Network Address Translation (NAT)

The following topics explain Network Address Translation (NAT) and how to configure it.

- Why Use NAT?, page 4-1
- NAT Basics, page 4-2
- Guidelines for NAT, page 4-6
- Dynamic NAT, page 4-12
- Dynamic PAT, page 4-18
- Static NAT, page 4-27
- Identity NAT, page 4-37
- Monitoring NAT, page 4-40
- History for NAT, page 4-41

Why Use NAT?

Each computer and device within an IP network is assigned a unique IP address that identifies the host. Because of a shortage of public IPv4 addresses, most of these IP addresses are private, not routable anywhere outside of the private company network. RFC 1918 defines the private IP addresses you can use internally that should not be advertised:

- 10.0.0.0 through 10.255.255.255
- 172.16.0.0 through 172.31.255.255
- 192.168.0.0 through 192.168.255.255

One of the main functions of NAT is to enable private IP networks to connect to the Internet. NAT replaces a private IP address with a public IP address, translating the private addresses in the internal private network into legal, routable addresses that can be used on the public Internet. In this way, NAT conserves public addresses because it can be configured to advertise at a minimum only one public address for the entire network to the outside world.

Other functions of NAT include:

- Security—Keeping internal IP addresses hidden discourages direct attacks.
- IP routing solutions—Overlapping IP addresses are not a problem when you use NAT.
NAT Basics

The following topics explain some of the basics of NAT.

- NAT Terminology, page 4-2
- NAT Types, page 4-3
- Network Object NAT and Twice NAT, page 4-3
- NAT Rule Order, page 4-5
- NAT Interfaces, page 4-6

NAT Terminology

This document uses the following terminology:

- Real address/host/network/interface—The real address is the address that is defined on the host, before it is translated. In a typical NAT scenario where you want to translate the inside network when it accesses the outside, the inside network would be the “real” network. Note that you can translate any network connected to the ASA, not just an inside network. Therefore if you configure NAT to translate outside addresses, “real” can refer to the outside network when it accesses the inside network.

- Mapped address/host/network/interface—The mapped address is the address that the real address is translated to. In a typical NAT scenario where you want to translate the inside network when it accesses the outside, the outside network would be the “mapped” network.

Note During address translation, IP addresses residing on the ASA’s interfaces are not translated.

- Bidirectional initiation—Static NAT allows connections to be initiated bidirectionally, meaning both to the host and from the host.

- Source and destination NAT—For any given packet, both the source and destination IP addresses are compared to the NAT rules, and one or both can be translated/untranslated. For static NAT, the rule is bidirectional, so be aware that “source” and “destination” are used in commands and descriptions throughout this guide even though a given connection might originate at the “destination” address.
NAT Types

You can implement NAT using the following methods:

- Dynamic NAT—A group of real IP addresses are mapped to a (usually smaller) group of mapped IP addresses, on a first come, first served basis. Only the real host can initiate traffic. See Dynamic NAT, page 4-12.
- Dynamic Port Address Translation (PAT)—A group of real IP addresses are mapped to a single IP address using a unique source port of that IP address. See Dynamic PAT, page 4-18.
- Static NAT—A consistent mapping between a real and mapped IP address. Allows bidirectional traffic initiation. See Static NAT, page 4-27.
- Identity NAT—A real address is statically translated to itself, essentially bypassing NAT. You might want to configure NAT this way when you want to translate a large group of addresses, but then want to exempt a smaller subset of addresses. See Identity NAT, page 4-37.

Network Object NAT and Twice NAT

The ASA can implement address translation in two ways: network object NAT and twice NAT.

We recommend using network object NAT unless you need the extra features that twice NAT provides. Network object NAT is easier to configure, and might be more reliable for applications such as Voice over IP (VoIP). (For VoIP, because twice NAT is applicable only between two objects, you might see a failure in the translation of indirect addresses that do not belong to either of the objects.)

