



# Cisco ASA 5500 Migration to Version 8.3 and Later

---

**Released: March 8, 2010**

**Updated: June 16, 2011 for 8.4(2)**

This guide describes the configuration migration process when you upgrade from an earlier version of the Cisco ASA 5500 operating system (OS) to Version 8.3 and later.

This guide includes the following sections:

- [Upgrading the Software, page 1](#)
- [Information About Migration, page 1](#)
- [Real IP Addresses in Access List Migration, page 4](#)
- [NAT Migration, page 15](#)
- [Network and Service Object Migration, page 38](#)
- [Downgrading from Version 8.3, page 41](#)

## Upgrading the Software

To upgrade the software using the CLI, see the “Managing Software and Configurations” chapter in *Cisco ASA 5500 Series Configuration Guide using the CLI*:

[http://www.cisco.com/en/US/docs/security/asa/asa83/configuration/guide/admin\\_swconfig.html](http://www.cisco.com/en/US/docs/security/asa/asa83/configuration/guide/admin_swconfig.html)

To upgrade the software using ASDM, see the “Managing Software and Configurations” chapter in *Cisco ASA 5500 Series Configuration Guide using ASDM*:

[http://www.cisco.com/en/US/docs/security/asa/asa83/asdm63/configuration\\_guide/admin\\_swconfig.html](http://www.cisco.com/en/US/docs/security/asa/asa83/asdm63/configuration_guide/admin_swconfig.html)

## Information About Migration

This section describes the migrated features, automatic backup of the original configuration file, and saving your new migrated configuration. This section includes the following topics:



---

**Americas Headquarters:**  
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

© 2011 Cisco Systems, Inc. All rights reserved.

- [Migrated Features, page 2](#)
- [Automatic Backup of the Old Configuration, NAT Migration File, Bootup Error Log, page 2](#)
- [Saving the Migrated Configuration, page 3](#)

## Migrated Features

The major changes in Version 8.3 that require migration are as follows:

- Real IP addresses in access lists, where access lists are used in supported features—When using NAT or PAT, you used to have to specify the *mapped* addresses and ports in an access list for all features that use access lists. Now, for several supported features, you must use the real, untranslated IP address and ports. (Other features continue to use the mapped IP address).
- NAT—The NAT feature has been redesigned for increased flexibility and functionality. All NAT and NAT-related commands have been redesigned.
- Named Network and Service Objects—Network and service objects are automatically created for NAT.



**Note** Although you can use named network and service objects in other features, such as access lists and object groups, objects are not automatically created for any feature other than NAT.

When upgrading from 8.3 or 8.4(1) to 8.4(2), migration for static identity NAT will occur to preserve existing functionality. See the “[Sample NAT Migration from 8.3 and 8.4 to 8.4\(2\)](#)” section on page 18 for more information.

## Automatic Backup of the Old Configuration, NAT Migration File, Bootup Error Log

The old startup configuration is automatically saved in flash memory. The NAT migration file and the bootup error log, which includes any migration messages, is automatically saved to flash memory as well.

This section includes the following topics:

- [Backup Configuration Files, page 2](#)
- [NAT Migration File, page 3](#)
- [Bootup Error Log File, page 3](#)

### Backup Configuration Files

The following old startup configuration files are saved in flash memory:

- Single mode configuration file or multiple mode system configuration—`disk0:major_minor_maint_interim_startup_cfg.sav` where *major\_minor\_maint\_interim* is the old OS version number.

For example, `8_2_1_0_startup_cfg.sav`.

- Multiple mode context configuration (if present in flash memory)—**disk0:major\_minor\_maint\_interim\_context\_cfg.sav** where *major\_minor\_maint\_interim* is the old OS version number and *context* is the context name.

For example, 8\_2\_1\_0\_context1\_cfg.sav.

If there is insufficient memory to save configuration files, an error message appears on the console of the adaptive security appliance and is saved in the bootup error log file; any files saved as part of the migration will be removed, and the migration will be aborted.

## NAT Migration File

When your NAT configuration is migrated, and the following file is added to the root directory: `nat_ident_migrate`. The presence of this empty file indicates that the configuration was migrated, and prevents re-migration at bootup.

## Bootup Error Log File

To view the bootup error log, enter the **show startup-config errors** command. See the following sample log:

```
hostname# show startup-config errors
Reading from flash...
!
REAL IP MIGRATION: WARNING
In this version access-lists used in 'access-group', 'class-map',
'dynamic-filter classify-list', 'aaa match' will be migrated from
using IP address/ports as seen on interface, to their real values.
If an access-list used by these features is shared with per-user ACL
then the original access-list has to be recreated.
INFO: Note that identical IP addresses or overlapping IP ranges on
different interfaces are not detectable by automated Real IP migration.
If your deployment contains such scenarios, please verify your migrated
configuration is appropriate for those overlapping addresses/ranges.
Please also refer to the ASA 8.3 migration guide for a complete
explanation of the automated migration process.

INFO: MIGRATION - Saving the startup configuration to file

INFO: MIGRATION - Startup configuration saved to file 'flash:8_2_1_15_startup_cfg.sav'
*** Output from config line 4, "ASA Version 8.2(1)15 "
NAT migration logs:
INFO: NAT migration completed.
Real IP migration logs:
    ACL <1> has been migrated to real-ip version
```

## Saving the Migrated Configuration

The migrated configuration is in running memory only; be sure to save the configuration to the startup configuration. If you do not save it, the next time you reload, the original configuration goes through the migration process again.

- CLI—Enter the **write memory** command.
- ASDM—Click **Save** at the top of the window.

# Real IP Addresses in Access List Migration

When using NAT or PAT, mapped addresses and ports are no longer required in an access list for several features. You should now always use the real, untranslated addresses and ports for these features. Using the real address and port means that if the NAT configuration changes, you do not need to change the access lists. This section includes the following topics:

- [Features That Use Real IP Addresses, page 4](#)
- [Features That Continue to Use Mapped IP Addresses, page 5](#)
- [Real IP Address Migration Naming Conventions, page 5](#)
- [Syslog Message Migration, page 5](#)
- [Sample Real IP Address Migration, page 6](#)
- [Real IP Address Migration Messages and Limitations, page 10](#)

## Features That Use Real IP Addresses

The following commands and features now use real IP addresses in the access lists. All of the **access-list** commands used for these features are automatically migrated unless otherwise noted. For access lists that use network object groups (the **object-group network** command), the IP addresses within the object group are migrated to the real IP addresses.

- **access-group** command
- Modular Policy Framework **match access-list** command
- Botnet Traffic Filter **dynamic-filter enable classify-list** command
- AAA **aaa ... match** commands
- WCCP **wccp redirect-list group-list** command



**Note** The WCCP **wccp redirect-list group-list** command is not automatically migrated. The WCCP access list is downloaded after startup, so automatic migration cannot occur. You need to manually change the **wccp redirect-list group-list** command to use an access list with the real IP address.

For example, formerly if you wanted to allow an outside host to access an inside host that used NAT, you applied an inbound access list on the outside interface using the **access-group** command. In this scenario, you needed to specify the mapped address of the inside host in the access list because that address was the address that can be used on the outside network. Starting in 8.3, you need to specify the real address in the access list.

### ASDM

Real IP addresses are now used in the following features instead of mapped addresses:

- Access Rules
- AAA Rules
- Service Policy Rules
- Botnet Traffic Filter classification
- WCCP redirection

**Note**

WCCP redirection is not automatically migrated. The WCCP ACL is downloaded after startup, so automatic migration cannot occur. You need to manually change the ACL to use the real IP address.

## Features That Continue to Use Mapped IP Addresses

The following features use access lists, but these access lists will continue to use the mapped values as seen on an interface:

- IPsec access lists
- **capture** command access lists
- Per-user access lists
- Routing protocol access lists
- All other feature access lists...

## Real IP Address Migration Naming Conventions

- In most cases after migration, the new **access-list** commands will be recreated with the original name so there will be no changes to the configuration that references the access list name. If an access list is applied to two or more features, and the conversion results in different ACEs, then two different access lists will be created; the original access list is removed. The new access lists will have the original name with appended suffixes: *oldname\_migration\_X*, where *X* is a number starting with 1.
- When contents of an object group need to be changed to the real IP addresses, a new **object-group** command called *oldname\_X* is created, where *X* is a number starting with 1. The new **object-group** command is referenced in the access list.

## Syslog Message Migration

For the following syslog messages, the destination IP address has been changed from *mapped-ip* to *real-ip* format so that the addresses in the syslog will match what is configured:

- Syslog ID 106001 is changed for the **access-group** command.
- Syslog ID 106100 is changed for the **access-group** command.
- Syslog ID 106023 is changed for the **access-group** command.
- Syslog ID 201010, 201011, 201012, 201013 are changed for the **set connection** command.

## Sample Real IP Address Migration

Table 1 shows how access lists are migrated to use the real IP address. The NAT configuration is shown in the old configuration for reference. For NAT migration, see the “NAT Migration” section on page 15.

