

## Release Note

# Software Release 2.6.4

## For AR400 & AR700 Series Routers

Introduction .....	4
Upgrading to Software Release 2.6.4 .....	4
Hardware Platforms .....	5
Overview of New Features .....	5
UPnP .....	6
Importing BGP routes into OSPF .....	7
Enabling BGP Route Import .....	7
Limiting the Number of Routes .....	7
Advertising Desired Routes .....	8
Configuration Example .....	8
Authenticating OSPF .....	8
Password Authentication .....	9
Cryptographic Authentication .....	9
add ospf md5key command .....	10
delete ospf md5key command .....	11
show ospf md5key command .....	12
SNMPv3 .....	13
ICMP Router Discovery Advertisements .....	13
Router Discovery Process .....	13
Router Advertisement Messages .....	14
Router Solicitation Messages .....	14
Router Advertisement Interval .....	14
Preference Level .....	15
Lifetime .....	15
Configuration Example .....	15
Support for Long File Names .....	16
Upgrading to new software releases .....	16
Regressing to Previous Software Releases .....	17
Interrupting Text Flow with the CLI .....	17
Specifying the Mode of Operation When IGMP Snooping is Enabled .....	18
IP Route Filter Changes to	
Protocol Parameter .....	19
Remote Security Officer (RSO) Login .....	19
Enable IPV6 MLD Interface Command .....	20
Text Message at Login .....	20
Priority-Based Routing .....	21
Probing IP Addresses .....	21
Valid Values for IPv6 Router Advertisement AdvRetransTimer .....	22
Valid Characters for File Names -	
Show File and Delete File Commands .....	22

Extended show debug Command .....	23
Email Relaying .....	25
Extended syslog format Parameter .....	26
TACACS+ Authentication and Telnet .....	27
TPAD Numeric Response .....	28
VLAN Tagging on ETH Interfaces .....	29
Modified Commands .....	29
Example .....	31
Adopting the VRRP IP Address .....	32
Benefits of VRRP IP Address Adoption .....	32
Risks of VRRP IP Address Adoption .....	32
Recommendations .....	32
Configuration of VR IP Address Adoption .....	33
Firewall Enhancements .....	33
ICMP Protocol for Firewall Policy Rule .....	33
Debug and Display Firewall ARP Requests .....	34
PPTP Pass Through .....	35
802.1x Port Authentication .....	37
Support for 802.1x .....	38
IPsec NAT-Traversal .....	39
Basic NAT-T Operations .....	40
NAT-T on the Router .....	40
Commands Modified for NAT-T .....	42
NAT-T Configuration Example .....	45

## Introduction

Allied Telesyn announces the release of Software Release 2.6.4 for the AR400 and AR700 Series routers. To see which new features and enhancements apply to each product type, see *Overview of New Features on page -4*.

This Release Note describes:

- Important factors you need to consider when upgrading to Software Release 2.6.4 from an earlier software release, in *Upgrading to Software Release 2.6.4 on page -3*
- The names of the software release, GUI and help files for this release, in *Upgrading to Software Release 2.6.4 on page -3*
- The hardware platforms supported by Software Release 2.6.4
- The new features in Software Release 2.6.4 since Software Release 2.6.1

This Release Note should be read in conjunction with the Quick Install Guide, User Guide, Hardware Reference, and Software Reference for your router. These documents are on the Documentation and Tools CD-ROM packaged with your router, or at

[www.alliedtelesyn.co.nz/documentation/documentation.html](http://www.alliedtelesyn.co.nz/documentation/documentation.html)



*WARNING: Information in this Release Note is subject to change without notice and does not represent a commitment on the part of Allied Telesyn International. While every effort has been made to ensure that the information contained within this document and the features and changes described are accurate, Allied Telesyn International can not accept any type of liability for errors in, or omissions arising from the use of this information.*

## Upgrading to Software Release 2.6.4

Software Release 2.6.4 is available as a flash release that can be downloaded directly from the Software Updates area of the Allied Telesyn web site at: [www.alliedtelesyn.co.nz/support/updates/patches.html](http://www.alliedtelesyn.co.nz/support/updates/patches.html)

Software releases must be licenced and require a password to activate. To obtain a licence and password, contact your authorised Allied Telesyn distributor or reseller.

The files in this software release are shown in Table 1.

Table 1: File names for Software Release 2.6.4

Product name	Software release file	GUI resource file	CLI help file
AR410 and AR410S	52-264.rez	d_410e10.rsc	410-264A.HLP
AR450S	54-264.rez	d450se10.rsc	450-264A.HLP
AR720	52-264.rez	r_720e09.rsc	700-264A.HLP
AR725	52-264.rez	d_725e10.rsc	700-264A.HLP
AR740	52-264.rez	r_740e09.rsc	700-264A.HLP
AR745	52-264.rez	d_745e10.rsc	700-264A.HLP

## Hardware Platforms

Software Release 2.6.4 supports the following hardware platforms:

- AR410, AR410S, and AR450S routers
- AR700 Series routers

## Overview of New Features

Table 2 summarises the new features and enhancements in Software Release 2.6.4 by product series. Each one is described in the following sections.

Table 2: New features and enhancements in Software Release 2.6.4 by product series

	AR410	AR450S	AR700
UPnP	✓	✓	
Importing BGP routes into OSPF	✓	✓	✓
Authenticating OSPF	✓	✓	✓
SNMPv3	✓	✓	✓
ICMP Router Discovery Advertisements	✓	✓	✓
Support for long file names (DOS 28.3)	✓	✓	✓
Interrupting text flow with the CLI	✓	✓	✓
Specifying the mode of operation when IGMP Snooping is enabled	✓	✓	✓
IP route filter changes to <b>protocol</b> parameter	✓	✓	✓
Remote Security Officer (RSO) login	✓	✓	✓
New <b>enable ipv6 mld interface</b> command	✓	✓	✓
Text Message at Login	✓	✓	✓
Priority-based routing	✓	✓	✓
Probing IP Addresses	✓	✓	✓
Valid values for IPv6 Router Advertisement AdvRetransTimer	✓	✓	✓
Valid characters for file names - <b>show file</b> and <b>delete file</b> commands	✓	✓	✓
Extended <b>show debug</b> command	✓	✓	✓
Email relaying	✓	✓	✓
New <b>syslogformat</b> parameter - <b>create log output</b> and <b>set log output</b> commands	✓	✓	✓
TACACS+ Authentication and Telnet	✓	✓	✓
TPAD numeric response	✓		✓
VLAN Tagging on ETH Interfaces	✓	✓	✓
Adopting the VRRP IP address	✓	✓	✓

Table 2: New features and enhancements in Software Release 2.6.4 by product series

	AR410	AR450S	AR700
Firewall Enhancements	✓	✓	✓
PPTP Pass Through	✓	✓	✓
802.1x port authentication		✓	
IPsec NAT-Traversal	✓	✓	✓



*802.1x port authentication was initially released in Software Release 2.6.1 for the AR410, AR410S, and the AR700.*

## UPnP

UPnP is an architecture that allows for dynamic connectivity between devices on a network. Devices may dynamically add themselves to a network without the need for user intervention or configuration. UPnP-enabled devices in a network can engage in real-time applications such as instant messaging.

The intent of UPnP is to support zero configuration, "invisible" networking of devices including intelligent appliances, PCs, printers, and other smart devices using standard protocols. A UPnP-enabled device may obtain an IP address, advertise its capabilities, learn about other connected UPnP devices and then communicate directly with those devices. The same device can terminate its connection cleanly when it wishes to leave the UPnP community.

When UPnP is enabled, the router acts as an Internet Gateway Device (IGD). The IGD manages UPnP sessions between the private LAN and the public WAN over a firewall policy. The firewall policy can have one or more LAN interfaces and only one WAN interface. Because peers on the public side cannot initiate sessions or connections across the firewall by default, *control points* on the private side can create port maps for these sessions.



*To use UPnP the firewall must be available on your router.*

For more information about this implementation of UPnP, see:

- the *UPnP* chapter in the *Software Reference*
- Technical Note, *A Technical Guide to UPnP* at [www.alliedtelesyn.co.nz/support/technotes/tn-index.html](http://www.alliedtelesyn.co.nz/support/technotes/tn-index.html)
- Configuration Note, *Configuring the AR450S as a UPnP Internet Gateway Device with a Windows® XP® Machine as a UPnP Control Point* at [www.alliedtelesyn.co.nz/solutions/solutions.html](http://www.alliedtelesyn.co.nz/solutions/solutions.html)

## Importing BGP routes into OSPF

With this enhancement you can import routes from BGP into OSPF. OSPF will then redistribute these routes. This enhancement adds three parameters to the **set ospf** command, and modifies the output of the **show ospf** command. The new parameters are **bgpimport**, **bgpfilter** and **bgplimit**.

BGP can learn thousands of routes, so it's important to consider the network impact of importing these routes. Routing devices in the OSPF domain may become overloaded if they store too many routes. You can prevent this by limiting the number of routes that will be imported.



*Do not enable the importing of BGP routes into OSPF unless you are sure about the consequences for the OSPF domain.*

### Enabling BGP Route Import

To enable importing BGP routes into OSPF, use the command:

```
set ospf bgpimport=on
```

### Limiting the Number of Routes

There are two ways to limit the number of BGP routes imported into OSPF. One way is to specify a maximum number of routes with the command:

```
set ospf bgplimit=1...300
```

When the limit is reached, the importing of routes will stop until existing routes are removed. Because they are BGP routes, actions of BGP control when the routes disappear.

The other way to limit the imported routes is to configure a routing filter. This filter is used in conjunction with the **bgpfilter** parameter in the **set ospf** command to control the passing of routing information in and out of the device. To configure a filter, use the **add ip filter** command:

```
add ip filter=filter-number {action=include|exclude}  
source=ipadd [smask=ipadd] [entry=entry-number]
```

Use this filter to limit imported BGP routes with the command:

```
set ospf bgpfilter=300...399
```

where the filter number is the previously configured filter.

Take care when configuring the IP filter. If the number of imported routes reaches the **bgplimit** parameter, you may not have imported all the routes specified with the **bgpfilter** parameter.

## Advertising Desired Routes

The order in which routes are added is arbitrary. This means that to have desired BGP routes advertised by OSPF, you must take care setting the **entry** number for the route filter with the **add ip route** command. Assign a low entry number to a filter used to import preferred BGP routes. Alternatively, set the **bgplimit** parameter above the total number of routes that BGP will ever add to the routing table.

## Configuration Example

This example supposes that you want to import the route 192.168.72.0 into the OSPF routing domain, but no other routes. This route is received on the gateway router as a BGP route. The following steps show the sequence of commands to use in this scenario.

1. Set up the IP filter:

```
add ip filter=300 source=192.168.72.0 smask=255.255.255.255
    action=include
```

2. Set up OSPF BGP import parameters:

```
set ospf bgpimport=on bgpfilter=300 bgplimit=1
```

3. Check that BGP has added the route to the IP route table:

```
show ip route=192.168.72.0
```

The route should be visible in the output of the command.