- Network Object NAT, page 4-3
- Twice NAT, page 4-3
- Comparing Network Object NAT and Twice NAT, page 4-4

Network Object NAT

All NAT rules that are configured as a parameter of a network object are considered to be network object NAT rules. Network object NAT is a quick and easy way to configure NAT for a network object, which can be a single IP address, a range of addresses, or a subnet.

After you configure the network object, you can then identify the mapped address for that object, either as an inline address or as another network object or network object group.

When a packet enters the ASA, both the source and destination IP addresses are checked against the network object NAT rules. The source and destination address in the packet can be translated by separate rules if separate matches are made. These rules are not tied to each other; different combinations of rules can be used depending on the traffic.

Because the rules are never paired, you cannot specify that sourceA/destinationA should have a different translation than sourceA/destinationB. Use twice NAT for that kind of functionality (twice NAT lets you identify the source and destination address in a single rule).

Twice NAT

Twice NAT lets you identify both the source and destination address in a single rule. Specifying both the source and destination addresses lets you specify that sourceA/destinationA can have a different translation than sourceA/destinationB.
For static NAT, the rule is bidirectional, so be aware that “source” and “destination” are used in commands and descriptions throughout this guide even though a given connection might originate at the “destination” address. For example, if you configure static NAT with port address translation, and specify the source address as a Telnet server, and you want all traffic going to that Telnet server to have the port translated from 2323 to 23, then in the command, you must specify the source ports to be translated (real: 23, mapped: 2323). You specify the source ports because you specified the Telnet server address as the source address.

The destination address is optional. If you specify the destination address, you can either map it to itself (identity NAT), or you can map it to a different address. The destination mapping is always a static mapping.

Twice NAT also lets you use service objects for static NAT with port translation; network object NAT only accepts inline definition.

**Comparing Network Object NAT and Twice NAT**

The main differences between these two NAT types are:

- **How you define the real address.**
  - Network object NAT—You define NAT as a parameter for a network object. A network object names an IP host, range, or subnet so you can then use the object in the NAT configuration instead of the actual IP addresses. The network object IP address serves as the real address. This method lets you easily add NAT to network objects that might already be used in other parts of your configuration.
  - Twice NAT—You identify a network object or network object group for both the real and mapped addresses. In this case, NAT is not a parameter of the network object; the network object or group is a parameter of the NAT configuration. The ability to use a network object group for the real address means that twice NAT is more scalable.

- **How source and destination NAT is implemented.**
  - Network object NAT—Each rule can apply to either the source or destination of a packet. So two rules might be used, one for the source IP address, and one for the destination IP address. These two rules cannot be tied together to enforce a specific translation for a source/destination combination.
  - Twice NAT—A single rule translates both the source and destination. A matching packet only matches the one rule, and further rules are not checked. Even if you do not configure the optional destination address for twice NAT, a matching packet still only matches one twice NAT rule. The source and destination are tied together, so you can enforce different translations depending on the source/destination combination. For example, sourceA/destinationA can have a different translation than sourceA/destinationB.

- **Order of NAT Rules.**
  - Network object NAT—Automatically ordered in the NAT table.
  - Twice NAT—Manually ordered in the NAT table (before or after network object NAT rules).
NAT Rule Order

Network object NAT rules and twice NAT rules are stored in a single table that is divided into three sections. Section 1 rules are applied first, then section 2, and finally section 3, until a match is found. For example, if a match is found in section 1, sections 2 and 3 are not evaluated. The following table shows the order of rules within each section.

<table>
<thead>
<tr>
<th>Table 4-1</th>
<th>NAT Rule Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table Section</strong></td>
<td><strong>Rule Type</strong></td>
</tr>
<tr>
<td>Section 1</td>
<td>Twice NAT</td>
</tr>
<tr>
<td>Section 2</td>
<td>Network object NAT</td>
</tr>
<tr>
<td>Section 3</td>
<td>Twice NAT</td>
</tr>
</tbody>
</table>

For section 2 rules, for example, you have the following IP addresses defined within network objects:

- 192.168.1.0/24 (static)
192.168.1.0/24 (dynamic)
10.1.1.0/24 (static)
192.168.1.1/32 (static)
172.16.1.0/24 (dynamic) (object def)
172.16.1.0/24 (dynamic) (object abc)

The resultant ordering would be:
192.168.1.1/32 (static)
10.1.1.0/24 (static)
192.168.1.0/24 (static)
172.16.1.0/24 (dynamic) (object abc)
172.16.1.0/24 (dynamic) (object def)
192.168.1.0/24 (dynamic)

**NAT Interfaces**

In routed mode, you can configure a NAT rule to apply to any interface (in other words, all interfaces), or you can identify specific real and mapped interfaces. You can also specify any interface for the real address, and a specific interface for the mapped address, or vice versa.

For example, you might want to specify any interface for the real address and specify the outside interface for the mapped address if you use the same private addresses on multiple interfaces, and you want to translate them all to the same global pool when accessing the outside.

![Figure 4-1 Specifying Any Interface](image)

In transparent mode, you must choose specific source and destination interfaces.

**Guidelines for NAT**

The following topics provide detailed guidelines for implementing NAT.
- Firewall Mode Guidelines for NAT, page 4-7
- IPv6 NAT Guidelines, page 4-7
Guidelines for NAT

Firewall Mode Guidelines for NAT

NAT is supported in routed and transparent firewall mode. However, transparent mode has the following restrictions:

- In transparent mode, you must specify the real and mapped interfaces; you cannot specify “any” as the interface.
- In transparent mode, you cannot configure interface PAT, because the transparent mode interfaces do not have IP addresses. You also cannot use the management IP address as a mapped address.
- In transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.

IPv6 NAT Guidelines

NAT supports IPv6 with the following guidelines and restrictions.

- For routed mode, you can also translate between IPv4 and IPv6.
- For transparent mode, translating between IPv4 and IPv6 networks is not supported. Translating between two IPv6 networks, or between two IPv4 networks is supported.
- For transparent mode, a PAT pool is not supported for IPv6.
- For static NAT, you can specify an IPv6 subnet up to /64. Larger subnets are not supported.
- When using FTP with NAT46, when an IPv4 FTP client connects to an IPv6 FTP server, the client must use either the extended passive mode (EPSV) or extended port mode (EPRT); PASV and PORT commands are not supported with IPv6.

IPv6 NAT Recommendations

You can use NAT to translate between IPv6 networks, and also to translate between IPv4 and IPv6 networks (routed mode only). We recommend the following best practices:

- NAT66 (IPv6-to-IPv6)—We recommend using static NAT. Although you can use dynamic NAT or PAT, IPv6 addresses are in such large supply, you do not have to use dynamic NAT. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only).
- NAT46 (IPv4-to-IPv6)—We recommend using static NAT. Because the IPv6 address space is so much larger than the IPv4 address space, you can easily accommodate a static translation. If you do not want to allow returning traffic, you can make the static NAT rule unidirectional (twice NAT only). When translating to an IPv6 subnet (/64 or lower), the resulting mapped address is by default an IPv4-embedded IPv6 address, where the 32-bits of the IPv4 address is embedded after the IPv6 prefix. For example, if the IPv6 prefix is a /96 prefix, then the IPv4 address is appended in the last 32-bits of the address. For example, if you map 192.168.1.0/24 to 201b::0/96, then 192.168.1.4 will
be mapped to 201b::0.192.168.1.4 (shown with mixed notation). If the prefix is smaller, such as /64, then the IPv4 address is appended after the prefix, and a suffix of 0s is appended after the IPv4 address. You can also optionally translate the addresses net-to-net, where the first IPv4 address maps to the first IPv6 address, the second to the second, and so on.

- **NAT64 (IPv6-to-IPv4)**—You may not have enough IPv4 addresses to accommodate the number of IPv6 addresses. We recommend using a dynamic PAT pool to provide a large number of IPv4 translations.

### Additional Guidelines for NAT

- (Network object NAT only.) You can only define a single NAT rule for a given object; if you want to configure multiple NAT rules for an object, you need to create multiple objects with different names that specify the same IP address, for example, `object network obj-10.10.10.1-01`, `object network obj-10.10.10.1-02`, and so on.