**Table 1** Real IP Address Migration Examples

Description	Configuration Migration
Static NAT with ingress access group	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.23.57.1 10.50.50.50 netmask 255.255.255.255  access-list 1 permit ip any host 172.23.57.1 access-group 1 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 permit ip any host 10.50.50.50 access-group 1 in interface outside</pre>
Static NAT with egress access group	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.23.57.170 10.50.50.50  object-group network hm   network-object host 172.23.57.170 access-list 2 extended deny tcp object-group hm any eq www access-group 2 out interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>object-group network hm_1   network-object host 10.50.50.50 access-list 2 extended deny tcp object-group hm_1 any eq www access-group 2 out interface outside</pre>
Static host NAT with access list matching mapped subnet	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.23.57.170 10.50.50.50  access-list 1 extended permit ip any 172.23.57.0 255.255.255.0 access-group 1 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 extended permit ip any host 10.50.50.50 access-list 1 extended permit ip any 172.23.57.0 255.255.255.0 access-group 1 in interface outside</pre>

**Table 1** Real IP Address Migration Examples (continued)

Description	Configuration Migration
Static PAT; only one ACE in the access rule matches the PAT	<p><b>Old Configuration</b></p> <pre>static (inside,outside) tcp 172.23.57.170 5080 10.50.50.50 80  access-list 1 extended permit tcp any host 172.23.57.170 eq 5080 access-list 1 extended permit udp any host 172.23.57.170 eq 5080 access-list 1 extended permit tcp any host 172.23.57.170 eq 10000 access-list 1 extended permit tcp any host 10.2.3.4 eq 5080 access-group 1 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 extended permit tcp any host 10.50.50.50 eq 80 access-list 1 extended permit udp any host 172.23.57.170 eq 5080 access-list 1 extended permit tcp any host 172.23.57.170 eq 10000 access-list 1 extended permit tcp any host 10.2.3.4 eq 5080 access-group 1 in interface outside</pre>
Dynamic NAT with AAA.	<p><b>Old Configuration</b></p> <pre>global (outside) 1 172.23.57.171-172.23.57.172 nat (inside) 1 10.50.50.0 255.255.255.0 nat (dmz) 1 192.168.4.0 255.255.255.0  object-group network mapped_pool   network-object host 172.23.57.171   network-object host 172.23.57.172  access-list 1 permit udp any object-group mapped_pool  aaa authentication match 1 outside TEST_SERVER</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 permit udp any 10.50.50.0 255.255.255.0 access-list 1 permit udp any 192.168.4.0 255.255.255.0</pre>
Interface-specific service policy	<p><b>Old Configuration</b></p> <pre>static (inside,outside) tcp 172.23.57.170 6021 10.50.50.50 21  access-list 1 permit tcp any host 172.23.57.170 eq 6021  class-map ftpclass   match access-list 1 policy-map ftp_pol   class ftpclass     inspect ftp service-policy ftp_pol interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 permit tcp any host 10.50.50.50 eq ftp  class-map ftpclass   match access-list 1 policy-map ftp_pol   class ftpclass     inspect ftp service-policy ftp_pol interface outside</pre>

Table 1 Real IP Address Migration Examples (continued)

Description	Configuration Migration
Global service policy; NAT for only a subset of interfaces	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.23.57.170 10.50.50.50  access-list 1 permit ip any host 172.23.57.170  class-map c1   match access-list 1 policy-map global_policy   class c1     ips inline fail-close service-policy global_policy global</pre> <p><b>Migrated Configuration</b></p> <pre><b>access-list 1 permit ip any host 10.50.50.50</b> access-list 1 permit ip any host 172.23.57.170  class-map c1   match access-list 1 policy-map global_policy   class c1     ips inline fail-close service-policy global_policy global</pre>
Shared access list between access group and service policy; Static host NAT with access list matching mapped subnet	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.23.57.170 10.50.50.50  access-list 1 extended permit ip any 172.23.57.0 255.255.255.0 access-group 1 in interface outside  class-map c1   match access-list 1 policy-map p1   class c1     inspect http service-policy global_policy global</pre> <p><b>Migrated Configuration</b></p> <pre><b>access-list 1 extended permit ip any host 10.50.50.50</b> access-list 1 extended permit ip any 172.23.57.0 255.255.255.0 access-group 1 in interface outside  class-map c1   match access-list 1 policy-map p1   class c1     inspect http service-policy global_policy global</pre>



**Table 1** Real IP Address Migration Examples (continued)

Description	Configuration Migration
Single access list converted to multiple access lists after migration	<p><b>Old Configuration</b></p> <pre>static (outside,inside) 172.23.1.10 10.132.44.12 netmask 255.255.255.255 static (outside,dmz) 172.23.1.10 10.132.44.135 netmask 255.255.255.255  access-list 1 extended permit ip any host 172.23.1.10 access-group 1 in interface inside access-group 1 in interface dmz</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1_1 extended permit ip any host 10.132.44.12 access-group 1_1 in interface inside access-list 1_2 extended permit ip any host 10.132.44.135 access-group 1_2 in interface dmz</pre>
Policy NAT migration	<p><b>Old Configuration</b></p> <pre>access-list policyacl1 extended permit ip host 10.50.50.50 10.0.0.0 255.0.0.0  global (outside) 1 172.23.57.170 nat (inside) 1 access-list policyacl1  access-list 1 permit ip any host 172.23.57.170 access-group 1 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 extended permit ip any host 10.50.50.50 access-group 1 in interface outside</pre>

**Table 1** Real IP Address Migration Examples (continued)

Description	Configuration Migration
Object Group expansion	<p><b>Old Configuration</b></p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  object-group network TEST   network-object object obj-10.1.2.0   network-object host 192.168.101.10  static (inside,outside) 10.1.2.1 172.16.2.1 static (mgmt,outside) 192.168.101.10 172.16.2.10  access-list 1 extended permit ip any object-group TEST access-group 1 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 1 remark Migration, ACE (line 1) expanded: permit ip any object-group TEST access-list 1 extended permit ip any host 172.16.2.1 access-list 1 extended permit ip any 10.1.2.0 255.255.255.0 access-list 1 extended permit ip any host 172.16.2.10 access-list 1 remark Migration: End of expansion</pre>
Access group with deny/permit ACEs	<p><b>Old Configuration</b></p> <pre>global (outside) 1 10.10.10.128-10.10.10.255 nat (inside) 1 172.16.10.0 255.255.255.0  access-list 100 extended deny ip any host 10.10.10.210 access-list 100 extended permit ip any 10.10.10.211 255.255.255.128 access-group 100 in interface outside</pre> <p><b>Migrated Configuration</b></p> <pre>access-list 100 extended deny ip any 172.16.10.0 255.255.255.0 access-group 100 in interface outside</pre>

## Real IP Address Migration Messages and Limitations

This section describes messages associated with real IP address migration. Some messages relate to configurations that cannot be migrated, and require user intervention. This section also lists any other conditions that do not result in a message. This section includes the following topics:

- [Real IP Address Migration Messages, page 10](#)
- [For Interface IP Address in ACE, Real vs. Mapped Status Cannot Be Determined, page 14](#)

## Real IP Address Migration Messages

When you first reload with 8.3, you see the following message:

```
REAL IP MIGRATION: WARNING
  In this version access-lists used in 'access-group', 'class-map',
  'dynamic-filter classify-list', 'aaa match' will be migrated from
  using IP address/ports as seen on interface, to their real values.
  If an access-list used by these features is shared with per-user ACL
  then the original access-list has to be recreated.
```

Please refer to documentation for more details.

Table 2 lists other messages you might see.

**Table 2** Real IP Address Migration Messages

---

**Message and Description**

---

**Error Message** Couldn't migrate ACL <name> into real values, please manually migrate. Associated access-group config is removed.

**Explanation** If an access list is used by an **access-group** command, and the access list is not migrated for some reason, then the **access-group** command is deleted so a security hole is not created.

---

**Error Message** ACE converted to real IP/port values based on dynamic NAT or PAT. The new ACE(s) could be broader in scope than this original ACE.

**Explanation** When you have dynamic NAT and the access list includes a subset of the addresses in the global pool, then the access list is not migrated because the NAT command is more broad than the access list. Migrating the access list using the real IP address in the **nat** command would create a much broader access rule than the original. Note that the **access-group** command is deleted so a security hole is not created.

```
global (outside) 1 10.10.10.128-10.10.10.255
nat (inside) 1 192.168.10.0 255.255.255.0
```

```
access-list 100 extended permit ip any host 10.10.10.210 <---If this were migrated, it would be
192.168.10.0, which is too broad.
access-group 100 in interface outside <---This is deleted
```

---

**Error Message** ACE converted to real IP/port values based on dynamic/static Policy NAT. The new ACE(s) need to be checked for enforcing policy NAT ACL.

