440S, AR441S, AR442S, and AR450S)
- Firewall SMTP Application Gateway (enabled on the AR440S, AR441S, AR442S, and AR450S)
- Firewall HTTP Application Gateway (enabled on the AR440S, AR441S, AR442S, and AR450S)
- DES encryption
- IPv6
- Resource Reservation Protocol (RSVP)
- BGP-4
- Load balancer
- WAN load balancing

Special feature packs

Most software features that require a special feature licence are bundled into one of the following special feature licence packs:

- Advanced Layer 3 Feature Licence
- Security Pack Feature Licence

Patches

Patch software contains changes to the product software. It often contains fixes to software problems, although it also include enhancements.

Patches are compressed files with filenames in the format *rrr-*vv*.paz* where:

- *rrr* identifies the associated version, such as 2.7.1
- *vv* is the version number that identifies the patch in a series, such as 1, 2, 3

For example, *271-01.paz* means 2.7.1 is the version that the patch modifies, and 01 is the version number that identifies the patch in a series of patches (1, 2, 3...).

Patches are specific to individual releases and router series. This means that patch files must be appropriate for your model and the release level on it. This information—filenames and models—is provided in release notes that accompany patches.

Patches are loaded into flash memory.

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.

Information on using the loader is in [“Loading Files onto the Router” on page 5-4](#). To verify the patch is loaded, use the commands:

```
show file
```

```
show patch
```

To remove a patch, use the command:

```
delete file
```



Patch files are ASCII files, and consist of a header followed by a sequence of Motorola S-records that contain code about the patch. The header has a standard format that gives the router important information. Do **not** change this header. Doing so might cause the file load or install to fail. The router could be put into a state where it would require field service in order to boot correctly.

Example: Upgrading to new software

This example assumes the router is correctly configured to allow TFTP to function. This means that IP is configured and the router can communicate with the designated TFTP server. It assumes that the TFTP server is functioning correctly and that correct files are in the server's directory. The IP address of the server is 172.16.1.1.

The name of the release file being loaded is *54-291.rez*.

I. Configure the loader.

The loader has defaults to make the process of downloading files easy. Release and patch files are stored in flash memory.

```
set loader method=tftp server=172.16.1.1 destination=flash
```

2. Load the new file onto the router.

Make sure there is room in the file system for the new file. Also make sure the software is compatible with the router model (see “[Filenames](#)” on [page 5-15](#)). Load any patch files required, and the help file for the version (see “[Loading Files onto the Router](#)” on [page 5-4](#)). To load the software file using the loader default settings, enter the command:

```
load file=54-291.rez
```

Wait for the file to load, which can take several minutes even over a high speed link. To see the progress of the load, enter the command:

```
show load
```

To check that the file is loaded, enter the command:

```
show file
```

3. Enter licence information for the new software if necessary.

Product software sometimes requires a release licence. If so, enter the licence password:

```
enable release=54-291.rez password=ce654398fbe  
number=2.9.1
```

A password is provided by your authorised distributor or reseller and is unique for the router serial number.

Enter passwords for any special feature licences by using the command:

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

4. Test the new software.

Set the new file to run as a temporary install. This sets the router to load the new file when it next reboots.

```
set install=temporary release=54-291.rez
```

To use the current router configuration again, store the dynamic configuration as a configuration file, and set the router to use this configuration when it restarts. Releases are typically backward-compatible, so your current configuration should run with little or no modification on a later release. Use the commands:

```
create config=myconfig.cfg
```

```
set config=myconfig.cfg
```

The **set config** information survives the software update.

Reboot the router with the command:

```
restart reboot
```

The router reboots, loading the new file and specified configuration. To check that the temporary file loaded properly, use the command:

```
show install
```

5. Make the new software the default (preferred) file.

If the router operates correctly with the new software, make it permanent with the command:

```
set install=preferred release=54-291.rez
```

Every time the router reboots from now on, it will load the new software from the file system.

Save the configuration again by using the commands:

```
create config=myconfig.cfg
```

```
set config=myconfig.cfg
```

Example: Upgrading to a new patch file

Use this procedure to upgrade the software version currently running on the router with a new patch. This example assumes that the Software Version 2.6.4 is set as the preferred version.

The name of the patch file being loaded is 54264-01.paz.

1. Load the new patch file onto the router.

Load the new file onto your router. For details, see [“Loading Files onto the Router” on page 5-4](#).

```
load file=54264-01.paz
```

Check that the file successfully loaded by using the command:

```
show file=*.paz
```

2. Test the patch.

Set the release to run as a temporary install, which means the router loads the patch just once—the next time it reboots.

```
set install=temporary release=54-264.rez  
patch=54264-01.paz
```

If you want to use the current router configuration again, store the dynamic configuration as a configuration script file, and set the router to use this configuration when it restarts. Use the commands:

```
create config=myconfig.scp  
set config=myconfig.scp
```

Reboot the router by using the command:

```
restart reboot
```

The router reboots and loads the new patch file and the specified configuration. Check that the router operates correctly with the new patch file.