- (Twice NAT only.) You cannot configure FTP destination port translation when the source IP address is a subnet (or any other application that uses a secondary connection); the FTP data channel establishment does not succeed. For example, the following configuration does not work:

```
object network MyInsNet
  subnet 10.1.2.0 255.255.255.0
object network MapInsNet
  subnet 209.165.202.128 255.255.255.224
object network Server1
  host 209.165.200.225
object network Server1_mapped
  host 10.1.2.67
object service REAL_ftp
  service tcp destination eq ftp
object service MAPPED_ftp
  service tcp destination eq 2021
object network MyOutNet
  subnet 209.165.201.0 255.255.255.224

nat (inside,outside) source static MyInsNet MapInsNet destination static Server1_mapped Server1 service MAPPED_ftp REAL_ftp
```

- If you change the NAT configuration, and you do not want to wait for existing translations to time out before the new NAT configuration is used, you can clear the translation table using the `clear xlate` command. However, clearing the translation table disconnects all current connections that use translations.

  **Note**
  
  If you remove a dynamic NAT or PAT rule, and then add a new rule with mapped addresses that overlap the addresses in the removed rule, then the new rule will not be used until all connections associated with the removed rule time out or are cleared using the `clear xlate` command. This safeguard ensures that the same address is not assigned to multiple hosts.

- Objects and object groups used in NAT cannot be undefined; they must include IP addresses.

- You cannot use an object group with both IPv4 and IPv6 addresses; the object group must include only one type of address.

- (Twice NAT only.) When using the `any` keyword in a NAT rule, the definition of “any” traffic (IPv4 vs. IPv6) depends on the rule. Before the ASA performs NAT on a packet, the packet must be IPv6-to-IPv6 or IPv4-to-IPv4; with this prerequisite, the ASA can determine the value of `any` in a NAT rule. For example, if you configure a rule from “any” to an IPv6 server, and that server was
mapped from an IPv4 address, then **any** means “any IPv6 traffic.” If you configure a rule from “any” to “any,” and you map the source to the interface IPv4 address, then **any** means “any IPv4 traffic” because the mapped interface address implies that the destination is also IPv4.

- You can use the same mapped object or group in multiple NAT rules.
- The mapped IP address pool cannot include:
  - The mapped interface IP address. If you specify “any” interface for the rule, then all interface IP addresses are disallowed. For interface PAT (routed mode only), use the `interface` keyword instead of the IP address.
  - (Transparent mode) The management IP address.
  - (Dynamic NAT) The standby interface IP address when VPN is enabled.
  - Existing VPN pool addresses.
- Avoid using overlapping addresses in static and dynamic NAT policies. For example, with overlapping addresses, a PPTP connection can fail to get established if the secondary connection for PPTP hits the static instead of dynamic xlate.
- For application inspection limitations with NAT or PAT, see Default Inspections and NAT Limitations, page 6-6.
- The default behavior for identity NAT has proxy ARP enabled, matching other static NAT rules. You can disable proxy ARP if desired. See Routing NAT Packets, page 5-11 for more information.
- If you specify an optional interface, then the ASA uses the NAT configuration to determine the egress interface, but you have the option to always use a route lookup instead. See Routing NAT Packets, page 5-11 for more information.
- You can improve system performance and reliability by using the transactional commit model for NAT. See the basic settings chapter in the general operations configuration guide for more information. Use the `asp rule-engine transactional-commit nat` command.

**Network Object NAT Guidelines for Mapped Address Objects**

For dynamic NAT, you must use an object or group for the mapped addresses. For the other NAT types, you can use an object or group, or you have the option of using inline addresses. Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. Use the `object network` and `object-group network` commands to create the objects.

Consider the following guidelines when creating objects for mapped addresses.