**Explanation** When you migrate policy NAT, check that the new access list does not open any security holes. For example, the following pre-migration configuration translates 10.50.50.50 to 172.23.57.170 only when the destination address is on 10.0.0.0:

```
access-list policyacl1 extended permit ip host 10.50.50.50 10.0.0.0 255.0.0.0
static (inside,outside) 172.23.57.170 access-list policyacl1
```

This access rule permits any traffic to the mapped address, but because this mapping only occurs when the traffic is to or from 10.0.0.0, this access list essentially only allows 10.0.0.0 to access the inside host:

```
access-list 1 permit ip any host 172.23.57.170
access-group 1 in interface outside
```

The migrated configuration permits any traffic to the inside host; however, because the access list now uses the real IP address, any traffic can access the inside host, and not just traffic from 10.0.0.0:

```
access-list 1 extended permit ip any host 10.50.50.50
access-group 1 in interface outside
```

**Recommended Action** You should fix the access list to be:

```
access-list 1 extended permit ip 10.0.0.0 255.0.0.0 host 10.50.50.50
access-group 1 in interface outside
```

---

Table 2 Real IP Address Migration Messages (continued)

---

**Message and Description**


---

**Error Message** ACL <inbound\_auth> has been successfully migrated to real-ip version

**Explanation** An access list was migrated, and the same name was used.

---

**Error Message** After migration source network is 'any', originally it wasn't 'any'.

**Error Message** After migration destination network is 'any', originally it wasn't 'any'.

**Explanation** The access list was not migrated. Because the NAT configuration includes **nat (inside) 1 0 0**, the access-list would be migrated to **any any**. Because an **any any** access list opens a security hole, this migration is skipped. For example, all addresses are translated to a global pool:

```
global (outside) 1 172.23.57.0-172.23.57.255
nat (inside) 1 0 0
```

Then all addresses are permitted to access the global pool addresses:

```
object-group network mapped_pool
  network-object network 172.23.57.0 255.255.255.0

access-list 1 permit udp any object-group mapped_pool
access-group 1 in interface outside
```

Because migration would create this access rule, the rule is not migrated to the following:

```
access-list 1 permit udp any any
access-group 1 in interface outside
```

---

**Error Message** Can't convert rule to hole.

**Explanation** Internal error condition.

---

**Error Message** Can't create new ACE with obj-grp.

**Explanation** Internal error condition.

---

**Error Message** Can't create new hole.

**Explanation** Internal error condition.

---

**Error Message** Conversion for interface <if\_name> failed for line.

**Explanation** Internal error condition.

---

**Error Message** Destination changed for egress ACL, can't migrate this ACL.

**Explanation** Internal error condition.

---

**Table 2** *Real IP Address Migration Messages (continued)***Message and Description**

**Error Message** During migration of access-list <name> expanded this object-group ACE.

**Explanation** Access lists needed to be created for each address in an object group. See the “Object Group expansion” migration example.

**Error Message** Failed to create acl element to track during migration.

**Explanation** Internal error condition.

**Error Message** INFO: Note that identical IP addresses or overlapping IP ranges on different interfaces are not detectable by automated Real IP migration. If your deployment contains such scenarios, please verify your migrated configuration is appropriate for those overlapping addresses/ranges. Please also refer to the ASA 8.3 migration guide for a complete explanation of the automated migration process.

**Explanation** In some cases, you can change the access rules to accommodate the overlapping addresses (see the following example). If you cannot change the access rules, you might need to use a new IP addressing scheme for the overlapping networks.

For example, the following pre-migration configuration includes two static rules where the IP address 192.168.1.1 on two inside interfaces (group1 and group2) is mapped separately when it goes to the outside interface:

```
static (group1,outside) 10.10.1.1 192.168.1.1
static (group2,outside) 10.10.2.1 192.168.1.1
```

The following ACEs, when used in an **access-group** command applied to the outbound direction of the outside interface, permit the group1 mapped address (10.10.1.1) to exit the outside interface, but deny the group2 mapped address (10.10.2.1):

```
access-list out_acl extended permit ip host 10.10.1.1 any
access-list out_acl extended deny ip host 10.10.2.1 any
access-group out_acl out interface outside
```

However, when the ACEs are converted to real IP addresses, both the 10.10.1.1 and 10.10.2.1 mapped addresses are changed to the 192.168.1.1 real address; because the first ACE permits traffic to 192.168.1.1, the deny ACE will never be hit, and traffic will go to both the group1 and group2 hosts:

```
object foo
  host 192.168.1.1
  nat (group1,outside) static 10.10.1.1
object bar
  host 192.168.1.1
  nat (group2,outside) static 10.10.2.1
access-list out_acl extended permit ip object foo any
access-list out_acl extended deny ip object bar any <----This ACE will never be hit
access-group out_acl out interface outside
```

**Recommended Action** In this case, you can alter the access rule as follows:

```
access-list out_acl1 extended permit ip object foo any
access-list out_acl2 extended deny ip object bar any
access-group out_acl1 in interface group1
access-group out_acl2 in interface group2
```

Table 2 Real IP Address Migration Messages (continued)

**Message and Description**

**Error Message** No ACL was changed as part of Real-ip migration

**Explanation** No access lists needed to be changed.

**Error Message** Removing ACL <name>, it has been migrated to one or more ACLs with name format <name\_x>, example <name\_7>

**Explanation** An access list was migrated and resulted in two or more access lists with new names. The old access list was removed.

**Error Message** Something changed in conversion but not clear what changed.

**Explanation** Internal error condition.

**Error Message** Source changed for ingress ACL, can't migrate this ACL.

**Explanation** Internal error condition.

## For Interface IP Address in ACE, Real vs. Mapped Status Cannot Be Determined

If you have an ACE with an IP address that belongs to an interface, but the corresponding NAT command uses the **interface** keyword to identify the interface IP address, then the migration script cannot match the NAT command with the ACE, and it cannot know if the IP address in the ACE is real or mapped.

In this case, the migration script will not migrate the IP address; you will have to manually change the IP address to the real IP address. Alternatively, you can change the ACE to use the **interface** keyword.

For example, pre-migration, outside interface PAT is defined for an inside host:

```
static (inside,outside) tcp interface 80 10.2.2.2 80
```

You define an access list using the interface IP address, instead of the **interface** keyword:

```
access-list outside_access_in permit tcp any host 192.168.1.1 eq 80
access-group outside_access_in in interface outside
```

When you migrate to 8.3, the access list will not be migrated to the real IP address (10.2.2.2) because the **static** command could not be matched to the **access-list** command. If you had used the **interface** keyword, then the access list would have migrated correctly to use the real IP address instead of the **interface** keyword.

To fix the access list after migration, change the access list to use the real IP address (10.2.2.2):

```
access-list outside_access_in permit tcp any host 10.2.2.2 eq 80
```

# NAT Migration

The NAT feature has been redesigned for increased flexibility and functionality. All NAT and NAT-related commands have been redesigned. This section describes how your NAT configuration is migrated to the new NAT commands. For ASDM users, see the relevant “ASDM” subsections. This section includes the following topics:

- [Old NAT Commands, page 15](#)
- [New NAT Commands, page 16](#)
- [Supporting Commands for NAT, page 17](#)
- [Preserving the Order of NAT Rules, page 17](#)
- [NAT Migration Guidelines and Limitations, page 18](#)
- [Sample NAT Migration from 8.3 and 8.4 to 8.4\(2\), page 18](#)
- [Sample NAT Migration from 8.2 and Earlier, page 19](#)
- [NAT Migration Messages, page 35](#)

**Note**

---

Almost all NAT configurations will migrate seamlessly. In the rare cases when user intervention is required, you will be notified. There will never be an unreported loss of security after migration. See the [“NAT Migration Messages” section on page 35](#).

---

## Old NAT Commands

The following commands are no longer supported; they are migrated to new commands, and are then removed from the configuration.

- **alias**
- **global**
- **nat** (old version)
- **nat-control**
- **static**
- **sysopt nodnsalias**—This command is not migrated; instead, configure the **dns** option within the new NAT commands.

**ASDM**

The **alias** command was never supported in ASDM.

## New NAT Commands

Table 3 lists the new NAT commands. See also the “Supporting Commands for NAT” section on page 17.

**Table 3**      *New NAT Commands*

New Commands	Configuration Mode	Syntax
<b>Network Object NAT</b> (Typically used for regular NAT configurations.)		
<b>nat dynamic</b>	Object network	<pre>object network name   nat [(real_ifc,mapped_ifc)] dynamic     {[mapped_inline_host_ip] [interface]       [mapped_obj] [pat-pool mapped_obj [round-robin]] [interface]} [dns]</pre>
<b>nat static</b>	Object network	<pre>object network name   nat [(real_ifc,mapped_ifc)] static     {mapped_inline_ip   mapped_obj   interface}     [dns   service {tcp   udp} real_port mapped_port]     [no-proxy-arp] [route-lookup]</pre>
<b>Twice NAT</b> (Typically used for policy NAT configurations.)		
<b>nat source dynamic</b>	Global	<pre>nat [(real_ifc,mapped_ifc)] [line   {after-object [line]}]   source dynamic {real_obj   any}     {[mapped_obj] [pat-pool mapped_obj [round-robin]] [interface]}     [destination static {mapped_obj   interface} {real_obj   any}]     [service {mapped_dest_svc_obj real_dest_svc_obj} [dns] [unidirectional]     [inactive] [description desc]</pre>
<b>nat source static</b>	Global	<pre>nat [(real_ifc,mapped_ifc)] [line   {after-object [line]}]   source static {real_obj   any} {mapped_obj   interface   any}     [destination static {mapped_obj   interface} {real_obj   any}]     [service {real_src_mapped_dest_svc_obj   any}     mapped_src_real_dest_svc_obj] [dns] [unidirectional   [no-proxy-arp]     [route-lookup]] [inactive] [description desc]</pre>



### Note

The **no-proxy-arp**, **route-lookup**, **pat-pool**, and **round-robin** keywords were added in 8.4(2).