3. Make the patch part of the default (permanent).

If the router operates correctly with the new patch, make the release permanent.

```
set install=preferred release=54-264.rez  
patch=54264-01.paz
```

Every time the router reboots from now on, it loads the new release and patch from flash memory.

Save the configuration again by using the commands:

```
create config=myconfig.cfg  
set config=myconfig.cfg
```

Upgrading the GUI

Before you start, ensure that the router is running the most recent release and patch files. The GUI is not part of the release file, but the most recent GUI resource file is compatible with the most recent software version. To check which files the router is running, refer to the Current Install section in the output from the [show install](#) command on page 5-54.

If you are updating both the release and the GUI resource file, set the preferred release and restart the router before installing the GUI. You can download the latest resource file from www.allied-teleasis.com/support. See “Example: Upgrading to new software” on page 5-17.

You can use the GUI to load the new resource file onto the router (Management > Software > Upgrade), but you need to use the CLI to install the new file.

1. If required, delete the old GUI resource file.

If required, you can store more than one GUI resource file on the router at a time. If you want to delete the previous GUI resource file (for example, to save memory), you must first disable the GUI by using the command:

```
disable gui
```

Then delete the GUI resource file by using the command:

```
delete file=old-gui.rsc
```

where *old-gui.rsc* is the name of the GUI resource file that you are replacing.

Wait until flash compaction finishes, which may take several minutes.



Caution While flash is compacting, do not restart the router or use commands that affect the flash file subsystem such as **create**, **edit**, **load**, **rename**, or **delete**. Wait until you get a message that file compaction is complete. Interrupting flash compaction may damage files.

If you have multiple valid resource files and releases stored on the router, use the [set install](#) command on page 5-40 to change the release and resource file the router uses (see below).

2. Load the new file onto the router.

Download the GUI resource file for your model of router from the web site to your TFTP server. Do **not** rename the file. If you do, make sure that you rename it with its original name. Resource files use a fixed naming convention. If you change the GUI resource file's name, the router will not recognise it as a valid file and you will be unable to use it for configuration.

Some TFTP servers do not support filenames longer than 8 characters and therefore will not allow you to load the file from the server. With such servers, you can rename the GUI file to a short name on the TFTP server. However, you must then rename it correctly on the router or switch.

Load the GUI resource file from your TFTP server to the router by using the command:

```
load file=filename.rsc server=server
```

where:

- *filename* is the name of the GUI resource file, as shown on the support site for your router. If the file is not in the top level of the TFTP server, include the path.
- *server* is the IP address of the TFTP server the file is loaded from.

When the router has loaded the file into its RAM, it displays *File transfer successfully completed*. It then writes the file to flash memory, which takes approximately 30 seconds after the message. Once the file has been copied to flash memory, you can enter commands that refer to it.

3. Install the new file as the preferred GUI.

If you are updating both the release and resource file, set the preferred release and restart the router, if you have not already done so.

To set the new GUI resource file as the preferred resource file, use the command:

```
set install=preferred gui=filename.rsc
```

If you disabled the GUI to delete the old resource file, enable it again by using the command:

```
enable gui
```

Check that the new GUI resource file is valid for your device by using the command:

```
show gui
```

If it is not, or if the file was corrupted during the download, disable the GUI, delete the file, and try again.

4. Point your web browser at the router's IP address.

You may have problems if your browser has stored a local copy of the old GUI file. If so, delete these temporary files, which means clearing the cache as follows:

To clear the cache in Internet Explorer:

1. From the Tools menu, select **Internet Options**.
2. On the General tab, click the **Delete Files** button.
3. The Delete Files dialogue box opens. Click the **OK** button.

To clear the cache in Netscape 6.2.x

1. From the Edit menu, select **Preferences**
2. Click the **Advanced** menu option to expand it.
3. Select the **Cache** menu option.
4. Click the **Clear Memory Cache** and **Clear Disk Cache** buttons.

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 lxv of About this Software Reference](#) for details of the conventions used to describe command syntax. See [Appendix A, Messages](#) for a complete list of messages and their meanings.

create config

Syntax CREate CONfig=*filename* [SET]

Description This command creates a configuration or script file that contains commands required to recreate the current dynamic configuration of the router. You can also set the switch to the file at the same time you create it. This command requires a user with security officer privilege when the router is in security mode.

Parameter	Description
CONfig	<p>Name of the configuration file or script to create. If one already exists, it is replaced.</p> <p>The <i>filename</i> is in the format [<i>device</i>:]filename.ext and can be:</p> <ul style="list-style-type: none"> • a maximum of 39 characters • uppercase and lowercase letters • digits • # \$ % & ! ' () + , - . ; = @ [] ^ _ ` { } ~ and space <p><i>device</i> indicates the physical location where the file is stored. The default is flash.</p> <p>.ext is an 3-letter extension, such as .txt or .scp.</p> <p>Invalid characters are * " \ : ? / < ></p> <p>Default: no default</p>
SET	Sets the switch to use the configuration file or script specified by <i>filename</i> when the switch boots up again.