4. Check that OSPF has imported the route:

```
show ospf lsa=192.168.72.0
```

The output should show that there is an AS external LSA with this ID.

## Authenticating OSPF

---

You can authenticate OSPF packets as described in Appendix D of RFC 2328. See this RFC for a detailed description of how OSPF packets are authenticated.

An authentication type can be chosen for each interface. RFC 1583 states that authentication is configured per area, but RFC 2328 states that authentication is configured per interface. This implementation of OSPF authentication represents a compromise for these two solutions. An authentication type is set up per area to act as a default for all interfaces in the area. The default setting for interfaces is to use the area default, but each interface can be individually set to any authentication method.

There are two ways to authenticate an OSPF packet:

- simple password authentication
- cryptographic authentication with MD5

## Password Authentication

Password authentication can be configured for OSPF areas to authenticate incoming packets. The password can be up to 8 characters long, and is configured for each interface.

To configure an OSPF area with password authentication, use the command:

```
add ospf area={backbone|area-number} authentication=password
```

The password itself is configured on a per-interface basis with the **add ospf interface** and **set ospf interface** commands.

To configure an OSPF interface with password authentication, use the command:

```
add ospf interface=interface authentication=password  
password=password
```

Valid characters for the password are any printable character. If the password contains spaces it must be enclosed in double quotes

For password authentication to succeed, you need to configure all interfaces in the same physical network with the same password.

## Cryptographic Authentication

An MD5 digest can be appended to the OSPF packet for authentication. The digest is based on the contents of the packet and a shared secret key. The key that you configure must be the same for all interfaces sharing the same physical network. MD5 keys are defined for each interface, and can be up to 16 characters long. The key is case-sensitive, and valid characters are letters and digits only. If authentication is set to MD5 and no key is configured, a default key is created that has an ID of 0 and no key.

For MD5 authentication to succeed, first configure the OSPF area, then the interface, and then add the MD5 key to the interface. You can use the **set ospf area** and **set ospf interface** commands to modify the authentication type.

To configure an OSPF area with MD5 authentication, use the command:

```
add ospf area={backbone|area-number} authentication=md5
```

To configure an OSPF interface with MD5 authentication, use the command:

```
add ospf interface=interface authentication=md5
```

To add an MD5 key that will be used for interface authentication, use the command:

```
add ospf md5key=key id=1...255 interface=interface
```

Normally, only one MD5 key is added at a time for any given OSPF interface. We recommend changing the MD5 key every month, and periodically deleting old and inactive keys. When changing MD5 keys, add the new key on all routers in the network. While the keys are being added, routers will send duplicate messages using both keys. The old key becomes inactive when all interfaces transmit packets using the new key. The packets will not be duplicated using the inactive key. The output of the **show ospf md5key** command shows whether a key is active or inactive.

## Deleting MD5 Keys

You can delete a MD5 key when it has become inactive or when it is being used. You may want to delete a key that is currently active if an illicit router is using the key. To delete an active key immediately, specify the **force** parameter in the **delete ospf md5key** command.



---

*Force deleting an active MD5 key may lead to partial network failure if succeeding keys are not configured on all interfaces in the physical network.*

---

Before deleting a key, configure a succeeding key on all interfaces in the physical network. You can delete the previously active key without using the FORCE parameter. The new key can take over authentication duties when the active key is deleted.

## add ospf md5key command

**Syntax** `ADD OSPF MD5KEY=key ID=1...255 INTERFACE=interface`

where:

- *key* is a character string, 1-16 characters in length and case sensitive. Valid characters are letters and digits only: a-z, A-Z, 0-9.
- *interface* is a valid interface name.

**Description** This command adds an MD5 key for use when authenticating OSPF packets on a particular interface. For MD5 authentication to succeed, all key and ID values must be identical on all interfaces in the same physical network.

The **md5key** parameter specifies the 1-16 character key used for MD5 authentication. The key is entered as alphanumeric characters and is case sensitive. The more characters in the key, the greater the security it offers. We recommend that the MD5 key is at least 9 characters long and is changed every month.

The **id** parameter specifies the identification number for this key. The ID is used in the authentication of packets to identify to the remote device which key is being used in this packet.

The **interface** parameter specifies the OSPF interface with which this key is associated. Each interface has its own set of keys, and keys must be identified by interface as well as key ID. The interface can be any OSPF interface for which MD5 authentication is required.

**Example** To add an MD5 authentication key called "mj48dhw05" with an ID of 3 to interface vlan1, use the command:

```
add ospf md5key=mj48dhw05 id=3 interface=vlan1
```

**Related Commands**

- add ospf interface
- delete ospf md5key
- show ospf md5key

## delete ospf md5key command

**Syntax** DELETE OSPF MD5KEY ID=1..255 INTERFACE=*interface* [FORCE]

where *interface* is a valid interface name

**Description** This command deletes an MD5 key used for authenticating OSPF packets on a particular interface. If the key is being used when the command is executed, the command will fail unless the **force** parameter is also specified. Before deleting a key, configure a succeeding key on all interfaces in the physical network. You can delete the previously active key without using the **force** parameter. The new key can take over authentication duties when the active key is deleted. The output of the **show ospf md5key** command shows whether a key is active or inactive.

The **id** parameter specifies the identification number of the key that will be deleted.

The **interface** parameter specifies the OSPF interface with which this key is associated. Each interface has its own set of keys, and keys must be identified by interface as well as key ID.

The **force** parameter specifies that the MD5 key should be deleted even when it is being used. If an illicit router is using the key, using the FORCE parameter will ensure that the key is deleted.



---

*Force deleting an active MD5 key may lead to partial network failure if succeeding keys are not configured on all interfaces in the physical network.*

---

**Example** To delete a key with the ID 35 that was used last week and should be replaced, use the command:

```
delete ospf md5key id=35 interface=vlan1
```

**Related Commands** add ospf md5key  
show ospf md5key

## show ospf md5key command

**Syntax** SHOW OSPF MD5KEY [INTERFACE=*interface*]

where *interface* is a valid interface name

**Description** This command displays information about the OSPF MD5 keys for all interfaces, or for a specific interface (Figure 1, Figure 2, Table 3). Each interface has its own set of keys.

The **interface** parameter specifies the OSPF interface for the MD5 keys to be displayed.

Figure 1: Example output from the **show ospf md5key** command

OSPF MD5 keys			
Interface	ID	Key	Active
vlan1	1	O3jf87Pls	No
	2	xm39s2F28	Yes
vlan2	3	ba2958d2x	Yes

Figure 2: Example output from the **show ospf md5key interface** command

OSPF MD5 keys			
Interface	ID	Key	Active
vlan1	1	O3jf87Pls	No
	2	xm39s2F28	Yes

Table 3: Parameters in the output of the **show ospf md5key** command

Parameter	Meaning
Interface	The OSPF interface to which the keys belong
ID	The key's identification number
Key	The MD5 key
Active	Whether or not the key is currently being used to authenticate packets being received from one or more neighbours.

**Example** To show the MD5 keys for the vlan1 interface, use the command:

```
show ospf md5key interface=vlan1
```

**Related Commands** add ospf md5key  
delete ospf md5key

## SNMPv3

---

SNMPv3 provides enhanced security management features whilst maintaining compatibility with earlier versions SNMPv1 and SNMPv2. The basic additional features of version 3 are:

- Message Authentication:
- Hashing and time stamping is employed to ensure that messages are received from valid sources.
- Message Confidentiality
- Encryption can be applied to messages to ensure content privacy.
- Compatibility with previous versions SNMPv1 and SNMPv2

For more information, see the *SNMP* chapter in the *Software Reference*.

## ICMP Router Discovery Advertisements

---

This release supports all of *RFC 1256, ICMP Router Discovery Messages, 1991* as it applies to routers. If this feature is configured, the router sends router advertisements periodically and in response to router solicitations. It does not support the Host Specification section of this RFC.

Before an IP host can send an IP packet, it has to know the IP address of a neighbouring router that can forward it to its destination. ICMP Router Discovery messages allow routers to automatically advertise themselves to hosts. Other methods either require someone to manually keep these addresses up to date, or require DHCP to send the router address, or require the hosts to be able to eavesdrop on whatever routing protocol messages are being used on the LAN.

### Router Discovery Process

For a summary of the processes that occur when Router Discovery advertisements are enabled for interfaces on the router see Table 4.

Table 4: Router Discovery Process

When ...	Then ...
Router Discovery advertising starts on a router interface because: <ul style="list-style-type: none"> <li>- the router starts up, or</li> <li>- advertisements are enabled on the router or on an interface</li> </ul>	the router multicasts a router advertisement and continues to multicast them periodically until router advertising is disabled.
a host starts up	the host may send a router solicitation message.
the router receives a router solicitation	the router multicasts an early router advertisement on the multicast interface on which it received the router solicitation.
a host receives a router advertisement	the host stores the IP address and preference level for the advertisement lifetime.

Table 4: Router Discovery Process (Continued)

When ...	Then ...
the lifetime of all existing router advertisements on a host expires	the host sends a router solicitation.
a host does not receive a router advertisement after sending a small number of router solicitations	the host waits for the next unsolicited router advertisement
a host needs a default router address	the host uses the IP address of the router or L3 switch with the highest preference level.
Router Discovery advertising is deleted from the physical interface ( <b>delete ip advertise</b> command), or the logical interface has <b>advertise</b> set to <b>no</b> ( <b>set ip interface</b> command)	the router multicasts a router advertisement with the IP address(es) that stopped advertising, and a lifetime of zero (0). It continues to periodically multicast router advertisements for other interfaces.
the router receives a router advertisement from another router	the router does nothing but silently discards the message.

## Router Advertisement Messages

A *router advertisement* is an ICMP (type 10) message containing:

- In the destination address field of the IP header, the interface's configured advertisement address, either 224.0.0.1 (ALL) or 255.255.255.255 (LIMITED).
- In the lifetime field, the interface's configured advertisement lifetime.
- In the Router Address and Preference Level fields, the addresses and preference levels of all the logical interfaces that are set to advertise.

## Router Solicitation Messages

A *router solicitation* is an ICMP (type 10) message containing:

- Source Address: an IP address belonging to the interface from which the message is sent
- Destination Address: the configured Solicitation Address, and
- Time-to-Live: 1 if the Destination Address is an IP multicast address; at least 1 otherwise.

## Router Advertisement Interval

The router advertisement *interval* is the time between router advertisements. For the first few advertisements sent from an interface (up to 3), the router sends the router advertisements at intervals of at most 16 seconds. After these initial transmissions, it sends router advertisements at random intervals between the minimum and maximum intervals that the user configures, to reduce the probability of synchronization with the advertisements from other routers on the same link. By default the minimum is 450 seconds (7.5 minutes), and the maximum is 600 seconds (10 minutes).