### ASDM

For ASDM, the existing NAT rules will be migrated to two new types of rules:

- Network Object NAT:  
Configuration > Firewall > Objects > Network Objects/Groups > Add/Edit Network Object.
- Twice NAT:  
Configuration > Firewall > NAT Rules



## Supporting Commands for NAT

To achieve migration to the new NAT commands, additional commands are created as shown in [Table 4](#):

**Table 4** Supporting Commands for NAT

Generated Commands	Description
<b>object network</b>	<p>For each network object NAT command, an <b>object network</b> command is created to represent the real IP address that you want to translate; the new <b>nat</b> command is a subcommand under the <b>object network</b> command. Similarly, <b>object network</b> commands are created for the mapped addresses inside the new <b>nat</b> commands when an inline address (one that is entered directly in the command) is not feasible.</p> <p>For twice NAT, which can use only <b>object network</b> commands to identify IP addresses, and not inline addresses or <b>access-list</b> commands, IP addresses from your old configuration are converted into <b>object network</b> commands.</p> <p>The <b>name</b> commands that are used in the NAT configuration are automatically migrated to the new <b>object network</b> commands; the <b>name</b> commands remain in the configuration for use with other features that do not yet support <b>object network</b> commands.</p>
<b>object service</b>	For twice NAT, <b>object service</b> commands are created for any inline services or services identified in an <b>access-list</b> command that was formerly used in policy NAT.
<b>object-group network</b>	In network object NAT, for multiple mapped addresses, an <b>object-group network</b> command is created that contains multiple <b>object network</b> commands.

See the “[Network and Service Object Migration](#)” section on page 38 for more information about network and service objects, including naming conventions for these generated commands.

### ASDM

ASDM has supported named network objects for a number of releases; now, the platform has the commands to properly support them as well. In addition to showing all named network objects in the configuration, ASDM automatically creates objects for any IP addresses used in the configuration; these auto-created objects are identified by the IP address, and are not present as objects in the platform configuration. If you assign a name to one of these objects, then ASDM adds the named network object to the platform configuration.



#### Note

ASDM no longer shows any objects derived from the **name** command. Previously, you might have used named objects derived from the **name** command in ASDM. If the **name** command IP address was not migrated (see the “[Network and Service Object Migration](#)” section on page 38), then these objects are replaced by auto-created objects identified by an IP address.

## Preserving the Order of NAT Rules

In the old NAT configuration, the order that NAT commands were assessed depended on the type of NAT, and in some cases, the order in which the commands appeared in the configuration. The new NAT order uses a table with three sections:

- Section 1 (twice NAT rules)—These rules are assessed based on the order they appear in the configuration. For migration purposes, this section includes migrated policy NAT rules.

- Section 2 (network object NAT (generated) rules)—These rules are assessed according to internal rules; the order they appear in the configuration does not matter (For more information, see the *Cisco ASA 5500 Series Configuration Guide using ASDM* or the *Cisco ASA 5500 Series Configuration Guide using the CLI*). For migration purposes, this section includes regular NAT rules.
- Section 3 (twice NAT rules that you specifically want to be evaluated after the network object NAT rules)—Like section 1, these rules are assessed in the order they appear in the configuration. However, they are assessed after section 1 and section 2 rules. This section is not used for NAT migration.

In the case of overlapping networks (for example, if a regular static NAT rule overlaps with a dynamic policy NAT rule), the regular static NAT rule will be migrated to section 1 instead of section 2 to preserve the order of the configuration. For example, the following old configuration has overlapping networks. In this case, the static command will be migrated to a twice NAT rule in section 1.

```
static (inside,outside) 209.165.202.129 10.1.1.6 netmask 255.255.255.255
access-list NET1 permit ip 10.1.1.0 255.255.255.0 209.165.202.0 255.255.255.0
nat (inside) 100 access-list NET1
```

## NAT Migration Guidelines and Limitations

- Dynamic identity NAT (the **nat 0** command) will not be migrated. See the “[NAT Migration Messages](#)” section on page 35. Static identity NAT is treated like any other **static** command, and is converted depending on whether it is regular or policy NAT.
- NAT exemption (the **nat 0 access-list** command) is migrated differently depending on the release to which you are upgrading. See the “[NAT Exemption](#)” section on page 24 for more information.
- When upgrading to 8.4(2) from 8.3(1), 8.3(2), or 8.4(1), migration for identity NAT will occur to preserve existing functionality. See the “[Sample NAT Migration from 8.3 and 8.4 to 8.4\(2\)](#)” section on page 18 for more information.
- Regular NAT commands with the **dns** option will be migrated. The **dns** option in static PAT and policy NAT commands will be ignored.
- Connection Settings in old NAT commands—Options such as **conn-max**, **emb-limit**, **norandomseq**, or **nailed** will be moved to service policies.

The following naming conventions are used for the new service policies:

- **class-map**—class-conn-param-*protocol-n*
- **access-list**—acl-conn-param-*protocol-n*
- **policy-map**—policy-conn-param-*interface*

For other naming conventions related to NAT migration, see the “[Object Migration Naming Conventions](#)” section on page 39.

## Sample NAT Migration from 8.3 and 8.4 to 8.4(2)

If you are already running 8.3(1), 8.3(2), or 8.4(2), then to preserve existing functionality, all identity NAT statements are migrated to use the following new keywords:

- **no-proxy-arp**
- **route-lookup** (routed firewall mode only)

Starting in version 8.4(2), identity NAT now performs proxy ARP and uses the NAT configuration to determine the egress interface by default. To maintain the functionality that was in 8.3(1), 8.3(2), and 8.4(2), proxy ARP is disabled, and a route lookup is performed to determine the egress interface using the new keywords. If you want to enable proxy ARP (a rare requirement) or use the NAT configuration to determine the egress interface, you must manually remove the keyword(s) after migration.

If the **unidirectional** keyword is present (for example, from an original migration of NAT exemption rules to 8.3(2) or 8.4(1)), then the keyword is removed.

Table 5 lists static identity NAT migration examples.

**Table 5** Identity NAT Migration Examples

Description	Configuration Migration
Static object NAT	<p><b>Old Configuration</b></p> <pre>object network obj-10.1.1.6   host 10.1.1.6   nat (inside,outside) static 10.1.1.6</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.1.6   host 10.1.1.6   nat (inside,outside) static no-proxy-arp route-lookup</pre>
Static twice NAT with unidirectional	<p><b>Old Configuration</b></p> <pre>nat (inside,any) source static any any destination static obj-192.168.90.0-01 obj-192.168.90.0-01 unidirectional</pre> <p><b>Migrated Configuration</b></p> <pre>nat (inside,any) source static any any destination static obj-192.168.90.0-01 obj-192.168.90.0-01 no-proxy-arp route-lookup</pre>
Static twice NAT	<p><b>Old Configuration</b></p> <pre>nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0  nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0</pre> <p><b>Migrated Configuration</b></p> <pre>nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup  nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup</pre>

## Sample NAT Migration from 8.2 and Earlier

This section includes the following topics:

- [Static NAT/PAT, page 20](#)
- [Dynamic NAT/PAT, page 20](#)
- [NAT Exemption, page 24](#)
- [NAT Control, page 30](#)
- [DNS Rewrite, page 30](#)

- [Connection Settings, page 31](#)
- [Source and Destination NAT, page 32](#)
- [alias Command, page 34](#)

## Static NAT/PAT

[Table 6](#) lists static NAT/PAT migration examples.

**Table 6** *Static NAT/PAT Migration Examples*

Description	Configuration Migration	Type / Section
Regular Static NAT	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 209.165.201.15 10.1.1.6 netmask 255.255.255.255</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.1.6   host 10.1.1.6   nat (inside,outside) static 209.165.201.15</pre>	Object / Section 2
Regular Static PAT	<p><b>Old Configuration</b></p> <pre>static (inside,outside) tcp 10.1.2.45 80 10.1.1.16 8080 netmask 255.255.255.255</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.1.16   host 10.1.1.16   nat (inside,outside) static 10.1.2.45 service tcp 8080 www</pre>	Object / Section 2
Static Policy NAT	<p><b>Old Configuration</b></p> <pre>access-list NET1 permit ip host 10.1.2.27 10.76.5.0 255.255.255.224</pre> <pre>static (inside,outside) 209.165.202.129 access-list NET1</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.2.27   host 10.1.2.27 object network obj-209.165.202.129   host 209.165.202.129 object network obj-10.76.5.0   subnet 10.76.5.0 255.255.255.224</pre> <pre>nat (inside,outside) source static obj-10.1.2.27 obj-209.165.202.129 destination static obj-10.76.5.0 obj-10.76.5.0</pre>	Twice / Section 1

## Dynamic NAT/PAT

[Table 7](#) lists dynamic NAT/PAT migration examples.