Examples To save the current dynamic configuration to a script file called test.cfg, use the command:

```
cre con=test.cfg
```

Related Commands [restart](#)
[set config](#)
[show config](#)

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.

Both 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 after you have finished testing it, use the command:

```
del inst=temp
```

Related Commands [set install](#)
[show install](#)

disable feature

Syntax `DISable FEAture={featurename|index}`

Description This command disables the specified special feature licence. This command requires a user with Security Officer privilege when the router is in security mode.

Parameter	Description
FEAture	The special feature being disabled, identified by specifying the name or the index number assigned to it. The special feature must exist on the router and currently be enabled. Default: no default
<i>featurename</i>	The name assigned to the special feature when it was enabled with the enable feature command on page 5-27 . The featurename is a string 1 to 12 characters long, containing any printable character.
<i>index</i>	A decimal number used to identify the feature. Features are sequentially assigned a number from 1 upwards. To display the feature numbers, use the show feature command on page 5-48 .

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 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. Debugging is disabled by default.

Parameter	Description
DEBug	The type of debugging which is disabled. Default: no default
ALL	All debugging is disabled.
AUTH	Debugging of authentication attempts is disabled.
MSG	Debugging is disabled for HTTP "get" and "set" requests and responses.
SESSion	Debugging is disabled for TCP state changes and session activity.
STATe	Debugging is disabled for state changes in the state machine. When enabled, this debugging 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 provides HTML pages out of the router's flash memory to a web browser, and allows users to log 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 stop LDAP debugging, use the command:

```
dis ldap deb
```

Related Commands [enable ldap debug](#)
[show ldap](#)

disable release

Syntax DISable RELease=*release-name*

Description This command removes the licence for the specified release file. The **release** parameter specifies the name of the release file. The name of a release file uses the format [*device:*]filename.ext. Invalid characters are * + = " | \ [] ; : ? / , < > and wildcards are not allowed. Valid characters are:

- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

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](#)

enable feature

Syntax `ENable FEAture=featurename PASSword=password`

Description This command enables the special feature licence identified by the special feature licence name and password. This command requires a user with Security Officer privilege when the router is in security mode. You must set the system date and time before you can enable a trial feature licence.

Contact your authorised Allied Telesis distributor or reseller for details and passwords of the product licence you have purchased.

Although this command enables ENCO licensed features, such as 3DES and AES, they are not active until the router restarts.

Parameter	Description
FEAture	Specifies a user-defined name for the special feature licence that is in output of the show feature command on page 5-48 and identifies the special feature licence in other commands. The <i>featurename</i> can be: <ul style="list-style-type: none"> from 1 to 12 characters long any printable characters Default: no default
PASSword	Password for the special feature licence that identifies the features being licenced, whether the licence is full or 30-day trial, and the router serial number. Password information is stored in flash memory. The <i>password</i> can be: <ul style="list-style-type: none"> 16 characters long minimum uppercase and lowercase characters and digits Default: no default

Examples To enable the special feature licence “Triple DES” with the sample password 591a9d5d9b2e8969cbf7, use the command:

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

Related Commands [disable feature](#)
[show feature](#)

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 where the command was entered. To enable combinations of debugging options, enter multiple commands. Debugging is disabled by default.

Parameter	Description
DEBUg	The type of debugging which is enabled. Default: no default
ALL	All debugging is enabled.
AUTH	Debugging of authentication attempts is enabled.
MSG	Debugging is enabled for HTTP "get" and "set" requests and responses.
SESSion	Debugging is enabled for TCP state changes and session activity.
STATe	Debugging is enabled for state changes in the state machine. This debugging 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 sends HTML pages from the router's flash memory to a web browser so that users can log 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 lets a user trace the execution of LDAP requests by displaying step-by-step information. By default, debugging is disabled.

Examples To start LDAP trace debugging, use the command:

```
ena ldap deb
```

Related Commands [disable ldap debug](#)
[show ldap](#)

enable release

Syntax ENable RElease=*release-name* [PASSword=*password*]
 NUMber=*release-number*

where:

- *release-name* is the name of a release file, in the device:]filename.ext format. Invalid characters are * + = " | \ [] ; : ? / , < > , and wildcards are not allowed. Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ~ ' ! @ # \$ % ^ & () _ - { }
- *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 on 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 if required, and is specific to a particular router serial number. The password enables the release with either a full licence or a 30-day licence.

The **number** parameter specifies the software version for the release file being licenced. This is entered in dotted decimal form, for example 2.3.1.