## Preference Level

The *preference level* is the preference of the advertised address as a default router address relative to other router addresses on the same subnet. By default, all routers and layer 3 switches have the same preference level, zero (0). While it is entered as a decimal in the range -2147483648..2147483647, it is encoded in router advertisements as a twos-complement hex integer in the range 0x8000000 to 0x7ffffff. A higher PREFERENCELEVEL is preferred over a lower value.

## Lifetime

The *lifetime* of a router advertisement is how long the information in the advertisement is valid. By default, the lifetime of all advertisements is 1800 seconds (30 minutes).

## Configuration Example

By default, the router does not send router advertisements.

### To configure the router to send router advertisements:

#### 1. Set the physical interface to advertise.

For each physical interface that is to send advertisements, add the interface. In most cases the default advertising parameters will work well, but you can change them if required. By default, the router sends router advertisements every 7.5 to 10 minutes, with a lifetime of 30 minutes. These settings are likely to work well in most situations, and will not cause a large amount of extra traffic, even if there are several routers on the LAN. If you change these settings, keep these proportions:

```
lifetime=3 x maxadvertisementinterval
minadvertisementinterval=0.75 x maxadvertisementinterval
```

To change these settings, use one of the commands:

```
add ip advertise interface=interface
[advertisementaddress={all|limited}]
[maxadvertisementinterval=4..1800]
[minadvertisementinterval=3..maxadvertisementinterval]
[lifetime=maxadvertisementinterval..9000]

set ip advertise interface=interface
[advertisementaddress={all|limited}]
[maxadvertisementinterval=4..1800]
[minadvertisementinterval=3..maxadvertisementinterval]
[lifetime=maxadvertisementinterval..9000]
```

#### 2. Stop advertising on other logical interfaces.

By default, logical interfaces are set to advertise if their physical interface is set to advertise. If the physical interface has more than one logical interface (IP multihoming), and you only want some of them to advertise, set the other logical interfaces not to advertise by using one of the commands:

```
add ip interface=interface ipaddress={ipadd|dhcp}
advertise=no [other-parameters...]

set ip interface=interface advertise=no
[other-parameters...]
```

### 3. Set preference levels.

By default, every logical interface has the same preference for becoming a default router (mid range, 0). To give a logical interface a higher preference, increase the **preferencelevel**. To give it a lower preference, decrease this value. If it should never be used as a default router, set it to **notdefault**.

```
add ip interface=interface ipaddress={ipadd|dhcp}
    preferencelevel={-2147483648..2147483647|notdefault}
    [other-parameters...]

set ip interface=interface
    [preferencelevel={-2147483648..2147483647|notdefault}]
    [other-parameters...]
```

### 4. Enable advertising.

To enable router advertisements on all configured advertising interfaces, use the command:

```
enable ip advertise
```

### 5. Check advertise settings.

To check the router advertisement settings, use the command:

```
show ip advertise
```

For full descriptions of these commands, see the *Internet Protocol (IP)* chapter in the *Software Reference*.

## Support for Long File Names

File names of up to twenty-eight characters long and extensions of three characters (DOS 28.3 format) are now supported.

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

Table 5: The DOS filename formats supported by different software releases

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

## Upgrading to new software releases

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

## Regressing to Previous Software Releases

If software release 2.6.4 is installed on the router and then a previous software release that supports **only** DOS 8.3 format is installed (see Table 6), long file names that were in DOS 28.3 format are truncated to DOS 8.3 format. When software release 2.6.4 or later is reinstalled, these truncated file names are restored to their DOS 28.3 format and no information is lost. Support for long file names in only DOS 8.3 format is a feature of software releases prior to software release 2.5.1.

If software release 2.6.4 is installed on the router and then a previous software release that supports DOS 16.3 format is installed (see Table 6), long file names in DOS 28.3 format are permanently truncated to DOS 8.3 format. For example, the file `AB12345678.SCP` is permanently renamed `AB123~01.SCP`. Any long file names that were in DOS 28.3 format remain truncated in DOS 8.3 format when software release 2.6.4 is reinstalled. Support for long file names in DOS 16.3 format is a feature of software release 2.5.1 up to software release 2.6.4.

For more information, see the *Operations* chapter in the *Software Reference*.

## Interrupting Text Flow with the CLI

---

A new function has been added for users of the Command Line Interface (CLI) to let them interrupt (or “break”) text paging or continuously streaming text. The key combination is Ctrl-Q.

This capability will be useful with stand alone commands such as **show** commands that display many output screens. The text is buffered and undisplayed text is deleted. The command prompt is then restored.

The paging prompt will continue giving users the option to display the next line of text output or next page, print text continuously with no further prompts, or abort text output.

This functionality will not work on commands that produce output of indeterminate length, such as **enable** and **disable** commands where output starts with *enable* and stops with *disable*.

For more information, see the *Operations* chapter in the *Software Reference*.

## Specifying the Mode of Operation When IGMP Snooping is Enabled

---

You can now specify the mode of operation when IGMP Snooping is enabled with the command:

```
set igmpsnooping routermode=[all|default|ip|multicastrouter|none]
```

If **all** is specified, all reserved multicast addresses (i.e. 224.0.0.1 to 224.0.0.255) are treated as router multicast addresses.

If **default** is specified, the following addresses are treated as router multicast addresses:

- IGMP Query, 224.0.0.1
- All routers on this subnet, 224.0.0.2
- DVMRP Routers, 224.0.0.4
- OSPFIGP all routers, 224.0.0.5
- OSPFIGP designated routers, 224.0.0.6
- RIP2 routers, 224.0.0.9
- All PIM routers, 224.0.0.13
- All CBT routers, 224.0.0.15

If **ip** is specified, you specify addresses treated as router multicast addresses using the **add igmpsnooping routeraddress** and the **delete igmpsnooping routeraddress** commands. When in this mode, your router retains previous addresses that have already been specified.

If **multicastrouter** is specified, the following addresses are treated as router multicast addresses:

- DVMRP Routers, 224.0.0.4
- All PIM routers, 224.0.0.13

If **none** is specified, the router does not create router ports at all.

To add and delete reserved IP multicast addresses to and from the list of router multicast addresses specified by the **set igmpsnooping routermode** command when the **ip** parameter is selected, use the commands:

```
add igmpsnooping routeraddress
delete igmpsnooping routeraddress
```

The IP addresses specified must be from 224.0.0.1 to 224.0.0.255.

To display information about the current list of configured IP multicast router addresses configured on your router, use the command:

```
show igmpsnooping routeraddress
```

For more information about IGMP Snooping, see the *IP Multicasting* chapter in the *Software Reference*.

## IP Route Filter Changes to Protocol Parameter

---

IP routing filters affect the interaction between routing protocols, such as RIP and OSPF, and the IP route table. Route filters control which routes received by routing protocols are added to the IP route table, and which routes in the route table can be advertised by routing protocols.

IP routing filters can no longer be applied to static routes and interface routes. The list of options accepted by the **protocol** parameter in the **add ip route filter** and **set ip route filter** commands has been modified. The new syntax is:

```
add ip route filter[=filter-id] ip=ipadd mask=ipadd
  action={include|exclude} [direction={receive|send|both}]
  [interface=interface] [nexthop=ipadd] [policy=0..7]
  [protocol={any|egp|ospf|rip}]

set ip route filter=filter-id [ip=ipadd] [mask=ipadd]
  [action={include|exclude}] [direction={receive|send|
  both}] [interface=interface] [nexthop=ipadd] [policy=0..7]
  [protocol={any|egp|ospf|rip}]
```

For more information about static routes and interface routes, see the *Internet Protocol (IP)* chapter in the *Software Reference*.

## Remote Security Officer (RSO) Login

---

The Remote Security Officer (RSO) feature lets a remote user connect to a router via Telnet from an authorised IP address, and login using a name with Security Officer privilege as if the user were at a terminal connected directly to the router. The RSO feature is configured by defining authorised IP addresses using the **add user rso** and **delete user rso** commands. These commands now accept ranges of IP addresses:

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

where *ipadd* is an IP address in dotted decimal notation. If a mask is not specified, the default is 255.255.255.255.

IPv6 addresses are now also supported, enabling Remote Security Officers to login over an IPv6 network:

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

where *ipv6add* is an IPv6 address. If a prefix length is not specified, the default is 128.

For more information about Remote Security Officer, see the *Operations* chapter in the *Software Reference*.

## Enable IPV6 MLD Interface Command

---

This command lets users enable the Multicast Listener Discovery (MLD) protocol on an interface that already exists. For Release 2.6.4, the **v1compatible** parameter for the command has been replaced with the **queryversion** parameter. The new syntax is:

```
enable ipv6 mld interface=interface [queryversion={1|2}]
```

where *interface* is a valid interface

**Queryversion** specifies the version of MLD Query to use on the interface. It is a more accurate way to specify interoperability between MLDv2 and MLDv1. The default is 2.

To avoid unnecessary error messages, we recommend that users replace **v1compatible** with **queryversion** along with their related values in scripts currently being used.

For more information about Multicast Listener Discovery, see the *IPv6 Multicasting* chapter in the *Software Reference*.

## Text Message at Login

---

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

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

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

Login:
```

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

For more information to help create a *login.txt* file, see the **edit** command and the **load** command in the *Software Reference*.

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

## Priority-Based Routing

---

An enhancement has been made to priority-based routing so that when you configure priority routing for an ETH or VLAN interface, you can now collect debug statistics. Priority routing is configured with the **add ip filter** and **add ip interface** commands.

Priority routing debug is disabled by default. When enabled, the logical priority queue lengths and queue occupancy statistics are collected. Priority routing debug can be enabled or disabled on a per-interface basis by using the commands:

```
enable interface={ifIndex|interface} [debug=priorityqueue]
disable interface={ifIndex|interface} [debug=priorityqueue]
```

where:

- *ifIndex* is the value of ifIndex for the interface in the interface table
- *interface* is the name of the interface

The status of priority routing on an ETH or VLAN interface and debug information (if enabled) can be displayed with the command:

```
show interface[={ifIndex|interface}] [counter]
[priorityqueue]
```

The **counter** and **priorityqueue** parameters are mutually exclusive.

For more information about priority-based routing, see the *Internet Protocol (IP)* chapter in the *Software Reference*.

## Probing IP Addresses

---

When creating a DHCP range, you can now specify how the DHCP server checks whether an IP address is being used by other hosts by specifying the new **probe** parameter, with the command:

```
create dhcp range=name ip=ipadd number=number policy=name
[gateway=ipadd] [probe={arp|icmp}]
```

The **probe** parameter specifies how the DHCP server checks whether an IP address is being used by other hosts. If **arp** is specified, the server sends ARP requests to determine if an address is in use. If **icmp** is specified, the server sends ICMP Echo Requests (pings). The default is **icmp**.

To modify the server's method for probing IP addresses, use the new command:

```
set dhcp range=name probe={arp|icmp}
```

Note that **arp** cannot be specified if the range includes a gateway (by specifying the **gateway** parameter when it was created), or if the network uses Proxy ARP.

For more information, see the *Dynamic Host Configuration Protocol (DHCP)* chapter in the *Software Reference*.

## Valid Values for IPv6 Router Advertisement AdvRetransTimer

---

The value you enter for the AdvRetrans timer is now rounded up to the nearest 100 milliseconds (for example, 301 becomes 400). The AdvRetrans timer is the interval between repeats of each Router Advertisement message sent by the router, and is specified by using the **retrans** parameter in the command:

```
set ipv6 nd interface=interface retrans=0..4294967295
[other-parameters]
```

The default is 0, which indicates that this timer is not specified.

For more information, see the *Internet Protocol Version 6 (IPv6)* chapter in the *Software Reference*.

## Valid Characters for File Names - Show File and Delete File Commands

---

For the **show file** and **delete file** commands only, the characters \* > [ ] | : can now be specified in the filename.

Files are uniquely identified by a file name in the format:

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

■ *filename* is a descriptive name for the file, and may be one to twenty eight characters long. Invalid characters are " \ ; ? / , < . Valid characters are:

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

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

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

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

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

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

## Extended show debug Command

The command **show debug** displays the output of a list of other **show** commands. A **full** parameter has been added and displays a longer list of commands:

```
show debug [stack|full]
```

The output also depends on the router's security mode and the user's privileges. The possible command list variations are given in Table 6.

The **stack** parameter limits the output to a stack dump, if one is available. The output depends on whether the last fatal condition was a hardware reset or a software reboot. After a software reboot, the output is a stack dump. After a hardware reset, no stack dump information is available and a message to this effect is displayed. If the **stack** parameter is not specified, both a stack dump if available and the output of a list of **show** commands is generated.

Table 6: The list of **show** commands that are executed by the **show debug** command, when the **full** parameter is or is not specified, under different combinations of security mode and privilege level

Full parameter specified?	Security mode	Privilege level	List of commands executed
No	normal	manager	show system
No	secure	security officer	show files show install show feature (AR400, AR700, AT-8800, Rapier and Rapier i series) show release show config dynamic show buffer scan show cpu show log show exception show ffile check
No	secure	manager	show system (without current configuration file) show files show install show release show buffer scan show cpu show log show exception show ffile check

Table 6: The list of **show** commands that are executed by the **show debug** command, when the **full** parameter is or is not specified, under different combinations of security mode and privilege level (Continued)

Full parameter specified?	Security mode	Privilege level	List of commands executed
Yes	normal	manager	show system
Yes	secure	security officer	show files show install show feature (AR400, AR700, AT-8800, Rapier and Rapier i series) show release show config dynamic show interface show ip interface show ip arp show ip route full show ip count show switch (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch counter (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch fdb (AR450, AT-8700XL, AT-8800, Rapier and Rapier i series) show startup show flash show switch port=all (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch port=all counter (AR450, AT-8700XL, AT-8800, Rapier and Rapier i series) show buffer scan show cpu show log show exception show ffile check

Table 6: The list of **show** commands that are executed by the **show debug** command, when the **full** parameter is or is not specified, under different combinations of security mode and privilege level (Continued)

Full parameter specified?	Security mode	Privilege level	List of commands executed
Yes	secure	manager	show system (without current configuration file) show files show install show release show interface show ip interface show ip arp show ip route full show ip count show switch (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch counter (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch fdb (AR450, AT-8700XL, AT-8800, Rapier and Rapier i series) show startup show flash show switch port=all (AR400, AT-8700XL, AT-8800, Rapier and Rapier i series) show switch port=all counter (AR450, AT-8700XL, AT-8800, Rapier and Rapier i series) show buffer scan show cpu show log show exception show ffile check

## Email Relaying

How the SMTP application gateway protects against third party relaying of email has been enhanced.

A third-party mail relay occurs when a mail server processes a mail message where neither the sender or the recipient is a local user. The mail server is an entirely unrelated party to mail processing. If an email originates from the public side of the firewall, the firewall SMTP proxy will reject the email if the address in the "RCPT TO" field has a different domain name to a mail server on the private side of the firewall. If an email originates from the private side of the firewall, the firewall SMTP proxy will reject the email if either:

- domain name in the "MAIL FROM" field is different to domain name specified by the **set firewall policy smtpdomain** command, or
- domain name in the "RCPT TO" field is not consistent with IP address of the IP packet.

Note that for the latter to occur, a DNS server must be must be setup by using the **add ip dns** command, *Internet Protocol (IP)* chapter. If a DNS server is not configured, the proxy will only check the email based on the "MAIL FROM" field.

The firewall SMTP proxy can relay any email that originates from the private side of the firewall. This happens when the IP packets for the email are only destined to the private interface of the firewall. The proxy will forward the email to the final destination specified in the "RCPT TO" field. Note that the relaying function requires that a DNS server is setup using the **add ip dns** command, *Internet Protocol (IP)* chapter.

The behaviour of the **disable firewall policy smtprelay** command has been enhanced. This command disables emails that intend to use the third party relaying mechanism for delivery from passing through the SMTP proxy.

The behaviour of the **enable firewall policy smtprelay** command has been enhanced. This command enables emails which intend to use the third party relaying mechanism to pass through the SMTP proxy.

The behaviour of the **set firewall policy smtpdomain** command has been enhanced. This command sets a domain name for the SMTP proxy. The domain name is normally the same as the SMTP server that is located on the private side of the firewall. The specified domain name is used to compare with either:

- the domains of the destination addresses of all SMTP sessions that originate from the public side of the firewall, or
- the domains of the source addresses of all SMTP sessions that originate from the private side of the firewall.

If the domain name does not match, the firewall concludes that the email is trying to use the third party relay mechanism for delivery. If SMTP relaying is disabled then the session is terminated.

For more information about SMTP proxy and the full syntax of these commands, see the *Firewall* chapter in the *Software Reference*.

## Extended syslog format Parameter

A new parameter **syslogformat** has been added to the **create log output** and **set log output** commands:

```
create log output={temporary|permanent|output-id}
  destination={email|memory|asyn|router|syslog}
  [syslogformat=extended|normal] [other-parameters...]

set log output={temporary|permanent|output-id}
  [syslogformat=extended|normal] [other-parameters...]
```

The **syslogformat** parameter specifies whether log messages sent to the syslog server contain the date, time, and system name. If the parameter is set to **extended**, the date, time and system name are included. If the parameter is set to **normal**, the date, time and system name are not included. This parameter is valid when **destination** is **syslog**. The default is **normal**.

Figure 3: Examples of syslog messages with **syslogformat=normal**

```
<12>SSH:SSH/ACPT, SSH connection accepted - pwduser
<14>CH:CMD/USER, logoff
<12>USER:USER/LOFF, pwduser logoff on TTY17
```

Figure 4: Examples of syslog messages with **syslogformat= extended**

```
23-Oct-2003 16:39:37 <12>SSH:SSH/ACPT, Src: AR450 ,SSH connection accepted - pwduser
23-Oct-2003 16:39:41 <14>CH:CMD/USER, Src: AR450 ,logoff
23-Oct-2003 16:39:41 <12>USER:USER/LOFF, Src: AR450 ,pwduser logoff on TTY17
```

To set the system name to a unique identifier, use the command **set system name**.

## TACACS+ Authentication and Telnet

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

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

The value **none** is the default and disables telnet for all TACACS+ authenticated users. The value **1** indicates that all users can telnet. TACACS+ privilege levels of 1-6 correspond to User level privilege, privilege levels 7-14 are mapped to Manager, and privilege level 15 are mapped to Security Officer. Therefore a value of **7-14** indicates that Manager privilege or better is required. A value of **15** is equivalent to Security Officer privilege.

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

For example, to allow telnet for TACACS+ authenticated Security Officers, use the command:

```
set tacplus telnet=15
```

To see the required privilege level, use the command:

```
show tacplus telnet
```

Figure 5: Example output from the **show tacplus telnet** command.

```
TACACS+ telnet privilege level: NONE
```

Table 7: Parameters in the output of the **show tacplus telnet** command.

Parameter	Meaning
TACACS+ telnet privilege level	The level of TACACS+ privilege required for using telnet on the router; a number in the range 0 to 15, or <b>none</b> . <b>None</b> indicates that no TACACS+ authenticated user can use telnet.

## TPAD Numeric Response

Your router supports a partial list of the Hayes Standard AT Command Set, or AT Command Set. An additional two commands are now supported that enable the router to return result code in either numeric or word format. The additional commands are shown in the following table.

Table 8: AT commands supported by Alliedware

Command	Description
ATV0 or ATV	<i>Result Code Format:</i> This command instructs the device to return results codes in numeric format.
ATV1	<i>Result Code Format:</i> This command instructs the device to return results codes in word (English) format. This is the default result code format.

The Alliedware TPAD result codes are shown in the following table.

Table 9: Alliedware TPAD result codes

Result code	Numeric code	Description
OK	0	Acknowledges the execution of a command line.
CONNECT	1	Confirms that ISDN number has been specified with ATDn.
NO CARRIER	2	Specifies that the device has dropped the ISDN call, or that the call has failed to connect.
ERROR	4	Invalid command.

For more information about the AT Command Set, see the *Transaction Packet Assembler Disassembler (TPAD)* chapter in the *Software Reference*.

# VLAN Tagging on ETH Interfaces

---

This enhancement enables Eth ports on the router to route IP packets between VLANs. It does this by applying a VLAN tag to frames that are transmitted out of the Eth port. You can configure multiple logical interfaces on the Eth port, so it can route frames to multiple VLANs.

An example of a possible scenario follows the description of the commands.

## Modified Commands

A new parameter, **vlan tag**, has been added to the **add ip interface** and **set ip interface** commands. The full syntax of these commands is now:

```
add ip interface=interface ipaddress={ipadd|dhcp}
[broadcast={0|1}] [directedbroadcast={false|no|off|on|
true|yes}] [filter={0..99|none}] [fragment={no|off|on|
yes}] [gratuitousarp={on|off}] [gre={0..100|none}]
[igmpproxy={off|upstream|downstream}] [mask=ipadd]
[metric=1..16] [multicast={both|no|off|on|receive|send|
yes}] [ospfmetric=1..65534] [policyfilter={100..199|
none}] [priorityfilter={200..299|none}] [proxyarp={false|
no|off|on|true|yes}] [ripmetric=1..16] [samode={block|
passthrough}] [vjc={false|no|off|on| true|yes}]
[vlan tag={1..4094|none}]

set ip interface=interface [broadcast={0|1}]
[directedbroadcast={false|no|off|on|true|yes}]
[filter={0..99|none}] [fragment={no|off|on|yes}]
[gratuitousarp={on|off}] [gre={0..100|none}]
[igmpproxy={off|upstream|downstream}] [ipaddress=ipadd|
dhcp] [mask=ipadd] [metric=1..16] [multicast={both|off|on|
receive|send}] [ospfmetric=1..65534]
[policyfilter={100..199|none}] [priorityfilter={200..299|
none}] [proxyarp={false|no|off|on|true|yes}]
[ripmetric=1..16] [samode={block|
passthrough}] [vjc={false|no|off|on| true|yes}]
[vlan tag={1..4094|none}]
```

The **vlan tag** parameter specifies the VID (VLAN Identifier) to be included in the header of each frame that is transmitted over the logical interface. This parameter is only valid for Eth interfaces. Multiple logical interfaces on the same physical interface can share the same VLAN tag. The default is **none**, which specifies that no VID is included.

The **show ip interface** command has been modified to show the VLAN tag associated with each interface.

Figure 6: Example output from the **show ip interface** command

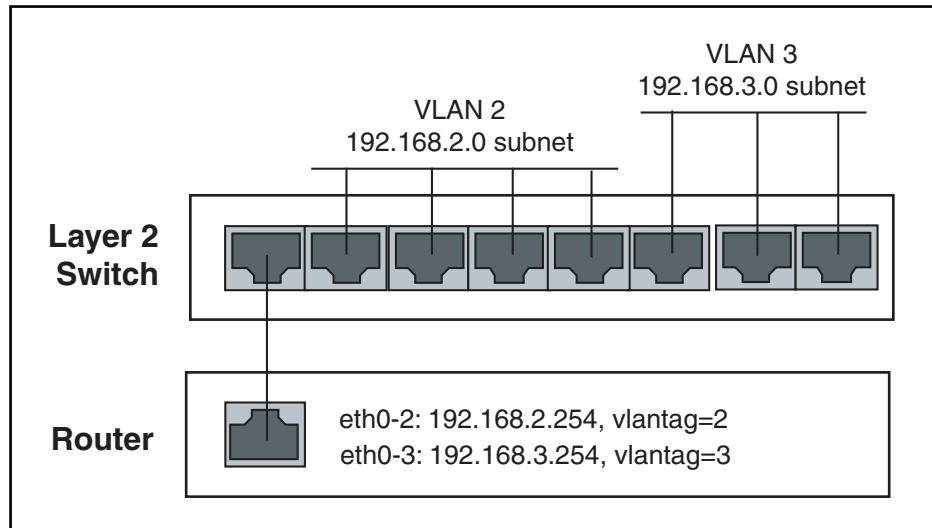
Interface	Type	IP Address	Bc	Fr	PArp	Filt	RIP Met.	SAMode	IPSc
Pri. Filt	Pol.Filt	Network Mask	MTU	VJC	GRE	OSPF Met.	DBcast	Mul.	
<b>VLAN Tag</b>									
Local	---	Not set	-	-	-	---	--	Pass	--
---	---	Not set	1500	-	---	---	---	---	---
---									
eth0-0	Static	192.168.2.1	1	n	On	---	01	Pass	No
---	---	255.255.255.0	1500	-	---	---	0000000001	No	Rec
<b>1</b>									
eth0-1	Static	192.101.2.1	1	n	On	---	01	Pass	No
---	---	255.255.255.0	1500	-	---	---	0000000001	No	Rec
<b>3</b>									
eth0-2	Static	192.168.23.3	1	n	On	---	01	Pass	No
---	---	255.255.255.0	1500	-	---	---	0000000001	No	Rec
<b>4</b>									

Table 10: The new parameter in the output of the **show ip interface** command

Parameter	Meaning
VLAN Tag	The VID (VLAN Identifier) that is included in the header of each frame that is transmitted over the Eth interface.

## Example

In this scenario, the router is acting as both a gateway and a routing device for a Layer 2 LAN. The eth0 port on the router is connected to a Layer 2 switch, through a port that is a tagged member of VLAN 2 and VLAN 3. Frames received on the eth0 port and destined for the Layer 2 switch are assigned the VID of the destination VLAN.



VLAN on L2 switch	IP subnet	logical ETH interface on router	Gateway address on router	VLAN tag on eth interface on router
vlan 2	192.168.2.0/24	eth0-2	192.168.2.254	2
vlan 3	192.168.3.0/24	eth0-3	192.168.3.254	3

To configure the router, use the following commands:

```
add ip interface=eth0-2 ipaddress=192.168.2.254
    mask=255.255.255.0 vlantag=2

add ip interface=eth0-3 ipaddress=192.168.3.254
    mask=255.255.255.0 vlantag=3
```

Note that only these logical Eth interfaces transmit tagged frames. Traffic transmitted out other interfaces (or logical interfaces) on the router are sent untagged.

# Adopting the VRRP IP Address

---

## Benefits of VRRP IP Address Adoption

The VRRP master router can *adopt* the IP address of the virtual router (VR), and respond to the following packets destined for the VR IP address, even if it does not own this IP address on any of its interfaces:

- ICMP echo requests (pings)
- Telnet and SSH connection requests
- HTTP and SSL GUI management requests
- SNMP requests, and
- DNS relay requests

VRRP IP Address Adoption allows continuous accessibility of the VR IP address even as the VR master changes. Using this feature:

- You can easily tell whether the VR is functioning, by pinging the single VR IP address.
- You can easily monitor the performance of the VR, regardless of which participating router is acting as master.
- DNS relay can continue functioning via the same IP address at all times.

## Risks of VRRP IP Address Adoption

When VRRP IP Address Adoption is used, the master router accepts packets destined for the virtual router, even though it may not own this IP address. This does not conform to RFC 2338. Because the same IP address refers to different devices at different times, there is a risk of confusion arising. This risk can be reduced by a suitable network management policy.

## Recommendations

Before using VR IP address adoption, consider the following guidelines to avoid confusion:

- Ensure that the VR has an IP address that is different from the interface IP addresses of any of the individual routers in the VR.
- Ensure that all routers in the virtual router use VRRP IP Address Adoption (or that none do).
- Use the VRRP IP address to monitor the VR master. Be aware that this does not give information about one particular participating router, but about the current VR master, whichever participating router is acting as the master at the time.
- When changing the configuration of the participating routers using Telnet, GUI or SNMP, configure each device individually by pointing to their individual IP addresses.
- When changing the configuration of the participating routers, do not use the VR IP address. Only one device, the VR master, is responding to this IP address, and you may not know which device it is.

## Configuration of VR IP Address Adoption

To configure VRRP IP Address Adoption, use the new parameter, **adoptvrip**, that has been added to the **create vrrp** and **set vrrp** commands:

```
create vrrp=vr-identifier over=physical-interface
    ipaddress=ipadd [adoptvrip={on|off}] [other-parameters...]
set vrrp=vr-identifier [adoptvrip={on|off}]
    [other-parameters]
```

The **adoptvrip** parameter specifies that when the router is acting as the VRRP master it should respond to requests directed at any IP address that it is backing up, even if it does not own that address. If it does not own the address the access requests that the router will permit are limited to: ICMP echo requests (pings), Telnet, SSH, HTTP and SSL GUI, SNMP and DNS relay. All other types of access to the address will be ignored. The default is OFF.



*If you set **adoptvrip** to on, give the VR an IP address that is different from the interface IP addresses of any of the individual routers in the VR, and only use the VR IP address to monitor the VR, not to configure any of its participating routers. Otherwise, you risk confusion when you monitor or configure individual routers. See [Synchronising Time Across Stacks on page -12](#) for more about risks and recommendations.*



*Configure all the routers in a virtual router with the same values for the VRRP virtual router identifier, IP address, adopt VR IP address mode, advertisement interval, preempt mode, authentication type and password. Inconsistent configuration will cause advertisement packets to be rejected and the virtual router will not perform properly.*

To display the value of the new parameter, use the **show vrrp** command.

Table 11: New parameter displayed in the output of the **show vrrp** command

Parameter	Meaning
Adopt VR IP Address(es)	Whether or not the router should respond to ICMP echo, Telnet, GUI, SNMP and DNS relay service requests targeted at the VR IP address(es) associated with the virtual router, even if it does not own those address(es).

## Firewall Enhancements

### ICMP Protocol for Firewall Policy Rule

A new option has been added to the **add firewall policy** and **set firewall policy** commands. **Icmp** (Internet Control Message Protocol) can now be specified as a **protocol** parameter option. To specific **icmp**, use the commands:

```
add firewall policy=policy rule=rule-id action={allow|deny|
    nat|nonat} interface=interface protocol={protocol|all|egp|
    gre|icmp|ospf|sa|tcp|udp} [other-parameters...]
set firewall policy=name rule=rule-id [protocol={protocol|
    all|egp|gre|icmp|ospf|sa|tcp|udp}] [other-parameters...]
```

For more information, see the *Firewall* chapter in the *Software Reference*.

## Debug and Display Firewall ARP Requests

A new option has been added to the **disable firewall policy** and **enable firewall policy** commands. **Arp** can now be specified as a **debug** parameter option. This option enables or disables the display of all ARP requests that have passed through the firewall.

To specific **arp**, use the commands:

```
enable firewall policy=name [debug={all|arp|http|packet|
  pkt|process|proxy|smtp|upnp}] [other-parameters...]
disable firewall policy=name [debug={all|arp|http|packet|
  pkt|process|proxy|smtp|upnp}] [other-parameters...]
```

The new command **show firewall arp** displays information about IP addresses specified in Firewall NAT configurations for which ARP responses from the router may be required. Use the command:

```
show firewall arp [policy=name]
```

The **policy** parameter specifies a firewall policy and displays IP addresses for NAT configurations with that policy. If this parameter is not specified, IP addresses are displayed for all policies.

The following is example output and parameter descriptions for the **show firewall arp**.

Figure 7: Example output from the **show firewall arp** command

IP (range)	ARP Interfaces Policy	NAT Type	Int	Gbl Int	Rule
172.20.8.50	Public Office	Int based	eth0-0	eth1-0	-
172.20.8.57 -172.20.8.62	All Public LAN	Rule	eth0-1	-	1

Table 12: Parameters in the output of the **show firewall arp** command

Parameter	Meaning
IP (range)	An IP address or range for which the router may be required to send ARP responses.
Policy	The name of the policy whose NAT configuration the IP address (range) belongs to.
ARP Interfaces	Interfaces in the policy on which ARP requests are permitted: Public - ARP requests are permitted on the public interface specified by the Gbl Int parameter All Public - ARP requests are permitted on all of the policy's public interfaces Private - ARP requests are permitted on the private interface specified by the Int parameter All Private - ARP requests are permitted on all of the policy's private interfaces An address in an ARP request must match the subnet of the interface on which the ARP request is received.

Table 12: Parameters in the output of the **show firewall arp** command (Continued)

Parameter	Meaning
NAT Type	The type of NAT configuration associated with the IP address: Int Based - The address (range) was specified by an interface-based NAT configured with the <b>add firewall policy nat</b> command Rule - The address (range) was specified by a NAT rule configured by the <b>add firewall policy rule</b> command, where the ACTION parameter was specified as NAT
Int	The private interface associated with the NAT configuration. If the NAT Type is Int based, this is the private interface specified by the INTERFACE parameter in the <b>add firewall policy nat</b> command. If the NAT Type is Rule, this is the interface to which the rule is attached. If this is a private interface, a dash indicates that the rule is attached to a public interface (see the Gbl Int parameter).
Gbl Int	The public interface associated with the NAT configuration. If the NAT Type is Int based, this is the public interface specified by the GBLINTERFACE parameter in the <b>add firewall policy nat</b> command. If the NAT Type is Rule, this is the interface to which the rule is attached. if this is a public interface, a dash indicates that the rule is attached to a private interface (see the Int parameter).
Rule	The number of the rule associated with this entry. When the NAT Type is Int based, no value is displayed.

For more information, see the *Firewall* chapter in the *Software Reference*.

## PPTP Pass Through

This enhancement enables multiple Point-To-Point Tunnelling Protocol (PPTP) tunnels to be initiated from the private side of the firewall by default. This will allow private users to terminate their PPTP tunnels on their respective corporate or private networks. From the public side of the firewall, PPTP tunnels are blocked by default.

PPTP is a tunnelled mechanism to transfer Point-to-Point Protocol (PPP) frames across an intermediate network. PPTP connection is a method to create secure connections across public networks, such as the Internet, for both remote access and router-to-router virtual private network (VPN) connections, utilising the authentication, encryption, and protocol configuration mechanisms of PPP.

PPTP uses a TCP connection for tunnel management and Generic Routing Encapsulation (GRE) is used to encapsulate PPP frames for tunnelled data. In this implementation of PPTP, GRE tunnels are created and closed automatically when PPTP is enabled and disabled. The payloads of the encapsulated PPP frames can be either encrypted or compressed, or both encrypted and compressed.

PPTP has been added to the list of pre-defined service names you can specify when adding or modifying a firewall policy rule.

Table 13: New pre-defined IP protocol service name

Service Name	Port Number	Standard Protocol
PPTP	1723	TCP

When PPTP clients are on the private side of the firewall, to add a rule **denying** PPTP tunnel traffic when adding or modifying a firewall policy rule, use the commands:

```
add firewall policy=policy-name rule=rule-id action=deny
    interface=interface protocol=tcp port=pptp
    [other-parameters]

set firewall policy=policy-name rule=rule-id action=deny
    interface=interface protocol=tcp port=pptp
    [other-parameters]
```

When PPTP clients are on the public side of the firewall, to add a rule **allowing** PPTP tunnel traffic when adding or modifying a firewall policy rule, use the commands:

```
add firewall policy=policy-name rule=rule-id action=allow
    interface=interface ip=ipadd[-ipadd] protocol=tcp
    port=pptp gblip=ipadd gblport=pptp [other-parameters]

set firewall policy=policy-name rule=rule-id action=allow
    interface=interface ip=ipadd[-ipadd] protocol=tcp
    port=pptp gblip=ipadd gblport=pptp [other-parameters]
```

The **port** parameter specifies a port number, a range of port numbers, or a predefined service name to match. If dynamic NAT is active on the interface, it is possible to re-map a global port number to a different internal port number. For rules applied to a private interface, PORT is the destination port on the public network. For rules applied to a public interface, PORT is either the destination port on the private network or, in the case of NAT being applied, the destination port on the private network where traffic will be mapped.

The **gblport** parameter specifies the port number, service name, or range of port numbers that apply to the rule if NAT is active on an interface.

PPTP has been added to the list of pre-defined service names you can specify when adding a local private IP network to the address translation table used by NAT.

Table 14: New service name for use with Network Address Translation (NAT)

Service Names	Value
PPTP	1723

To specify PPTP when adding a local private IP network to the address translation table used by NAT, use the command:

```
add ip nat ip=ipadd gblport=pptp port=pptp protocol=tcp
    [mask=ipadd] [gblip=ipadd] [gblmask=ipadd]
    [gblinterface=interface]
```

The **port** parameter specifies the port number or service name for the port used on the private IP host when specifying a static ENAT entry.

The **gblport** parameter specifies the port number or service name for the port available to global Internet access, when creating a static ENAT.

For more information, see the *Firewall* and the *Internet Protocol (IP)* chapters in the *Software Reference*.

## 802.1x Port Authentication

---

The IEEE 802.1x standard provides a method of restricting access to networks based on authentication information. 802.1x provides port based network access control for devices connected to an Ethernet LAN. This functionality allows a network controller to restrict external devices from gaining access to the network behind a 802.1x controlled port. External devices that wish to access services via a port under 802.1x control must firstly authenticate themselves and gain authorisation before any packets originating from, or destined for, the external device are allowed to pass through the 802.1x controlled port.

Authentication is controlled on a per-port basis.



---

*802.1x port authentication was initially released in Software Release 2.6.1 for the AR410, AR410S, and the AR700. This section describes support for 802.1x port authentication on AR450 routers only.*

---



---

*On the AR450, 802.1x port authentication is supported on all ports. On the AR410 and AR410S, 802.1x port authentication is supported on eth ports only.*

---

The main components of a 802.1x implementation are:

- the authenticator - the port that wishes to enforce authentication before allowing access to services that are accessible behind it.
- the supplicant - the port that wishes to access services offered by the authenticator's system.
- the authentication server - a device that uses the authentication credentials supplied by the supplicant, via the authenticator, to determine if the authenticator should grant access to its services. This implementation of 802.1x requires that a port acting as an authenticator must communicate with a RADIUS authentication server.

The 802.1x configurations supported on each switch and Ethernet port are:

- supplicant
- single-supplicant authenticator, where a single supplicant is directly connected to a single authenticator
- authenticator. The VLAN ports each support a single supplicant, and the Ethernet ports each support up to 10 supplicants connected to the authenticator via a hub.



---

*A multi-supplicant configuration does not conform to IEEE 802.1x and introduces security risks. To minimise the risk of unauthorised access and denial-of-service attacks, 802.1x control in a multi-supplicant configuration is not recommended.*

---

## Support for 802.1x

Port authentication is disabled by default. To enable it, use the command:

```
enable portauth
```

To enable the specified port(s) to act as an authenticator, use the command:

```
enable portauth port={all|port-name} type=authenticator  
[other-options...]
```

where *port-name* is an Ethernet interface (eth0 or eth1), a port number, a range of port numbers (specified as *n-m*), or a comma-separated list of port numbers and/or ranges. Port numbers start at *n* and end at *m*, where *n* is the lower value port number and *m* the upper value port number, including uplink ports.

On a router acting as an authenticator, if port authentication is enabled on all ports then the router will not perform port authentication. This is because the router will have no way of passing authentication requests from supplicants to the authentication server. Therefore, the authentication server must be connected to a port which does not have port authentication enabled, or is set with **control=authorised**.

To enable the specified port(s) to act as a supplicant, use the command:

```
enable portauth port={all|port-name} type=supplicant  
[other-options...]
```

To enable the specified port(s) to act as both an authenticator and a supplicant, use the command:

```
enable portauth port={all|port-name} type=both  
[other-options...]
```

To enable the specified port(s) to act as an authenticator, connected to either a single supplicant or multiple supplicants, use the command:

```
enable portauth port={all|port-name} type=authenticator  
[mode={multi|single}] [other-options...]
```

If **multi** is specified, the port distinguishes between multiple supplicants attached to it and requires each supplicant to authenticate themselves separately. If **single** is specified, the port is authenticated by the first supplicant attached to it. The default is **single**. **Multi** can only be specified for Ethernet ports.



---

*Multi-supplicant configuration is supported on Ethernet ports only.*

---

To change the configuration parameters for a port(s) under 802.1x port control, use the commands:

```
set portauth port={all|port-name} type=[authenticator|  
supplicant|both] [other-options...]
```

To reinitialise 802.1x port control on a specified port(s), use the command:

```
reset portauth port={all|port-name} [other-options...]
```

To display information about each port's capabilities and protocol implementation detail, use the command:

```
show portauth
```

To display counter information for ports on the router that have port authentication enabled, use the command:

```
show portauth counter port={all|port-name}
```

To display the current configuration for ports on the router that have port authentication enabled, use the command:

```
show portauth port={all|port-name}
```

To display the amount of time remaining in seconds until the timeout is due for each port authentication timer associated with the specified port or ports, use the command:

```
show portauth timer port={all|port-name}
```

To force any supplicant currently authorised on a port acting as an authenticator to immediately reauthenticate itself, use the command:

```
activate portauth port={all|port-name} reauthenticate
[other-options...]
```

To configure a global username and global password for all Supplicant Port Access Entities (PAEs) to use during authentication, use the command:

```
set portauth username=login-name password=password
[method={otp[encryption={md4|md5}]|standard}]
```

The default **method** is **standard**. The global settings may be overridden using the **set portauth port** command to set specific supplicant passwords and usernames.




---

*When you enter passwords, ensure you do so in a secure environment because passwords appear on the screen as typed. Also, all configuration file backups should be secured because passwords appear in plain text in these files.*

---

For background information, examples, more information about these commands, and information about other optional settings for port authentication, see the *Port Authentication* chapter in the *Software Reference*.

## IPsec NAT-Traversal

IPsec NAT-T is an enhancement to IPsec and ISAKMP protocols that lets Virtual Private Network (VPN) clients communicate through NAT gateways over the Internet. For example, business travellers commonly use IPsec on their laptops to gain remote VPN access to the central office. When working off-site, these users sometimes need to connect to the Internet through a NAT gateway such as from a hotel. Network Address Translation (NAT) gateways are often part of a company's firewall and let its Local Area Network (LAN) appear as one IP address to the world. For more information about NAT gateways, refer to RFC 1631 and to the *Network Address Translation* section in the *Internet Protocol* chapter of the *Reference Manual*.

Problems arise with NAT gateways for a number of reasons. A key one is that when they handle IPsec packets, they cannot access encrypted UDP or TCP headers. Therefore, NAT gateways cannot identify traffic for different private devices and cannot properly track individual sessions.

NAT-T is not on the NAT gateway and is not an "IPsec pass-through". NAT-T lets IPsec/ISAKMP peers send traffic through NAT gateways by putting packets inside UDP packets. This solution enables remote VPN users to communicate successfully when NAT gateways are part of the connection.

## Basic NAT-T Operations

Using NAT-D (discovery) messages, NAT-T negotiates with a peer to determine if NAT gateways are present and at which end of the network. Each peer sends at least two NAT-D messages as part of the ISAKMP phase 1 negotiation. The first message contains a hash of a destination IP address; subsequent messages contain source addresses. A NAT gateway is detected when address messages from the peer have incorrect hash values, which indicates that a NAT gateway changed IP addresses.

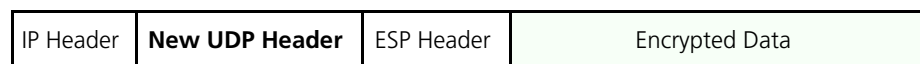
Also during phase 1, NAT-T determines whether a peer has NAT-T capabilities by detecting a vendor ID. Vendor IDs tell what version of NAT-T the peer supports. When a NAT gateway is not detected or a peer does not support NAT-T, normal IPsec negotiations and protection occur.

When an ISAKMP initiator detects a NAT gateway during an exchange, communication changes from UDP port 500 to port 4500. Log messages inform users that the UDP port has changed. Main or Aggressive mode packets received on the old port are discarded and a separate log is created.

Because IPsec traffic can also be received on port 4500, ISAKMP adds and removes the non-ESP marker at the start of the ISAKMP message so that messages can be detected and passed to the ISAKMP module. ISAKMP drops packets when it receives them on port 4500 without a non-ESP marker.

NAT-T inserts a UDP header between the outer IP and ESP headers thereby encapsulating the ESP data (Figure 8). NAT-T encapsulates IPsec traffic only when a NAT gateway is detected.

Figure 8: UDP Encapsulation for NAT-T



IPsec intercepts UDP-encapsulated ESP packets before they are passed to UDP.

A peer behind a NAT gateway sends keepalive messages to ensure that port mappings in the device remain active between peers. Keepalive intervals are not configurable. The purpose of keepalive messages is different from heartbeat messages controlled by the **heartbeatmode** parameter in ISAKMP Policy commands, which detect an IKE peer. IKE heartbeat messages and NAT-T keepalive messages do not affect each other.

## NAT-T on the Router

This NAT-T implementation supports interoperability with the following VPN clients:

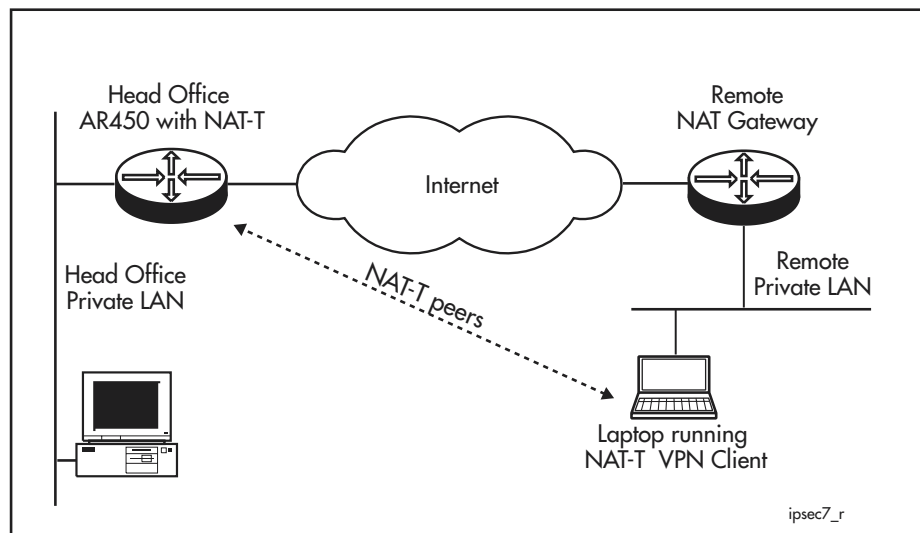
- SafeNet SoftRemote<sup>®</sup>
- Microsoft Windows 2000<sup>®</sup>
- Microsoft Windows XP<sup>®</sup>

NAT-T can also be implemented router-to-router for offices with their own IPsec router behind a NAT gateway. For router-to-VPN examples, see Configuration Notes for interoperability with SafeNet SoftRemote clients, and Microsoft Windows 2000 and XP VPN clients at [www.alliedtelesyn.co.nz/solutions/solutions.html](http://www.alliedtelesyn.co.nz/solutions/solutions.html).

Versions 02 and 08 of the following NAT-T IETF drafts have been implemented:

- *Negotiation of NAT-Traversal in the IKE*, draft-ietf-ipsec-nat-t-ike-02, which describes the modifications to IKE to support NAT detection and UDP tunnel negotiation
- *UDP Encapsulation of IPsec Packets*, draft-ietf-ipsec-udp-encaps-02, which defines the method of UDP encapsulation of IPsec packets
- *Negotiation of NAT-Traversal in the IKE*, draft-ietf-ipsec-nat-t-ike-08, which describes the modifications to IKE to support NAT detection and UDP tunnel negotiation
- *UDP Encapsulation of IPsec Packets*, draft-ietf-ipsec-udp-encaps-08, which defines the method of UDP encapsulation of IPsec packets

Figure 9: IPsec NAT-T peers negotiate traffic through a NAT gateway device



NAT-T is enabled by default, and is enabled or disabled in the ISAKMP policy. We recommend that users carefully read security considerations in the IETF drafts to fully understand the implications of using NAT-T. When users create an ISAKMP policy with the **create isakmp policy** command, they define how peers respond during an ISAKMP exchange, and can use the **nattraversal** parameter to disable NAT-T if they prefer. They can later use the same parameter with the **set isakmp policy** command to enable NAT-T again.

Users should configure an IPsec policy to allow ISAKMP traffic on UDP port 4500 so that it flows through the IPsec layer to ISAKMP. Refer to the ISAKMP policy commands in this chapter to change or add a policy.

Peers send their original, untranslated addresses to each other, which they store in the ISAKMP SA. Recipients use original addresses (OAs) to correct the checksums in the UDP or TCP headers in the IPsec payload.

NAT-T is implemented for IPv4 for both transport and tunnel modes. We recommend transport mode for MS clients. For SafeNet and router-to-router connections, we recommend tunnel mode and specifying unique IP addresses for remote peers. Refer to the **set ipsec saspecification** command to set modes.

## Commands Modified for NAT-T

- set isakmp policy
- show ipsec sa
- show ipsec counter=main
- show isakmp policy
- show isakmp sa
- show isakmp counter=general
- show isakmp counter=aggressive
- show isakmp counter=main
- show isakmp counter=network
- show isakmp counter=quick

### ISAKMP policy commands

The existing **create isakmp policy** and **set isakmp policy** commands have been modified to add a **nattraversal** parameter. This parameter enables or disables NAT-T to let peers negotiate a UDP-encapsulated mode so that IPsec traffic can flow through a NAT gateway. The default is enabled.

**Syntax** `CREate ISAkmp POLICY=name PEer={ipv4add|ipv6add|ANY}  
 [NATTraversal={ON|OFF|TRUE|FALSE}]  
 [other-isakmp-parameters]`

`SET ISAkmp POLIcy=name [NATTraversal={ON|OFF|TRUE|FALSE}]  
 [other-isakmp-parameters]`

### show ipsec sa

Output for the existing **show ipsec sa** command has been modified to include the following parameters that are specific to NAT-T.

Parameter	Meaning
<b>SA id</b>	The identification number for the SA.
Role	Whether this peer acted as the initiator or responder in order to create this SA.
Mode	The IPsec operational mode for this SA: TUNNEL TRANSPORT UDP_ENCAPSULATED_TUNNEL UDP_ENCAPSULATED_TRANSPORT
<b>NAT-Traversal NAT-OA</b>	Information about original IP addresses.
Peer original source IP address	Source IP address that the remote peer uses when sending packets to this peer. UDP-encapsulated transport mode only.
Peer original destination IP address	Destination IP address that the remote peer uses when sending packets to this peer. UDP-encapsulated transport mode, and IETF draft v08, <i>Negotiation of NAT-T in the IKE</i> .
<b>Filters</b>	Information about the packet selections for this SA.

Parameter	Meaning
NAPT remote port number	Network Address Port Translation number. Multiple clients in UDP-encapsulated transport mode appear to come from the same source. Therefore, NAT-T changes the source port to this value to maintain a distinction.

### show ipsec counter=main

Parameter	Meaning
<b>IPsec main packet processing counters</b>	
inProcessPktKeepalive	The number of NAT keepalive packets received.

### show isakmp policy

#### show isakmp sa

Parameter	Meaning
NAT Traversal	Whether NAT-T is enabled or disabled.

Parameter	Meaning
<b>NAT-Traversal Information</b>	Information about NAT-T capability.
NAT-T enabled	Whether NAT-T is enabled on the router.
Peer NAT-T capable	Whether the remote peer sent a valid NAT-T vendor ID.
NAT discovered	Whether a NAT gateway has been detected between peers: <ul style="list-style-type: none"> <li>No - Not detected</li> <li>Remote - Detected at the remote site</li> <li>Local - Detected at this site</li> <li>Both - Detected at local and remote sites</li> <li>Unknown - Peer is not NAT-T capable or the NAT discovery process was incomplete</li> </ul>

## show isakmp counter

- main
- network
- quick

Table 15: NAT-T parameters in the output of the **show isakmp counter=general** command

Parameter	Meaning
msgRxBadPortIpChange	The number of ISAKMP packets received with an unexpected source IP address or source port and discarded.
msgRxFailOldPort	The number of ISAKMP packets discarded because they were received on port 500 after NAT-T had moved the ISAKMP traffic to port 4500.

Table 16: NAT-T parameters in the output of the **show isakmp counter=aggressive** command

Parameter	Meaning
initSendNatD	The number of NAT-D messages sent by the initiator.
respSendNatD	The number of NAT-D messages sent by the responder.
initRecvNatD	The number of NAT-D messages received by the initiator.
respRecvNatD	The number of NAT-D messages received by the responder.

Table 17: NAT-T parameters in the output of the **show isakmp counter=main** command

Parameter	Meaning
initSendNatD	The number of NAT-D messages sent by the initiator.
initRecvNatD	The number of NAT-D messages received by the initiator.
respSendNatD	The number of NAT-D messages sent by the responder.
respRecvNatD	The number of NAT-D messages received by the responder.

Table 18: NAT-T parameter in the output of the **show isakmp counter=network** command

Parameter	Meaning
rxFailNoNonEspMarker	The number of packets ISAKMP dropped because they were received on NAT-T port 4500 without a non-ESP marker.

Table 19: NAT-T parameters in the output of the **show isakmp counter=quick** command

Parameter	Meaning
natOaSent	The number of NAT originating messages sent.
natOaReceived	The number of NAT originating messages received.
badNatOa	The number of non-conforming NAT-OA (originating address) messages received such as unknown type.

## NAT-T Configuration Example

This is a basic router-to-router solution with NAT-Traversal for a Virtual Private Network (VPN) that shows:

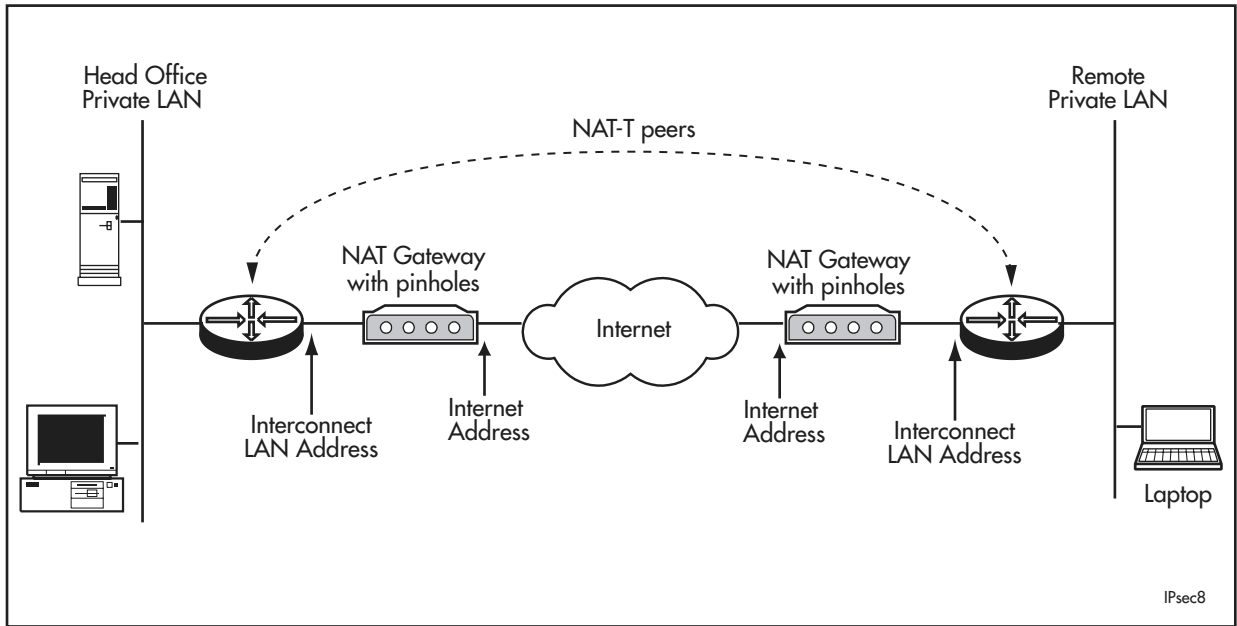
- NAT gateways at both ends of the VPN link
- a firewall configuration at both ends

For solutions that enable office-to-office VPN access through the Internet, as well as VPN access for travellers with VPN clients, see [www.alliedtelesyn.co.nz/solutions/solutions.html](http://www.alliedtelesyn.co.nz/solutions/solutions.html) for interoperability with Microsoft Windows 2000 and XP VPN clients, and with SafeNet SoftRemote clients.

### General Considerations

- This example also works with a NAT gateway at the initiator end, responder end, or neither end.
- IPsec and ISAKMP policies must refer to valid Internet peer addresses. When a peer router is directly connected to the Internet, this address is on its local WAN interface. When a peer accesses the Internet through a NAT gateway, this address is on the NAT gateway.
- If you have a NAT gateway at the responder end, it must be configured to allow traffic (*pinholes*) for UDP ports 500 and 4500.
- One end of the link must have a fixed Internet address. To enable both peers to initiate an IPsec link, both ends must have fixed addresses and both NAT gateways must have pinholes for UDP ports 500 and 4500.  
Many ISPs assign dynamic addresses that may change periodically, and you may need to ask for a fixed address.
- ISAKMP peers behind NAT gateways must identify themselves using a name string.

Figure 10: NAT-T router-to-router solution in IPsec tunnel mode



The router must be in secure mode and you must log on as a user with security officer privilege. The router must also already have pre-shared keys for ISAKMP to use. See the *IP Security* chapter in the *Software Reference* for details.

Figure 11: Head Office Router