- A network object group can contain objects or inline addresses of either IPv4 or IPv6 addresses. The group cannot contain both IPv4 and IPv6 addresses; it must contain one type only.
- See Additional Guidelines for NAT, page 4-8 for information about disallowed mapped IP addresses.
- Dynamic NAT:
  - You cannot use an inline address; you must configure a network object or group.
  - The object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
  - If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and then the host IP addresses are used as a PAT fallback.
- Dynamic PAT (Hide):
Instead of using an object, you can optionally configure an inline host address or specify the interface address.

If you use an object, the object or group cannot contain a subnet. The object must define a host, or for a PAT pool, a range. The group (for a PAT pool) can include hosts and ranges.

- Static NAT or Static NAT with port translation:
  - Instead of using an object, you can configure an inline address or specify the interface address (for static NAT-with-port-translation).
  - If you use an object, the object or group can contain a host, range, or subnet.

- Identity NAT
  - Instead of using an object, you can configure an inline address.
  - If you use an object, the object must match the real addresses you want to translate.

**Twice NAT Guidelines for Real and Mapped Address Objects**

For each NAT rule, configure up to four network objects or groups for:

- Source real address
- Source mapped address
- Destination real address
- Destination mapped address

Objects are required unless you specify the **any** keyword inline to represent all traffic, or for some types of NAT, the **interface** keyword to represent the interface address. Network object groups are particularly useful for creating a mapped address pool with discontinuous IP address ranges or multiple hosts or subnets. Use the **object network** and **object-group network** commands to create the objects.

Consider the following guidelines when creating objects for twice NAT.

- A network object group can contain objects or inline addresses of either IPv4 or IPv6 addresses. The group cannot contain both IPv4 and IPv6 addresses; it must contain one type only.
- See Additional Guidelines for NAT, page 4-8 for information about disallowed mapped IP addresses.
- Source Dynamic NAT:
  - You typically configure a larger group of real addresses to be mapped to a smaller group.
  - The mapped object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
  - If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and the host IP addresses are used as a PAT fallback.
- Source Dynamic PAT (Hide):
  - If you use an object, the object or group cannot contain a subnet. The object must define a host, or for a PAT pool, a range. The group (for a PAT pool) can include hosts and ranges.
- Source Static NAT or Static NAT with port translation:
  - The mapped object or group can contain a host, range, or subnet.
  - The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired.
• Source Identity NAT
  – The real and mapped objects must match. You can use the same object for both, or you can create separate objects that contain the same IP addresses.

• Destination Static NAT or Static NAT with port translation (the destination translation is always static):
  – Although the main feature of twice NAT is the inclusion of the destination IP address, the destination address is optional. If you do specify the destination address, you can configure static translation for that address or just use identity NAT for it. You might want to configure twice NAT without a destination address to take advantage of some of the other qualities of twice NAT, including the use of network object groups for real addresses, or manually ordering of rules. For more information, see Comparing Network Object NAT and Twice NAT, page 4-4.
  – For identity NAT, the real and mapped objects must match. You can use the same object for both, or you can create separate objects that contain the same IP addresses.
  – The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired.
  – For static interface NAT with port translation (routed mode only), you can specify the interface keyword instead of a network object/group for the mapped address.

Twice NAT Guidelines for Service Objects for Real and Mapped Ports

You can optionally configure service objects for:

• Source real port (Static only) or Destination real port
• Source mapped port (Static only) or Destination mapped port

Use the object service command to create the objects.

Consider the following guidelines when creating objects for twice NAT.

• NAT only supports TCP or UDP. When translating a port, be sure the protocols in the real and mapped service objects are identical (both TCP or both UDP).
• The “not equal” (neq) operator is not supported.
• For identity port translation, you can use the same service object for both the real and mapped ports.
• Source Dynamic NAT—Source Dynamic NAT does not support port translation.
• Source Dynamic PAT (Hide)—Source Dynamic PAT does not support port translation.
• Source Static NAT, Static NAT with port translation, or Identity NAT—A service object can contain both a source and destination port; however, you should specify either the source or the destination port for both service objects. You should only specify both the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. For example, if you want to translate the port for the source host, then configure the source service.
• Destination Static NAT or Static NAT with port translation (the destination translation is always static)—For non-static source NAT, you can only perform port translation on the destination. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored.
Dynamic NAT

The following topics explain dynamic NAT and how to configure it.