Examples To enable version 28-231.rel with the password CE645398FBE for software version 2.3.1, use the command:

```
ena rel=28-231.rel pass=ce645398fbe num=2.3.1
```

Related Commands [disable release](#)
[show release](#)

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]`

`LOAD [METHod=SCP] [DELay=delay] [DESTFile=destfilename]
 [DESTination={CFlash|FLash|NVs}]
 [{File|SRCFile}=filename]
 [{KEYid=key-id|PASSword=password}]
 [Server={hostname|ipadd|ipv6add}] [USERName=username]`

where:

- *delay* is a time delay in seconds.
- *hostname* is a string 1 to 40 characters long.
- *ipadd* is an IP address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address.
- *filename* is a 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 49-4 of Chapter 49, Public Key Infrastructure \(PKI\)](#).
- *key-id* is a decimal ID number of an encryption key used for authentication.
- *password* is a string 1 to 60 characters long that is used for basic server authentication.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with asyn0.
- *username* is a string 1 to 60 characters long that is used for basic server authentication.

Description This command downloads a file to the router using one of the following:

- Trivial File Transfer Protocol (TFTP)
- HyperText Transfer Protocol (HTTP)
- Lightweight Directory Access Protocol (LDAP)
- Secure Copy (SCP)
- ZMODEM
- direct input from an asynchronous port

Parameters not specified use the defaults set with the [set loader command on page 5-42](#). Some parameters are invalid or have different meanings depending on the method used to download the file. This command requires a user with security officer privilege when the router is in security mode.



Caution While flash is compacting, do not restart the router or use commands that affect the flash file system such as **create**, **edit**, **load**, **rename**, or **delete**. Wait until you get a message that file compaction is complete. Interrupting flash compaction may damages files.

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 already set with the [set loader command on page 5-42](#).

The **attribute** parameter defines a keyword specifying the type of object to retrieve from an LDAP repository. A list of currently recognised keywords and their respective object types are in the following table.

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 memory 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** memory is specified, the file is stored in the Flash File System (FFS) on the router. If **destination** is not specified, and has not already been set with the [set loader command on page 5-42](#), the default is **flash**.



Caution 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 bootblock is being written, so 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 **keyid** parameter specifies the ID number of a RSA private or public key that is held on the router. This parameter is only valid when using SCP. The server receiving the load request must have the public key for this authentication to work.

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 **scp** is specified, SCP is used to download the file. When **scp** is specified, the **username** parameter must be set, along with either the **password** or **keyid** parameter. If **zmodem** is specified, the ZMODEM protocol is used to download the file. When **zmodem** is specified, the **asyn** parameter is required unless it has already been set with the [set loader command on page 5-42](#). 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**, **scp**, or **tftp**. The default is **tftp**, unless another method has been specified using the [set loader command on page 5-42](#).

The **password** parameter specifies the password for the LDAP, HTTP or SCP methods if server authentication is required. As the password is typed it appears as plain text on the screen, so it should only be used in a secure area.

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 host name (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 22-162 of Chapter 22, Internet Protocol \(IP\)](#) for more information about setting up name servers. The [ping command on page 22-135 of Chapter 22, 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 5-42](#). The **server** parameter is not valid when **method** is set to **zmodem** or **none**. The following are examples of valid server names for the HTTP or LDAP method:

```
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 5-42](#). The **file** parameter is a full path name rather than just a filename. The only restriction is that the last part of the file parameter must be a valid filename for the loader. When **method** is set to **tftp**, **http**, **zmodem**, or **none**, valid filenames have the format *filename.ext* and can be 1 to 32 characters long. The following are examples of valid filenames 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 filename and working backwards, the first character not valid in filenames delimits a valid filename 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 filenames 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 5-42](#), use the command:

```
loa
```

To download the 28-761.rez version 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.rez des=f1 se=172.16.8.5 del=60
```

To load the script file show.scp from asynchronous port 1, use the command:

```
loa fi=show.scp asy=1
```

In this example, the router is downloading the file abc.cfg from a SCP server with the IP address 172.16.8.5. The user has the username "john" and the password "secret" on the server. To download the file and save it as abc.cfg in flash memory, use this command:

```
loa met=scp fi=/downloads/abc.cfg se=172.16.8.5 des=f1
  usern=john pass=secret
```

To load the script file show.scp from asynchronous port 1 by 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 download 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 memory, use the command:

```
loa fi=ar410B10.fbr ser=172.16.8.5 des=boot
```

To download reallylongfile.rez into the flash memory 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 memory 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](#)

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](#)

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.

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 file transfer currently being loaded. All resources used by the transfer are released and any file in the process of being created is deleted. The loader becomes ready immediately for a new load to be initiated.

Related Commands [load](#)
[set loader](#)
[show loader](#)

restart

Syntax RESTART Router [CONfig={*filename*|NONE}]

RESTART REBoot

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
- ~ ' ! @ # \$ % ^ & () _ - { }

Description This command restarts the router with either the current configuration file (set with the [set config command on page 5-38](#)) 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 restarts without executing a 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 prompts for a confirmation.