```

set system name="IPsec Head Office"
set user securedelay=600
add user=secoff pass=secoff priv=sec login=yes telnet=yes
del user=manager

# The IPsec peer will be authenticated by the following name
# and password
add user=remote password=friend

# IP configuration
# Smaller MRU/MTU settings are needed for IPsec tunnel
# mode so that larger payload packets successfully pass
# through the IPsec tunnel
enable ip
set int=eth0 mtu=1300
add ip int=eth0 ip=interconnect-LAN-address frag=yes
add ip int=vlan1 ip=head-office-LAN-address
add ip rou=0.0.0.0 mask=0.0.0.0 int=eth0
    next=NAT-gateway-address

enable fire
create fire poli=main
add fire poli=main int=vlan1 type=private
add fire poli=main int=eth0 type=public
add fire poli=main nat=enhanced int=vlan1 gblint=eth0
add fire poli=main rule=1 int=eth0 action=allow
    ip=interconnect-LAN-address prot=udp port=500
    gblip=interconnect-LAN-address gblpo=500
add fire poli=main rule=2 int=eth0 action=allow
    ip=interconnect-LAN-address prot=udp port=4500
    gblip=interconnect-LAN-address gblpo=4500
add fire poli=main rule=3 int=eth0 action=nonat prot=all
    ip=head-office-LAN-ip-range x.x.x.x - x.x.x.x encap=ipsec

# Rule 4 for internally initiated VPN traffic to the
# remote site
add firewall poli=main ru=4 ac=nonat int=vlan1 prot=all
    ip=head-office-LAN-ip-range x.x.x.x - x.x.x.x
    poli=main ru=4 remoteip=remote-LAN-ip-range
    x.x.x.x - x.x.x.x

# IPsec configuration
create ipsec sas=1 key=isakmp prot=esp enc=des hash=sha
create ipsec bund=1 key=isakmp string="1"
create ipsec pol=isakmp int=eth0 ac=permit
set ipsec pol=isakmp lp=500
create ipsec pol=natt_udp int=eth0 ac=permit
set ipsec pol=natt_udp lp=4500

# The following peer's internet address may be on its
# NAT gateway or its router. If the remote site has a
# dynamically assigned address, use the keyword "dynamic".
create ipsec pol=remote_site int=eth0 ac=ipsec key=isakmp
    bund=1 peer=remote-internet-address isa=to_remote

```

Figure 11: Head Office Router (Continued)

```
set ipsec pol=remote_site
  lad=head-office-LAN-ip-subnet-address
  lmask=head-office-LAN-ip-subnet-mask
  rad=remote-LAN-ip-subnet-address
  rmask=remote-LAN-ip-subnet-mask

# If you need both VPN and internet-browsing access, use
# the following internet policy. Do not use this policy
# for VPN only.
create ipsec pol=internet int=eth0 ac=permit
enable ipsec

# ISAKMP configuration
# The following peer's internet address may be on its
# NAT gateway or its router. If the remote site has a
# dynamically assigned address, use the keyword "any".
create isakmp pol=to_remote
  peer=remote-internet-address key=1

# The following heartbeat parameter is optional and lets
# the office delete inactive SAs if the remote office
# connection to the Internet drops. Heartbeats can be set
# to send, receive or both. The receiver expects to receive
# heartbeats; when three are missing, it deletes the
# associated SA to avoid SA "out of step" fault conditions.
set isakmp pol=to_remote localid=head_office
  heartbeat=receive
set isakmp pol=to_remote sendd=true setc=true

# For XAUTH to authenticate the IPsec peer
set isakmp pol=to_remote xauth=server xauthtype=generic
enable isakmp
```

Figure 12: Remote Site Router

```

set system name="IPsec Remote Site"
set user securedelay=600
add user=secoff pass=secoff priv=sec login=yes telnet=yes
del user=manager

# IP configuration
# Smaller MRU/MTU settings are needed for IPsec tunnel
# mode so that larger payload packets successfully pass
# through the IPsec tunnel
enable ip
set int=eth0 mtu=1300
add ip int=eth0 ip=interconnect-LAN-address frag=yes
add ip int=vlan1 ip=remote-LAN-address
add ip rou=0.0.0.0 mask=0.0.0.0 int=eth0
    next=NAT-gateway-address

enable fire
create fire poli=main
add fire poli=main int=vlan1 type=private
add fire poli=main int=eth0 type=public
add fire poli=main nat=enhanced int=vlan1 gblint=eth0
add fire poli=main rule=1 int=eth0 action=allow
    ip=interconnect-LAN-address prot=udp port=500
    gblip=interconnect-LAN-address gblpo=500
add fire poli=main rule=2 int=eth0 action=allow
    ip=interconnect-LAN-address prot=udp port=4500
    gblip=interconnect-LAN-address gblpo=4500
add fire poli=main rule=3 int=eth0 action=nonat prot=all
    ip=remote-LAN-ip-range x.x.x.x - x.x.x.x encap=ipsec

# Rule 4 for internally initiated VPN traffic to the remote
# site
add firewall poli=main ru=4 ac=nonat int=vlan1 prot=all
    ip=remote-LAN-ip-range x.x.x.x - x.x.x.x
    poli=main ru=4 remoteip=head-office-LAN-ip-range
    x.x.x.x - x.x.x.x

# IPsec configuration
create ipsec sas=1 key=isakmp prot=esp enc=des hash=sha
create ipsec bund=1 key=isakmp string="1"
create ipsec pol=isakmp int=eth0 ac=permit
set ipsec pol=isakmp lp=500 rp=500
create ipsec pol=natt_udp int=eth0 ac=permit
set ipsec pol=natt_udp lp=4500 rp=4500

# The following peer's internet address may be on its
# NAT gateway or its router. This example assumes the head
# office has a fixed address.
create ipsec pol=head_office int=eth0 ac=ipsec
    key=isakmp bund=1 peer=head-office-internet-address
    isa=to_office
set ipsec pol=head_office
    lad=remote-LAN-ip-subnet-address
    lmask=remote-LAN-ip-subnet-mask
    rad=head-office-LAN-ip-subnet-address
    rmask=head-office-LAN-ip-subnet-mask

```

Figure 12: Remote Site Router (Continued)

```
# If you need both VPN and internet-browsing access, use
# the following internet policy. Do not use this policy
# for VPN only.
create ipsec pol=internet int=eth0 ac=permit
enable ipsec

# ISAKMP configuration
# The following peer's internet address may be on its
# NAT gateway or its router. This example assumes the head
# office has a fixed address.
create isakmp pol=to_office
  peer=head-office-internet-address key=1

# The following heartbeat parameter is optional and lets
# the office delete inactive SAs if the remote office
# connection to the Internet drops. Heartbeats can be set
# to send, receive or both. The receiver expects to receive
# heartbeats; when three are missing, it deletes the
# associated SA to avoid SA "out of step" fault conditions.
set isakmp pol=to_office localid=remote_site heartbeat=send
set isakmp pol=to_office sendd=true setc=true

# For XAUTH to authenticate the IPsec peer
set isakmp pol=to_office xauth=client xauthname=remote
  xauthpass=friend
enable isakmp
```