**Table 7**      **Dynamic NAT/PAT Migration Examples**

Description	Configuration Migration	Type / Section
Regular Dynamic PAT	<p><b>Old Configuration</b></p> <pre>nat (inside) 1 192.168.1.0 255.255.255.0 nat (dmz) 1 10.1.1.0 255.255.255.0 global (outside) 1 209.165.201.3</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-192.168.1.0   subnet 192.168.1.0 255.255.255.0   nat (inside,outside) dynamic 209.165.201.3 object network obj-10.1.1.0   subnet 10.1.1.0 255.255.255.0   nat (dmz,outside) dynamic 209.165.201.3</pre>	Object / Section 2
Regular Dynamic PAT (2)	<p><b>Old Configuration</b></p> <pre>nat (inside) 1 10.1.2.0 255.255.255.0 global (outside) 1 209.165.201.3 global (dmz) 1 172.16.4.5</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0   nat (inside,outside) dynamic 209.165.201.3 object network obj-10.1.2.0-01   subnet 10.1.2.0 255.255.255.0   nat (inside,dmz) dynamic 172.16.4.5</pre>	Object / Section 2
Regular Dynamic PAT (3)	<p><b>Old Configuration</b></p> <pre>nat (inside) 1 0 0 global (outside) 1 interface</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj_any   subnet 0.0.0.0 0.0.0.0   nat (inside,outside) dynamic interface</pre>	Object / Section 2

**Table 7**      **Dynamic NAT/PAT Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Dynamic Policy NAT	<p><b>Old Configuration</b></p> <pre> object-group network og-net-src   network-object 192.168.1.0 255.255.255.0   network-object 192.168.2.0 255.255.255.0 object-group network og-net-dst   network-object 209.165.201.0 255.255.255.224 object-group service og-ser-src   service-object tcp gt 2000   service-object tcp eq 1500  access-list NET6 extended permit object-group og-ser-src object-group og-net-src object-group og-net-dst  nat (inside) 10 access-list NET6 global (outside) 10 209.165.200.225                     </pre> <p><b>Migrated Configuration</b></p> <pre> object-group network og-net-src   network-object 192.168.1.0 255.255.255.0   network-object 192.168.2.0 255.255.255.0 object-group network og-net-dst   network-object 209.165.201.0 255.255.255.224 object network obj-209.165.200.225   host 209.165.200.225 object service obj_tcp_range_2001_65535   service tcp destination range 2001 65535 object service obj_tcp_eq_1500   service tcp destination eq 1500  nat (inside,outside) source dynamic og-net-src obj-209.165.200.225 destination static og-net-dst og-net-dst service obj_tcp_range_2001_65535 obj_tcp_range_2001_65535  nat (inside,outside) source dynamic og-net-src obj-209.165.200.225 destination static og-net-dst og-net-dst service obj_tcp_eq_1500 obj_tcp_eq_1500                     </pre>	Twice / Section 1

**Table 7**      **Dynamic NAT/PAT Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Policy Dynamic NAT (with multiple ACEs)	<p><b>Old Configuration</b></p> <pre>access-list ACL_NAT permit ip 172.29.0.0 255.255.0.0 172.29.37.0 255.255.255.0 access-list ACL_NAT permit ip 172.29.0.0 255.255.0.0 10.231.110.0 255.255.255.0 access-list ACL_NAT permit ip 172.29.0.0 255.255.0.0 10.107.204.0 255.255.255.0 access-list ACL_NAT permit ip 172.29.0.0 255.255.0.0 192.168.5.0 255.255.255.0  nat (inside) 1 access-list ACL_NAT global (outside) 1 209.165.200.225</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-172.29.0.0   subnet 172.29.0.0 255.255.0.0 object network obj-209.165.200.225   host 209.165.200.225 object network obj-172.29.37.0   subnet 172.29.37.0 255.255.255.0 object network obj-10.231.110.0   subnet 10.231.110.0 255.255.255.0 object network obj-10.107.204.0   subnet 10.107.204.0 255.255.255.0 object network obj-192.168.5.0   subnet 192.168.5.0 255.255.255.0  nat (inside,outside) source dynamic obj-172.29.0.0 obj-209.165.200.225 destination static obj-172.29.37.0 obj-172.29.37.0  nat (inside,outside) source dynamic obj-172.29.0.0 obj-209.165.200.225 destination static obj-10.231.110.0 obj-10.231.110.0  nat (inside,outside) source dynamic obj-172.29.0.0 obj-209.165.200.225 destination static obj-10.107.204.0 obj-10.107.204.0  nat (inside,outside) source dynamic obj-172.29.0.0 obj-209.165.200.225 destination static obj-192.168.5.0 obj-192.168.5.0</pre>	Twice / Section 1

Table 7 Dynamic NAT/PAT Migration Examples (continued)

Description	Configuration Migration	Type / Section
Outside NAT	<p><b>Old Configuration</b></p> <pre>global (inside) 1 10.1.2.30-10.1.2.40 nat (dmz) 1 10.1.1.0 255.255.255.0 outside  static (inside,dmz) 10.1.1.5 10.1.2.27 netmask 255.255.255.255</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-10.1.2.30-10.1.2.40   range 10.1.2.30 10.1.2.40 object network obj-10.1.2.27   host 10.1.2.27   nat (inside,dmz) static 10.1.1.5 object network obj-10.1.1.0   subnet 10.1.1.0 255.255.255.0   nat (dmz,inside) dynamic obj-10.1.2.30-10.1.2.40</pre>	Object / Section 2
NAT & Interface PAT together	<p><b>Old Configuration</b></p> <pre>nat (inside) 1 10.1.2.0 255.255.255.0 global (outside) 1 interface global (outside) 1 209.165.201.1-209.165.201.2</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-209.165.201.1_209.165.201.2   range 209.165.201.1 209.165.201.2 object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0   nat (inside,outside) dynamic obj_209.165.201.1_209.165.201.2 interface</pre>	Object / Section 2

## NAT Exemption

NAT exemption (the **nat 0 access-list** command) is a form of policy NAT, and is converted to static twice NAT. Rules are created between the exempted interface and all lower-security level interfaces. For outside NAT, rules are created between the exempted interface and all higher-security level interfaces. If you enabled same security level communication, rules are also created between the exempted interface and same-security level interfaces.

These rules will be placed at the top of section 1.

NAT exemption (the **nat 0 access-list** command) is migrated to a twice NAT rule. See the following notes for the version you are upgrading to for specific information about how NAT exemption is migrated:

- For Version 8.3(1)—In some cases, you might see caveat #CSCtf89372. We recommend migrating directly to 8.4(2). For more information about this caveat, see the Bug Toolkit at the following URL: <http://tools.cisco.com/Support/BugToolKit/action.do?hdnAction=searchBugs>
- For Version 8.3(2) through 8.4(1)—The **unidirectional** keyword was added. The **unidirectional** keyword only allows traffic on the source network to initiate connections. This migration change was made to fix CSCtf89372. Because NAT exemption is normally bidirectional, you might need to remove the **unidirectional** keyword to restore the original function. Specifically, this change adversely affects many VPN configurations that include NAT exemption rules (see CSCti36048 for this new issue). To avoid manual intervention, we recommend migrating to 8.4(2) instead.



If you are impacted by this issue, you will see a syslog message like the following:

```
%ASA-5-305013: Asymmetric NAT rules matched for forward and reverse flows; Connection
for icmp src Outside:192.168.1.5 dst inside:10.10.5.20 (type 8, code 0) denied due to
NAT reverse path failure
```

- For Version 8.4(2) and later—The **unidirectional** keyword is no longer added. Instead, the new **no-proxy-arp** and **route-lookup** keywords are added. Both the CSCtf89372 and CSCti36048 caveats are resolved in this release.

The examples in this section are for a system with three interfaces: inside (level 100), outside (level 0), and dmz (level 50).

[Table 8](#) lists NAT exemption migration examples.

**Table 8** NAT Exemption Migration Examples

Description	Configuration Migration	Type / Section
Regular NAT exemption (overlapping dynamic NAT shown)	<p><b>Old Configuration</b></p> <pre>access-list outside_nat_outbound extended permit ip 192.168.90.0 255.255.254.0 host 10.1.4.5  nat (outside) 2 access-list outside_nat_outbound outside global (inside) 2 interface  access-list inside_nat0_outbound_1 extended permit ip any 192.168.90.0 255.255.254.0  nat (inside) 0 access-list inside_nat0_outbound_1</pre> <p><b>Migrated Configuration</b></p> <p>8.3(1):</p> <pre>nat (outside,inside) source dynamic obj-192.168.90.0-01 interface destination static obj-10.1.4.5 obj-10.1.4.5  nat (inside,any) source static any any destination static obj-192.168.90.0-01 obj-192.168.90.0-01</pre> <p>8.3(2) through 8.4(1):</p> <pre>nat (outside,inside) source dynamic obj-192.168.90.0-01 interface destination static obj-10.1.4.5 obj-10.1.4.5  nat (inside,any) source static any any destination static obj-192.168.90.0-01 obj-192.168.90.0-01 unidirectional</pre> <p>8.4(2) and later:</p> <pre>nat (outside,inside) source dynamic obj-192.168.90.0-01 interface destination static obj-10.1.4.5 obj-10.1.4.5  nat (inside,any) source static any any destination static obj-192.168.90.0-01 obj-192.168.90.0-01 no-proxy-arp route-lookup</pre>	Twice / Section 1 (placed at the top)