Examples To warm start the router using a configuration file named test.cfg instead of the default configuration file, use the command:

```
restart rou config=test.cfg
```

Related Commands [set config](#)
[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. Valid characters are:

- a maximum of 39 characters
- uppercase and lowercase letters
- digits
- ~ ' ! @ # \$ % ^ & () _ - { }

Description This command sets the configuration file that the router uses as its default configuration. The file name is stored in flash memory.

The command requires a user with security officer privilege when the router is in security mode. 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 "myboot.cfg", use the command:

```
set con=myboot.cfg
```

Related Commands [restart](#)
[create config](#)
[show config](#)

set http server

Syntax SET HTTP SERVER [PORT=0..65535]
[SECURITY=ON|OFF|ENABLED|DISABLED|TRUE|FALSE]
[SSLKEY=0..65535]

Description This command sets the options for when the router acts as an HTTP server.

Parameter	Description
PORT	Specifies which TCP port number, from 0 to 65535, the HTTP server is listening on. Default: 80
SECURITY	Specifies whether the HTTP server accepts SSL secured HTTPS connections or unsecured HTTP connections. Default: off ON, ENABLED, TRUE All connections made to the server must be SSL connections. Setting security to on enables SSL on the router. See Chapter 45, Secure Sockets Layer (SSL) for details on configuring SSL. OFF, DISABLED, FALSE All connections made to the server must be non-SSL connections.
SSLKEY	The number, from 0 to 65535, of a valid private key ID for SSL operation. This parameter is required when the security parameter is set to on . Default:

Examples To change port that the HTTP server is listening on to the TCP port 550, use the command:

```
set http serv po=550
```

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](#)
[set ssl](#)
[show http server](#)

set install

Syntax SET INSTall={TEMPorary|PREFerred|DEFault}
 [GUI=[*file-name*|NONE]] [RELease=[*release-name*]
 [PATch=[*patch-name*]]

where:

- *release-name* is the name of a release file in [*device*:]filename.ext format. Invalid characters are * + = " | \ [] ; : ? / , < > , and wildcards are not allowed. Valid characters are:
 - uppercase and lowercase letters
 - digits
 - ~ ' ! @ # \$ % ^ & () _ - { }
- *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. The first time you upgrade to a new software version, you need to install the release before the GUI. If you are upgrading from a base version to a later maintenance version, you can install the release and GUI in the same step.

This command requires a user with Security Officer privilege when the router is in security mode.

The **install** parameter specifies which install is to be set. The Install module maintains installation information and loads the correct information at bootup. An *install* is a record that identifies a release, a GUI resource file, 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. The release, GUI resource file, and an associated patch 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. You can set the release file for the preferred install to a release file that does not exist in the file system. If the router reboots before the release file is loaded into the file system, the system boots using the default install.

The **gui** parameter specifies the resource file used when the GUI is accessed. The resource file name includes information about the router model, the software version, and the language. An example on an AR415S is 415s_291-00_en_d.rsc for the English version of 2.9.1 software.

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 version. 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 5-54** 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. You can not set the patch for the preferred install to a file that does not exist in the file system.

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 `415s_291-00_en_d.rsc`, use the command:

```
set inst=pref gui=415s_291-00_en_d.rsc
```

Related Commands

- [delete install](#)
- [reset gui](#)
- [show install](#)
- [show system](#)

set loader

Syntax SET LOAdER [ASyn={*port*|DEFault}]
 [ATtribute={CErt|CRl|CAcert|DEFault}]
 [BASEobject={*dist-name*|DEFault}]
 [DElay={*delay*|DEFault}] [DESTfile=*destfilename*]
 [DESTination={BOOTblock|FLash}|DEFault]
 [HTTProxy={*hostname*|*ipadd*|DEFault}]
 [METhod={HTTP|LDAP|SCP|TFtp|WEB|WWW|ZModem|NONE|
 DEFault}] [{KEYid=*key-id*|PASSword=*password*|DEFault}]
 [PROxyport={1..65535|DEFault}] [SRCfile|FILE=*filename*]
 [Server={*hostname*|*ipadd*|*ipv6add*|DEFault}]
 [SERVPort={1..65535|DEFault}]
 [USERName={*username*|DEFault}]

where:

- *dist-name* is an X.500 distinguished name, as described in “Distinguished names (DN)” on page 49-4 of Chapter 49, Public Key Infrastructure (PKI).
- *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 string 1 to 40 characters long.
- *ipadd* is an IP address in dotted decimal notation.
- *ipv6add* is a valid IPv6 address.
- *key-id* is a decimal ID number of an encryption key used for authentication.
- *password* is a string 1 to 60 characters long that is used for basic server authentication.
- *port* is the number of an asynchronous port. Ports are numbered sequentially from `asyn0`.
- *filename* is a 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 string 1 to 60 characters long that is used for basic server authentication.

Description This command sets defaults for the [load command on page 5-31](#) and [upload command on page 5-62](#). All values that can be specified with the **load** and **upload** commands can also be specified as defaults with the **set loader** command. Parameters not specified in the **load** and **upload** commands use this default.