- **About Dynamic NAT, page 4-12**
- **Configure Dynamic Network Object NAT, page 4-14**
- **Configure Dynamic Twice NAT, page 4-16**

### About Dynamic NAT

Dynamic NAT translates a group of real addresses to a pool of mapped addresses that are routable on the destination network. The mapped pool typically includes fewer addresses than the real group. When a host you want to translate accesses the destination network, the ASA assigns the host an IP address from the mapped pool. The translation is created only when the real host initiates the connection. The translation is in place only for the duration of the connection, and a given user does not keep the same IP address after the translation times out. Users on the destination network, therefore, cannot initiate a reliable connection to a host that uses dynamic NAT, even if the connection is allowed by an access rule.

---

**Note**

For the duration of the translation, a remote host can initiate a connection to the translated host if an access rule allows it. Because the address is unpredictable, a connection to the host is unlikely. Nevertheless, in this case you can rely on the security of the access rule.

---

The following figure shows a typical dynamic NAT scenario. Only real hosts can create a NAT session, and responding traffic is allowed back.

**Figure 4-2 Dynamic NAT**

![Dynamic NAT Diagram](image-url)
The following figure shows a remote host attempting to initiate a connection to a mapped address. This address is not currently in the translation table; therefore, the ASA drops the packet.

**Figure 4-3  Remote Host Attempts to Initiate a Connection to aMapped Address**

Dynamic NAT Disadvantages and Advantages

Dynamic NAT has these disadvantages:

- If the mapped pool has fewer addresses than the real group, you could run out of addresses if the amount of traffic is more than expected.
  
  Use PAT or a PAT fall-back method if this event occurs often because PAT provides over 64,000 translations using ports of a single address.

- You have to use a large number of routable addresses in the mapped pool, and routable addresses may not be available in large quantities.

The advantage of dynamic NAT is that some protocols cannot use PAT. PAT does not work with the following:

- IP protocols that do not have a port to overload, such as GRE version 0.
- Some multimedia applications that have a data stream on one port, the control path on another port, and are not open standard.

See Default Inspections and NAT Limitations, page 6-6 for more information about NAT and PAT support.
Configure Dynamic Network Object NAT

This section describes how to configure network object NAT for dynamic NAT.

**Procedure**

**Step 1**
Create a host or range network object (**object network** command), or a network object group (**object-group network** command), for the mapped addresses.

- The object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
- If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and then the host IP addresses are used as a PAT fallback.

**Step 2**
Create or edit the network object for which you want to configure NAT.

```
object network obj_name
```

Example

```
hostname(config)# object network my-host-obj1
```

**Step 3**
(Skip when editing an object that has the right address.) Define the real IPv4 or IPv6 addresses that you want to translate.

- **host** `{IPv4_address | IPv6_address}`—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8::0DB8:800:200C:417A.
- **subnet** `{IPv4_address IPv4_mask | IPv6_address/IPv6_prefix}`—The address of a network. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8:0:CD30::/60.
- **range** `start_address end_address`—A range of addresses. You can specify IPv4 or IPv6 ranges. Do not include masks or prefixes.

Example

```
hostname(config-network-object)# host 10.2.2.2
```

**Step 4**
Configure dynamic NAT for the object IP addresses. You can only define a single NAT rule for a given object.

```
nat [(real_ifc,mapped_ifc)] dynamic mapped_obj [interface [ipv6]] [dns]
```

Example

```
hostname(config-network-object)# nat (inside,outside) dynamic MAPPED_IPS interface
```

Where:
- Interfaces—(Required for transparent mode) Specify the real (**real_ifc**) and mapped (**mapped_ifc**) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces, for example (any,outside).
- Mapped IP address—Specify the network object or network object group that includes the mapped IP addresses.
• Interface PAT fallback—(Optional) The **interface** keyword enables interface PAT fallback. After the mapped IP addresses are used up, then the IP address of the mapped interface is used. If you specify **ipv6**, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the mapped_ifc. (You cannot specify **interface** in transparent mode).