**Table 8 NAT Exemption Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Regular NAT exemption	<p><b>Old Configuration</b></p> <pre>access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any  nat (inside) 0 access-list EXEMPT nat (dmz) 0 access-list EXEMPT</pre> <p><b>Migrated Configuration</b></p> <p>8.3(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0</pre> <p>8.3(2) through 8.4(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional</pre> <p>8.4(2) and later:</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup</pre>	Twice / Section 1 (placed at the top)

**Table 8 NAT Exemption Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Same security level enabled	<p><b>Old Configuration</b></p> <pre>same-security-level permit intra-interface  access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any  nat (dmz) 0 access-list EXEMPT</pre> <p><b>Migrated Configuration</b></p> <p>8.3(1):</p> <pre>same-security-level permit intra-interface  object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0  nat (dmz,dmz) source static obj-10.1.2.0 obj-10.1.2.0</pre> <p>8.3(2) through 8.4(1):</p> <pre>same-security-level permit intra-interface  object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional  nat (dmz,dmz) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional</pre> <p>8.4(2) and later:</p> <pre>same-security-level permit intra-interface  object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0  nat (dmz,outside) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup  nat (dmz,dmz) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup</pre>	Twice / Section 1 (placed at the top)

**Table 8 NAT Exemption Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Outside NAT	<p><b>Old Configuration</b></p> <pre>access-list EXEMPT permit ip 10.1.2.0 255.255.255.0 any nat (dmz) 0 access-list EXEMPT outside nat (outside) 0 access-list EXEMPT outside</pre> <p><b>Migrated Configuration</b></p> <p>8.3(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 nat (dmz,inside) source static obj-10.1.2.0 obj-10.1.2.0 nat (outside,dmz) source static obj-10.1.2.0 obj-10.1.2.0 nat (outside,inside) source static obj-10.1.2.0 obj-10.1.2.0</pre> <p>8.3(2) through 8.4(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 nat (dmz,inside) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional nat (outside,dmz) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional nat (outside,inside) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional</pre> <p>8.4(2) and later:</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 nat (dmz,inside) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup nat (outside,dmz) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup nat (outside,inside) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup</pre>	Twice / Section 1 (placed at the top)

**Table 8 NAT Exemption Migration Examples (continued)**

Description	Configuration Migration	Type / Section
Multiple ACEs	<p><b>Old Configuration</b></p> <pre>access-list EXEMPT extended permit ip 10.1.2.0 255.255.255.0 any access-list EXEMPT extended permit ip 10.1.3.0 255.255.255.0 20.2.4.0 255.255.255.0 access-list EXEMPT extended permit ip any 20.2.20.0 255.255.255.0 nat (inside) 0 access-list EXEMPT</pre> <p><b>Migrated Configuration</b></p> <p>8.3(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 object network obj-10.1.3.0   subnet 10.1.3.0 255.255.255.0 object network obj-20.2.4.0   subnet 20.2.4.0 255.255.255.0 object network obj-20.2.20.0   subnet 20.2.20.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 nat (inside,any) source static obj-10.1.3.0 obj-10.1.3.0 destination static obj-20.2.4.0 obj-20.2.4.0 nat (inside,any) source static any any destination static obj-20.2.20.0 obj-20.2.20.0</pre> <p>8.3(2) through 8.4(1):</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 object network obj-10.1.3.0   subnet 10.1.3.0 255.255.255.0 object network obj-20.2.4.0   subnet 20.2.4.0 255.255.255.0 object network obj-20.2.20.0   subnet 20.2.20.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 unidirectional nat (inside,any) source static obj-10.1.3.0 obj-10.1.3.0 destination static obj-20.2.4.0 obj-20.2.4.0 unidirectional nat (inside,any) source static any any destination static obj-20.2.20.0 obj-20.2.20.0 unidirectional</pre> <p>8.4(2) and later:</p> <pre>object network obj-10.1.2.0   subnet 10.1.2.0 255.255.255.0 object network obj-10.1.3.0   subnet 10.1.3.0 255.255.255.0 object network obj-20.2.4.0   subnet 20.2.4.0 255.255.255.0 object network obj-20.2.20.0   subnet 20.2.20.0 255.255.255.0  nat (inside,any) source static obj-10.1.2.0 obj-10.1.2.0 no-proxy-arp route-lookup nat (inside,any) source static obj-10.1.3.0 obj-10.1.3.0 destination static obj-20.2.4.0 obj-20.2.4.0 no-proxy-arp route-lookup nat (inside,any) source static any any destination static obj-20.2.20.0 obj-20.2.20.0 no-proxy-arp route-lookup</pre>	Twice / Section 1 (placed at the top)

## NAT Control

The **nat-control** command is deprecated. To maintain the requirement that all traffic from a higher security interface to a lower security interface be translated, a NAT rule will be inserted at the end of section 2 for each interface to disallow any remaining traffic. The **nat-control** command was used for NAT configurations defined with earlier versions of the adaptive security appliance. The best practice is to use access rules for access control instead of relying on the absence of a NAT rule to prevent traffic through the adaptive security appliance.

Table 9 lists NAT control migration examples.

**Table 9 NAT Control Migration Examples**

Description	Configuration Migration	Type / Section
Four interfaces: inside, outside, dmz, and mgmt	<p><b>Old Configuration</b></p> <pre>nat-control</pre> <p><b>Migrated Configuration</b> <pre>object network obj_any   subnet 0.0.0.0 0.0.0.0   nat (inside,outside) dynamic obj-0.0.0.0 object network obj-0.0.0.0   host 0.0.0.0 object network obj_any-01   subnet 0.0.0.0 0.0.0.0   nat (inside,mgmt) dynamic obj-0.0.0.0 object network obj_any-02   subnet 0.0.0.0 0.0.0.0   nat (inside,dmz) dynamic obj-0.0.0.0 object network obj_any-03   subnet 0.0.0.0 0.0.0.0   nat (mgmt,outside) dynamic obj-0.0.0.0 object network obj_any-04   subnet 0.0.0.0 0.0.0.0   nat (dmz,outside) dynamic obj-0.0.0.0 object network obj_any-05   subnet 0.0.0.0 0.0.0.0   nat (dmz,mgmt) dynamic obj-0.0.0.0</pre> </p>	Object / Section 2 (placed at the bottom)

## DNS Rewrite

Regular NAT commands with the **dns** option will be migrated. The **dns** option in static PAT and policy NAT commands will be ignored.

Table 10 lists DNS rewrite migration examples.

**Table 10** *DNS Rewrite Migration Examples*

Description	Configuration Migration	Type / Section
Static command with dns option	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.20.1.10 192.168.100.10 netmask 255.255.255.255 dns</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-192.168.100.10   host 192.168.100.10   nat (inside,outside) static 172.20.1.10 dns</pre>	Object / Section 2

## Connection Settings

Connection Settings in old NAT commands—Options such as **conn-max**, **emb-limit**, **norandomseq**, or **nailed** will be moved to service policies.

For naming conventions, see the “[NAT Migration Guidelines and Limitations](#)” section on page 18.

[Table 11](#) lists connection setting migration examples.

**Table 11** *Connection Settings Migration Examples*

Description	Configuration Migration	Type / Section
TCP and UDP Max Connections, random sequence number disabling, and nailed option.	<p><b>Old Configuration</b></p> <pre>static (inside,outside) 172.20.1.10 192.168.100.10 netmask 255.255.255.255 tcp 10 20 norandomseq nailed</pre> <p><b>Migrated Configuration</b></p> <pre>access-list acl-conn-param-tcp-01 extended permit tcp host 192.168.100.10 any t class-map class-conn-param-tcp-01   match access-list acl-conn-param-tcp-01  policy-map policy-conn-param-inside   class class-conn-param-tcp-01     set connection per-client-max 10 per-client-embryonic-max 20     random-sequence-number disable      set connection advanced-options tcp-state-bypass  service-policy policy-conn-param-inside interface inside  object network obj-192.168.100.10   host 192.168.100.10   nat (inside,outside) static 172.20.1.10</pre>	Object / Section 2

Table 11 Connection Settings Migration Examples (continued)

Description	Configuration Migration	Type / Section
UDP Max Connections	<p><b>Old Configuration</b></p> <pre>access-list NAT_ACL permit ip host 10.76.6.111 any  nat (dmz) 101 access-list NAT_ACL udp 6 global (outside) 101 225.22.22.1</pre> <p><b>Migrated Configuration</b></p> <pre>access-list NAT_ACL extended permit ip host 10.76.6.111 any  class-map class-conn-param-udp-01   match access-list NAT_ACL  policy-map policy-conn-param-dmz   class class-conn-param-udp-01     set connection per-client-max 6  service-policy policy-conn-param-dmz interface dmz  object network obj-10.76.6.111   host 10.76.6.111   nat (dmz,outside) dynamic 225.22.22.1</pre>	Object / Section 2

## Source and Destination NAT

Before 8.3, policy NAT let you specify the source and destination addresses, but NAT was only performed on the source address. In 8.3 and later, you can also configure NAT for the destination address if desired. In the old configuration to achieve this functionality, you had to configure two separate NAT rules for source and destination NAT for a single connection. As part of migration the two, independent NAT rules are tied together to form a single twice NAT command.