All parameters except **destfile**, **srcfile**, and **file** can be returned to their defaults with the **default** option.

The **asyn** parameter specifies the asynchronous port via which the file is to be downloaded using **zmodem** or **none**, or uploaded using **zmodem**. If **default** is specified, previous defaults are cleared and the parameter is set to no ASYN.

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. If **default** is specified, this parameter is set to **cert**.

Keyword	Object type
cert	userCertificate
crl	certificateReservationList
cacert	cACertificate

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, no delay is set.

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 to store the file. 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. Release files, patch files, and script files may be stored in flash. If **destination** is not specified and has not already been set with the [set loader command on page 5-42](#), the default is flash.



Caution 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 **keyid** parameter specifies the ID number of a RSA private key that is held on the router. This parameter is only valid when loading or uploading using SCP. The server receiving the load request must have the public key for this authentication to work. If **default** is specified, the previous default is cleared and server authentication is not used.

The **method** parameter specifies the method used to download the file. If **http** is specified, HTTP downloads the file. The **web** and **www** options are synonyms for HTTP. If **ldap** is specified, LDAP downloads the file. If **tftp** is specified, TFTP downloads the file. If **scp** is specified, SCP is the default method for loading and uploading.

If **zmodem** is specified, the ZMODEM protocol downloads 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 system. 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, LDAP or SCP 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 **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 filename. The only restriction is that the last part of the parameter must be a valid filename for the loader. When **method** is set to **tftp**, **http**, **zmodem**, or **none**, valid filenames have the format *filename.ext* and can be 1 to 32 characters long. The following are examples of valid filenames 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 filename and working backwards, the first character not valid in filenames delimits a valid filename 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 filenames 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 HTTP, SCP 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 22-162 of Chapter 22, Internet Protocol \(IP\)](#) for more information about setting up name servers. The [ping command on page 22-135 of Chapter 22, 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 SCP, 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 parameters to download release 55281-01.rez to flash from the TFTP server with IP address 172.16.8.5, with a delay of one minute, use the command:

```
set loa fi=55281-01.rez des=flash se=172.16.8.5 del=60
```

To clear defaults previously set with the **set loader** command (except the filename), and restore the factory defaults, use the command:

```
set loa asy=def att=def del=def des=def http=def key=def
met=def pas=def pro=def se=def servp=def usern=def
```

Related Commands [load](#)
[reset loader](#)
[show loader](#)
[upload](#)

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 B-2 of [Appendix B, 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 specific software module. It requires a user with security officer privilege when the router is in security mode.

If no optional parameters are specified, the current default configuration file (set with the [set config command on page 5-38](#)) is displayed, along with information about how the current configuration was obtained ([Figure 5-1, Table 5-1](#)).

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 5-1: Example output from the **show config** command

```
Boot configuration file: boot.cfg (exists)
Current configuration: boot.cfg
```

Table 5-1: Parameters in output of the **show config** command

Parameter	Meaning	
Boot configuration file	Not set	Boot configuration file has not been set
	<filename> (exists)	Boot configuration file has been set to <filename> and it exists.
	<filename> (doesn't exist)	Boot configuration file has been set to <filename> but it does not exist.
Current Configuration	Source of the current configuration:	
	None	The router started with no configuration because one was not set, a valid CFG file 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 model without NVS); or the user entered "S" in response to the prompt during startup.
	<filename> (warm start)	The router started using <filename>, but this was a warm restart (restart router conf=<filename>).

Table 5-1: Parameters in output of the **show config** command (cont)

Parameter	Meaning
None (file not found)	The router started with no configuration because the required 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 started from the <filename> configuration file. This is the typical 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.
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 forces this configuration, and the other runs the special configuration when a valid configuration file is not found (either one set or boot.cfg).
<file> (default)	The router started from the default configuration file because a configuration file was not set. The router looks for the file in flash memory.

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 routed protocol, use the command:

```
sh con dyn=ipx
```

Related Commands [restart](#)
[create config](#)
[set config](#)

show feature

Syntax SHow FEAture [= { *featurename* | *index* }]

where:

- *featurename* is a string 1 to 12 characters long. Valid characters are any printable character.
- *index* is a decimal number in the range from 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 5-2, Table 5-2). If a special feature licence name or index is specified, detailed information about it is displayed (Figure 5-3 on page 5-49, Table 5-3 on page 5-49). This command requires a user with security officer privilege when the router is in security mode.