[Table 12](#) lists source and destination NAT migration examples.



**Table 12**      **Source and Destination NAT Migration Examples**

Description	Configuration Migration	Type / Section
Static commands for source and destination NAT	<p><b>Old Configuration</b></p> <pre>access-list NET1 permit ip host 192.168.1.1 host 192.168.1.10 access-list NET2 permit ip host 209.165.200.225 host 209.165.200.228 static (inside,outside) 209.165.200.228 access-list NET1 static (outside,inside) 192.168.1.10 access-list NET2</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-192.168.1.1   host 192.168.1.1 object network obj-209.165.200.228   host 209.165.200.228 object network obj-209.165.200.225   host 209.165.200.225 object network obj-192.168.1.10   host 192.168.1.10  nat (inside,outside) source static obj-192.168.1.1 obj-209.165.200.228 <b>destination static obj-192.168.1.10 obj-209.165.200.225</b></pre> <p>(The following rules are created by the migration script, but they may not be necessary; in rare circumstances, traffic might use one of these rules.)</p> <pre>nat (inside,outside) source static obj-192.168.1.1 obj-209.165.200.228 destination static obj-192.168.1.10 obj-192.168.1.10  nat (outside,inside) source static obj-209.165.200.225 obj-192.168.1.10 destination static obj-209.165.200.228 obj-209.165.200.228</pre>	Twice / Section 1

**Table 12** Source and Destination NAT Migration Examples (continued)

Description	Configuration Migration	Type / Section
Static and Dynamic commands for source and destination NAT	<p><b>Old Configuration</b></p> <pre>access-list NET1 permit ip host 192.168.1.1 host 192.168.1.10 access-list NET2 permit ip host 209.165.200.225 host 209.165.200.228 static (outside,inside) 192.168.1.10 access-list NET2 global (outside) 100 209.165.200.228 nat (inside) 100 access-list NET1</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-192.168.1.1   host 192.168.1.1 object network obj-209.165.200.228   host 209.165.200.228 object network obj-209.165.200.225   host 209.165.200.225 object network obj-192.168.1.10   host 192.168.1.10  nat (inside,outside) source dynamic obj-192.168.1.1 obj-209.165.200.228 destination <b>static obj-192.168.1.10 obj-209.165.200.225</b>  (The following rule is created by the migration script, but it may not be necessary; in rare circumstances, traffic might use this rule.)  nat (inside,outside) source dynamic obj-192.168.1.1 obj-209.165.200.228 destination static obj-192.168.1.10 obj-192.168.1.10  nat (outside,inside) source static obj-209.165.200.225 obj-192.168.1.10 destination static obj-209.165.200.228 obj-209.165.200.228</pre>	Twice / Section 1

## alias Command

The **alias** command translates addresses on an IP network residing on any interface into addresses on another IP network connected through a different interface.

[Table 13](#) lists alias migration examples.

**Table 13** alias Command Migration Examples

Description	Configuration Migration	Type / Section
Alias command	<p><b>Old Configuration</b></p> <pre>alias (inside) 209.165.200.225 192.168.100.10</pre> <p><b>Migrated Configuration</b></p> <pre>object network obj-192.168.100.10   host 192.168.100.10 nat (any,inside) static 209.165.200.225 dns</pre>	Object / Section 2

## NAT Migration Messages

Some NAT configurations cannot be migrated automatically, or are slightly different from the original configuration. [Table 14](#) lists error messages you might see, and information about the messages.

**Table 14** NAT Migration Messages

---

### Message and Description

---

**Error Message** The following 'nat' command didn't have a matching 'global' rule on interface '<name>' and was not migrated.

**Explanation** Missing **global** command. If a **nat** command does not have a matching **global** command, the **nat** command will be removed and will not be migrated.

**Recommended Action** If you intended to have a matching **global** command, you will need to recreate the configuration using the new NAT commands.

#### Example:

##### Old Configuration

```
nat (dmz) 1 10.1.1.0 255.255.255.0
```

##### Migrated Configuration

Not migrated.

---

**Error Message** Alias command was migrated between interfaces 'any' and 'inside' as an estimate.

**Explanation** **alias** command migration. The **alias** command is applied between same and lower security level interfaces. After migration, the rules are added between a given interface and **any**. This is semantically different as the new rule applies to all interfaces including itself.

**Recommended Action** This is relatively safe to migrate and needs no attention in most cases. See the [“alias Command” section on page 34](#) for an example migration.

#### Example:

##### Old Configuration

```
alias (inside) 209.165.200.225 192.168.100.10
```

##### Migrated Configuration

```
object network obj-192.168.100.10
  host 192.168.100.10
  nat (any,inside) static 209.165.200.225 dns
```

---

**Table 14** NAT Migration Messages (continued)

---

**Message and Description**

---

**Error Message** Identity-NAT was not migrated. If required, an appropriate bypass NAT rule needs to be added.

**Explanation** Identity NAT not migrated. Identity NAT (the **nat 0** command) is not migrated; also a **nat-control** command on that interface is not migrated.

**Recommended Action** Manually add a new Identity NAT rule using a static NAT command (either object or twice NAT).

**Example:****Old Configuration**

```
nat (inside) 0 192.168.1.0 255.255.255.0
```

**Migrated Configuration**

Not migrated.

---

**Table 14** NAT Migration Messages (continued)**Message and Description**

**Error Message** Range a.b.c.d-p.q.r.s also includes broadcast address as mapped value.

**Explanation** Outside static policy NAT with overlapping destination and broadcast address. You used to be able to configure the old **global** command to automatically remove the broadcast addresses from the global pool by using a /31 subnet. You cannot configure the same functionality in the new NAT commands. If there is a dynamic NAT rule and an outside static policy NAT rule with overlapping destinations, then the migrated configuration will include the broadcast address in the mapped source. User intervention is required to manually remove those addresses.

**Recommended Action** Remove the broadcast address from the mapped object.

**Example:****Old Configuration**

```
nat (inside) 10 10.0.0.0 255.0.0.0
global (outside) 10 192.168.1.3-192.168.2.3 netmask 255.255.255.254
```

(The following broadcast address is automatically removed from the pool: 192.168.1.255.)

```
access-list SNAT extended permit ip 10.10.10.0 255.255.255.0 192.168.2.0 255.255.255.0

static (outside,inside) 10.1.1.0 access-list SNAT
```

**Migrated Configuration**

```
object network obj-192.168.1.3-192.168.2.3
  range 192.168.1.3 192.168.2.3
```

(192.168.1.255 is not automatically removed from this pool. To avoid assigning 192.168.1.255, you should instead create a network group, and use it in the **nat** command:

```
object network global_pool1
  range 192.168.1.3 192.168.1.254
object network global_pool2
  range 192.168.2.1 192.168.2.3
object-group network global_pool
  network-object object global_pool1
  network-object object global_pool2
)

object network obj-10.10.10.0
  subnet 10.10.10.0 255.255.255.0
object network obj-10.1.1.0
  subnet 10.1.1.0 255.255.255.0
object network obj-192.168.2.0
  subnet 192.168.2.0 255.255.255.0

object network obj-10.0.0.0
  subnet 10.0.0.0 255.0.0.0
  nat (inside,outside) dynamic obj-192.168.1.3-192.168.2.3

nat (inside,outside) source dynamic obj-10.0.0.0 obj-192.168.1.3-192.168.2.3 destination static obj-10.1.1.0
obj-10.10.10.0

nat (outside,inside) source static obj-10.10.10.0 obj-10.1.1.0 destination static obj-192.168.2.0
obj-192.168.2.0
```

Table 14 NAT Migration Messages (continued)

**Message and Description**

**Error Message** The `nodnsalias` option is deprecated. Use 'dns' option in `nat` command to enable/disable dns rewrite.

**Explanation** `sysopt nodnsalias` command not migrated. The `sysopt nodnsalias` command is deprecated because the `alias` command is no longer supported.

**Recommended Action** Use the `dns` option in the new NAT commands to enable/disable DNS rewrite.

**Example:****Old Configuration**

```
sysopt nodnsalias
```

**Migrated Configuration**

Not migrated.

## Network and Service Object Migration

This section describes network and service object migration and includes the following topics:

- [Supported Features for Objects, page 38](#)
- [Object Migration, page 38](#)

### Supported Features for Objects

Version 8.3 introduces named network and service objects for use with the following features:

- NAT—See the “[NAT Migration](#)” section on [page 15](#) for more information. You can no longer use a named IP address (using the `name` command) in NAT.
- Access lists—`access-list` command. You can no longer use a named IP address (using the `name` command) in an access list.
- Object groups—`object-group network` and `object-group service` commands. Named IP addresses are still allowed in object groups, as well as network objects.

### Object Migration

New network and service objects (the `object network` and `object service` commands) are substituted into existing commands in the following cases:

- For each network object NAT command, an `object network` command is created to represent the real IP address that you want to translate.
- When new `nat` commands require an object instead of an inline value, network and service objects are automatically created.