Figure 5-2: 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 2004 to 16 SEP 2004
3      Test2        password incorrect

The current valid features:

Triple DES Encryption
SW Compression
  
```

Table 5-2: Parameters in 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 5-27 .
Licence	Whether the licence is full or a 30-day trial. A password error is displayed if there is a mismatch between the software being licenced and the serial number of the router.
Period	Timeframe for which the trial licence is valid.
The current valid features	List of the special features enabled by this licence.

Figure 5-3: Example output from the **show feature** command for a specific special feature licence

```
The special feature licence : ENCO
Licence Type                : full
Period                      : -

The included features       : 3des Encryption
```

Table 5-3: Parameters in output of the **show feature** command for a specific special feature licence

Parameter	Meaning
The special feature licence	Name assigned to the special feature licence with the enable feature command on page 5-27.
Licence Type	Whether the licence is full or a 30-day trial. A password error is displayed if there is a mismatch between the software being licenced and the serial number of the router.
Period	Timeframe for which the trial licence is valid.
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 release](#)

show http client

Syntax SHow HTTP CLient

Description This command displays the current state of the HTTP client (Figure 5-4, Table 5-4).

Figure 5-4: Example output from the **show http client** command

```

HTTP Client
-----
Sessions opened ..... 1
Sessions closed ..... 1
Transmitted requests ..... 1
Received replies ..... 1
-----

```

Table 5-4: Parameters in 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 the client has transmitted.
Received replies	Number of HTTP responses the client has received.

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 5-5). For possible debug modes, see the **enable http debug** command.

Figure 5-5: Example output from the **show http debug** command

```
Enabled Debug Modes
-----
AUTH, MSG
-----
```

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 5-6, Table 5-5).

Figure 5-6: 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 5-5: Parameters in 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; 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 the server received.
Unknown requests	Number of unrecognised HTTP requests the server received.
Transmitted replies	Number of HTTP responses the server transmitted.
Aborted replies	Number of HTTP replies the server aborted.
Transmitted replies on bad session	Number of HTTP replies the server transmitted 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 5-7, Table 5-6).

Figure 5-7: 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 5-6: Parameters in 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	Status 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, patch file used, and GUI resource file if applicable (Figure 5-8, Table 5-7).

If the selected GUI resource file fails to pass validation checks on boot up, described under the [set install command on page 5-40](#), the 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 5-8: Example output from the **show install** command after a new release file is installed

```

Install      Release          Patch          GUI
-----
Temporary   -                -              -
Preferred   flash:55-291.rez -              750s_291-00_en_d.rsc
Default     EPROM (PR1-1.1.0) -              -
-----

Current install
-----
Preferred   flash:55-291.rez -              750s_291-00_en_d.rsc
-----

Install history
-----
No Temporary release selected
Preferred release selected
Preferred release successfully installed
Preferred GUI successfully installed

```

Table 5-7: Parameters in output of the **show install** command

Parameter	Meaning
Install	Types of files available to run: temporary, preferred, or default.
GUI	GUI resource file installed and currently used, if any. For models with a GUI, the filename is displayed regardless of whether the GUI is enabled.
Release	Release filename used.
Patch	Patch filename used.
Dmp	Third-party data manipulation program for the install, if any. This is not present on most models and software releases.
Current install	Names of files currently running.
Install history	A list of checks carried out during the install boot. The list shows how the current install was selected and loaded.

Related Commands [delete install](#)
[set install](#)
[show system](#)

show ldap

Syntax SHow LDAP [DEBug]

Description This command summarises information about the LDAP module (Figure 5-9, Table 5-8).

If **debug** is specified, debug status for the LDAP module is displayed.

Figure 5-9: 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 5-8: Parameters in 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 LDAP module generated the request internally
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 5-10, Table 5-9). If the **request** parameter is specified with the identification number of an open request, information is displayed for that request.

Figure 5-10: 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 5-9: Parameters in output of the **show ldap request** command

Parameter	Meaning
Schema	LDAP Schema under which the request was made.
Operation	Whether the operation requested under the schema is 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
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 49-4 of Chapter 49, Public Key Infrastructure (PKI).

Table 5-9: Parameters in output of the **show ldap request** command (cont)

Parameter	Meaning
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 objects 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 also (False).
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 defaults for the loader and the progress of the current load (Figure 5-11, Table 5-10).

Figure 5-11: Example output from the **show loader** command

```

Loader Information
-----
Defaults:
Method..... TFTP
File ..... /netupgrades/new.cfg
Destination File.... -
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/ospf.cfg
Destination File.... -
Server ..... www.company.com (192.168.163.22)
TCP Port ..... 80
Destination ..... Flash
Delay (sec) ..... 0
Status ..... Loading
Load Level ..... 8%
-----

```

Table 5-10: Parameters in output of the **show loader** command

Parameter	Meaning
Defaults	Defaults used as parameters not specified in the load and upload commands.
Current Load	Values currently being used to load a file to or from the router.
Last Load	Values last used to load a file to or from the router.
Method	Method used to load files, one of: LDAP, SCP, TFTP, HTTP, WEB, WWW, ZMODEM, or None.
File	Name of the file being copied.
Destination File	Name assigned to the new file copy. This defaults to the original file name if not specified.
Server	IP address or host name of the server. Used when method is set to SCP, TFTP or HTTP.
HTTP Proxy	IP address or host name of the proxy server when method is set to HTTP and access is via a proxy server.
Proxy Port	The port number of the proxy server, if the load method is HTTP and httpproxy is specified.
TCP Port	TCP port used during a load using HTTP.

Table 5-10: Parameters in output of the **show loader** command (cont)

Parameter	Meaning
LDAP BaseobjectDN	The repository location of the object to load if using LDAP. Displays only if LDAP is the default load method.
LDAP Username	The LDAP username set. Displays only if LDAP is the default load method.
LDAP Password	The LDAP password set. Displays only if LDAP is the default load method.
LDAP Attribute	The type of object being retrieved from an LDAP repository. Displays only if LDAP is the default load method.
Username	The username set for the load or upload. This only displays if a username is set.
Asyn	The asynchronous port via which the file is being loaded using zmodem or none, or uploaded using zmodem.
Destination	Where the file is stored, or will be stored
Delay	The delay, in seconds, between initiating the file download and the download actually starting.
Status	Current status of the load or upload.
Load Level	The percentage of the file transfer completed.
Last Message	Last error or informational message sent to the device where the last load command was issued. At router boot up, the Last Message is undefined and displays a dash. This is not displayed when the loader status is "Loading".

Related Commands

- [load](#)
- [set loader](#)
- [upload](#)

show patch

Syntax SHow PATch

Description This command displays all patch files stored in NVS and flash memory (Figure 5-12, Table 5-11).

Figure 5-12: 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 5-11: Parameters in 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 version to which the patch applies, followed by a hyphen, and the generation number of the patch itself.

Related Commands [load](#)

show release

Syntax SHow RELEase

Description This command shows the release licence information in the router (Figure 5-13, Table 5-12). All releases that have a licence are displayed, along with the status of the licence.

Figure 5-13: 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 5-12: Parameters in output of the **show release** command

Parameter	Meaning
Release	Full name of the release file.
Licence	Whether the licence is full or a 30-day trial.
Period	Period of the licence when it is a 30-day trial.

Related Commands [disable release](#)
[enable release](#)

upload

Syntax UPLoad [METHod=SCP] [DESTFile=*path/destfilename*]
 [File=*filename*] [{KEYid=*key-id*|PASSword=*password*}]
 [Server={*hostname*|*ipadd*|*ipv6add*}] [USERName=*username*]

UPLoad [METHod=TFTP] [DESTFile=*path/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 string up to 40 characters long.
- *key-id* is a decimal ID number of an encryption key used for authentication.
- *password* is a string 1 to 60 characters long that is used for basic server authentication.
- *port* is the number of an asynchronous port. Ports are numbered sequentially starting with asyn 0.
- *username* is a string 1 to 60 characters long that is used for basic server authentication.
- *path* is a character string specifying the destination file location path on the server. The combination of path and filename can be a maximum of 256 characters long.
- *destfilename* is a string 1 to 256 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 SCP, TFTP or ZMODEM. This command requires a user with security officer privilege when the router is in security mode.

Any parameters that are not specified use the defaults set with the [set loader command on page 5-42](#). Some parameters are invalid or have different meanings depending on the method that downloads 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 5-42](#).

The **destfile** parameter specifies the name that the file is saved under in the destination file system. For SCP and TFTP, you can specify a path as well as a filename.

The **file** parameter specifies the name of the file on the router's file system and should be a fully qualified filename, including the device name. This parameter is required unless it was already set with the [set loader command on page 5-42](#).

The **keyid** parameter specifies the ID number of a RSA private or public key that is held on the router. This parameter is only valid when uploading using SCP. The server receiving the upload request must have the public key for this authentication to work.

The **method** parameter specifies the method that uploads the file. If **tftp** is specified, TFTP uploads the file. If **method** is **tftp**, the **file** and **server** parameters are required unless they were already set with the [set loader command on page 5-42](#). If **scp** is specified, SCP uploads the file. When **scp** is specified, the **username** parameter must be set, along with either the **password** or **keyid** parameter. If **zmodem** is specified, the ZMODEM protocol uploads the file. When **zmodem** is specified, the **asyn** parameter is required, but may have been already set with the [set loader command on page 5-42](#). 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**, unless another method has been specified using the [set loader command on page 5-42](#).

The **password** parameter specifies the password for server authentication, if RSA authentication is not being used. This parameter is only valid when uploading using SCP. As the password is typed it appears as plain text on the screen, so it should only be used in a secure area.

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 translates this to an IP address. See the [set ip nameserver command on page 22-162 of Chapter 22, Internet Protocol \(IP\)](#) for more information about setting up name servers. Use the [ping command on page 22-135 of Chapter 22, Internet Protocol \(IP\)](#) 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 5-42](#). The **server** parameter cannot be used when **method** is **zmodem**.

The **username** parameter specifies the username needed when uploading using SCP.

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 file debug.txt to a SCP server with the IP address 172.16.8.5, use the command:

```
upl met=scp fo=debug.txt destf=/tmp/debug.txt se=172.16.8.5
    usern=username password=password
```

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](#) in Chapter 6, Managing the File System
[show loader](#)