- If you use a named IP address in NAT (using the **name** command) and the **names** command is enabled, then a network object is created even if an inline IP address could be used in the new **nat** command.
- If an **access-list** command includes an IP address that was used in NAT, and the NAT migration created a network object for that IP address, then the network object replaces the IP address in the **access-list** command.
- If you use a named IP address in the **access-list** command (using the **name** command) and the **names** command is enabled, then an object replaces the name.
- For multiple **global** commands that share the same NAT ID, a network object group is created that contains the network objects created for the inline IP addresses.

Objects are not created for the following cases:

- A **name** command exists in the configuration, but is not used in a **nat** or **access-list** command.
- An inline value that is still allowed in the **nat** command.
- **name** commands used under **object-group** commands.
- IP addresses used in **access-list** commands that are not used in NAT or named with a **name** command.


**Note**

The **name** commands continue to exist in your configuration for use with other features that do not yet support network objects.

**ASDM**

ASDM has supported named network objects for a number of releases; now, the platform has the commands to properly support them as well. In addition to showing all named network objects in the configuration, ASDM automatically creates objects for any IP addresses used in the configuration; these auto-created objects are identified by the IP address, and are not present as objects in the platform configuration. If you assign a name to one of these objects, then ASDM adds the named network object to the platform configuration.


**Note**

ASDM no longer shows any objects derived from the **name** command. Previously, you might have used named objects derived from the **name** command in ASDM. If the **name** command IP address was not migrated, then these objects are replaced by auto-created objects identified by an IP address.

## Object Migration Naming Conventions

This section includes the following topics:

- [name Command Naming Conventions, page 40](#)
- [Inline IP Address Naming Conventions, page 40](#)
- [Inline Protocol Naming Conventions, page 40](#)
- [Network Object Naming Conventions with Multiple global Commands with the Same NAT ID, page 41](#)

For details about when a name or IP address is migrated, see the “Object Migration” section on page 38.

## name Command Naming Conventions

When the **names** command is enabled, then for migrated **name** commands, the same name is used for the **object network** command.

For example, for the following **name** command used in NAT:

```
name 10.1.1.1 test
```

An **object network** command is created:

```
object network test
  host 10.1.1.1
```

If the **names** command is not enabled, and IP addresses are migrated to network objects, then your configuration might include network objects where the IP addresses are the same as in **name** commands, but the name of the network object is automatically generated (see the [“Inline IP Address Naming Conventions”](#) section on page 40), and not the same name as the **name** command.

## Inline IP Address Naming Conventions

For migrated IP addresses used inline, network objects are created using the following naming convention:

- Hosts and subnets—**obj-a.b.c.d.**



**Note** Only one instance of NAT can be enabled on an object. If you have more than one NAT policy applied on a given host or subnet, then a separate network object will be created: **obj-a.b.c.d-01**.

Table 15 lists host and subnet inline object migration naming examples.

**Table 15** Host and Subnet Inline Object Migration Naming Examples

Inline Value	Network Object Name
10.76.6.111 255.255.255.255	obj-10.76.6.111
10.76.0.0 255.255.0.0	obj-10.76.0.0

- Ranges—**obj-a.b.c.d-p.q.r.s**

Table 16 lists range inline object migration naming examples.

**Table 16** Range Inline Object Migration Naming Examples

Inline Value	Network Object Name
10.76.6.111-10.76.6.112	obj-10.76.6.111-10.76.6.112

## Inline Protocol Naming Conventions

For migrated protocols used inline, service objects are created using the following naming convention: **obj-inline\_text**.



Table 17 lists protocol inline object migration naming examples.

**Table 17 Protocol Inline Object Migration Naming Examples**

Inline Value	Service Object Name
tcp source range 20 50 eq 2000	obj-tcp_source_range_20_50_eq_2000
tcp gt 1500	obj-tcp_gt_1500

## Network Object Naming Conventions with Multiple global Commands with the Same NAT ID

For multiple **global** commands that share the same NAT ID, a network object group is created that contains the network objects created for the inline IP addresses. The following naming convention is used: **og-global-interface\_nat-id**.

### Old Configuration

```
global (outside) 1 10.76.6.111
global (outside) 1 10.76.6.109-10.76.6.110
```

### New Network Objects and Groups

```
object network obj-10.76.6.111
  host 10.76.6.111
object network obj-10.76.6.109-10.76.6.110
  range 10.76.6.109-10.76.6.110
object-group og-global-outside_1
  network-object obj-10.76.6.111
  network-object obj-10.76.6.109-10.76.6.110
```

## Downgrading from Version 8.3

When you upgrade to Version 8.3, your configuration is migrated. The old configuration is automatically stored in flash memory. For example when you upgrade from 8.2(1) to 8.3(1), the old 8.2(1) configuration is stored in flash memory in a file called 8\_2\_1\_0\_startup\_cfg.sav.

This section describes how to downgrade, and includes the following topics:

- [Information About Activation Key Compatibility, page 41](#)
- [Performing the Downgrade, page 42](#)

## Information About Activation Key Compatibility

Your activation key remains compatible if you upgrade to the latest version from any previous version. However, you might have issues if you want to maintain downgrade capability:

- Downgrading to Version 8.1 or earlier—After you upgrade, if you activate additional feature licenses that were introduced *before 8.2*, then the activation key continues to be compatible with earlier versions if you downgrade. However if you activate feature licenses that were introduced in *8.2 or later*, then the activation key is not backwards compatible. If you have an incompatible license key, then see the following guidelines:
  - If you previously entered an activation key in an earlier version, then the adaptive security appliance uses that key (without any of the new licenses you activated in Version 8.2 or later).

- If you have a new system and do not have an earlier activation key, then you need to request a new activation key compatible with the earlier version.
- Downgrading to Version 8.2 or earlier—Version 8.3 introduced more robust time-based key usage as well as failover license changes:
  - If you have more than one time-based activation key active, when you downgrade, only the most recently activated time-based key can be active. Any other keys are made inactive. If the last time-based license is for a feature introduced in 8.3, then that license still remains the active license even though it cannot be used in earlier versions. Reenter the permanent key or a valid time-based key.
  - If you have mismatched licenses on a failover pair, then downgrading will disable failover. Even if the keys are matching, the license used will no longer be a combined license.
  - If you have one time-based license installed, but it is for a feature introduced in 8.3, then after you downgrade, that time-based license remains active. You need to reenter the permanent key to disable the time-based license.

## Performing the Downgrade

To downgrade from 8.3, perform the following steps.

### Detailed Steps

For the CLI:

**Step 1** Enter the following command:

```
hostname(config)# downgrade [/noconfirm] old_image_url old_config_url [activation-key
old_key]
```

Where the **/noconfirm** option downgrades without prompting. The *image\_url* is the path to the old image on disk0, disk1, tftp, ftp, or smb. The *old\_config\_url* is the path to the saved, pre-migration configuration (by default this was saved on disk0). If you need to revert to a pre-8.3 activation key, then you can enter the old activation key.

This command is a shortcut for completing the following functions:

1. Clearing the boot image configuration (**clear configure boot**).
2. Setting the boot image to be the old image (**boot system**).
3. (Optional) Entering a new activation key (**activation-key**).
4. Saving the running configuration to startup (**write memory**). This sets the BOOT environment variable to the old image, so when you reload, the old image is loaded.
5. Copying the old configuration to the startup configuration (**copy old\_config\_url startup-config**).
6. Reloading (**reload**).

For example:

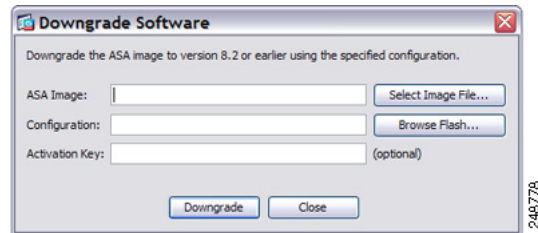
```
hostname(config)# downgrade /noconfirm disk0:/asa821-k8.bin disk0:/8_2_1_0_startup_cfg.sav
```

For ASDM:

**Step 1** Choose **Tools > Downgrade Software**.

The Downgrade Software dialog box appears.

**Figure 1** Downgrade Software

**Step 2** For the ASA Image, click **Select Image File**.

The Browse File Locations dialog box appears.

**Step 3** Click one of the following radio buttons:

- **Remote Server**—Choose **ftp**, **smb**, or **http** from the drop-down list, and type the path to the old image file.
- **Flash File System**—Click **Browse Flash** to choose the old image file on the local flash file system.

**Step 4** For the Configuration, click **Browse Flash** to choose the pre-migration configuration file. (By default this was saved on disk0).**Step 5** (Optional) In the Activation Key field, enter the old activation key if you need to revert to a pre-8.3 activation key.**Step 6** Click **Downgrade**.

This tool is a shortcut for completing the following functions:

1. Clearing the boot image configuration (**clear configure boot**).
2. Setting the boot image to be the old image (**boot system**).
3. (Optional) Entering a new activation key (**activation-key**).
4. Saving the running configuration to startup (**write memory**). This sets the BOOT environment variable to the old image, so when you reload, the old image is loaded.
5. Copying the old configuration to the startup configuration (**copy old\_config\_url startup-config**).
6. Reloading (**reload**).

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Copyright © 2010-2011 Cisco Systems, Inc. All rights reserved.