• DNS—(Optional) The **dns** keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 5-21 for more information.

### Examples
The following example configures dynamic NAT that hides the 192.168.2.0 network behind a range of outside addresses 10.2.2.1 through 10.2.2.10:

```
hostname(config)# object network my-range-obj
hostname(config-network-object)# range 10.2.2.1 10.2.2.10
hostname(config-network-object)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic my-range-obj
```

The following example configures dynamic NAT with dynamic PAT backup. Hosts on inside network 10.76.11.0 are mapped first to the nat-range1 pool (10.10.10.10-10.10.10.20). After all addresses in the nat-range1 pool are allocated, dynamic PAT is performed using the pat-ip1 address (10.10.10.21). In the unlikely event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
hostname(config)# object network nat-range1
hostname(config-network-object)# range 10.10.10.10 10.10.10.20
hostname(config-network-object)# object network pat-ip1
hostname(config-network-object)# host 10.10.10.21
hostname(config-network-object)# object-group network nat-pat-grp
hostname(config-network-object)# network-object object nat-range1
hostname(config-network-object)# network-object object pat-ip1
hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 10.76.11.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic my-range-obj
```

The following example configures dynamic NAT with dynamic PAT backup to translate IPv6 hosts to IPv4. Hosts on inside network 2001:DB8::/96 are mapped first to the IPv4_NAT_RANGE pool (209.165.201.1 to 209.165.201.30). After all addresses in the IPv4_NAT_RANGE pool are allocated, dynamic PAT is performed using the IPv4_PAT address (209.165.201.31). In the event that the PAT translations are also used up, dynamic PAT is performed using the outside interface address.

```
hostname(config)# object network IPv4_NAT_RANGE
hostname(config-network-object)# range 209.165.201.1 209.165.201.30
hostname(config-network-object)# object network IPv4_PAT
hostname(config-network-object)# host 209.165.201.31
hostname(config-network-object)# object-group network IPv4_GROUP
hostname(config-network-object)# network-object object IPv4_NAT_RANGE
hostname(config-network-object)# network-object object IPv4_PAT
hostname(config-network-object)# object network my_net_obj5
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(config-network-object)# nat (inside,outside) dynamic IPv4_GROUP
```
Configure Dynamic Twice NAT

This section describes how to configure twice NAT for dynamic NAT.

Procedure

**Step 1**  Create host or range network objects (**object network** command), or network object groups (**object-group network** command), for the source real addresses, the source mapped addresses, the destination real addresses, and the destination mapped addresses.

- If you want to translate all source traffic, you can skip adding an object for the source real addresses, and instead specify the **any** keyword in the **nat** command.
- If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the **interface** keyword in the **nat** command.

If you do create objects, consider the following guidelines:

- You typically configure a larger group of real addresses to be mapped to a smaller group.
- The object or group cannot contain a subnet; the object must define a range; the group can include hosts and ranges.
- If a mapped network object contains both ranges and host IP addresses, then the ranges are used for dynamic NAT, and then the host IP addresses are used as a PAT fallback.

**Step 2**  (Optional.) Create service objects for the destination real ports and the destination mapped ports.

For dynamic NAT, you can only perform port translation on the destination. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored.

**Step 3**  Configure dynamic NAT.

```
hostname(config)# nat (inside,outside) source dynamic MyInsNet NAT_POOL 
   destination static Server1_mapped Server1 service MAPPED_SVC REAL_SVC
```

Where:

- Interfaces—(Required for transparent mode) Specify the real (**real_ifc**) and mapped (**mapped_ifc**) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces, for example (any,outside).
- Section and Line—(Optional.) By default, the NAT rule is added to the end of section 1 of the NAT table (see **NAT Rule Order**, page 4-5). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the **after-auto** keyword. You can insert a rule anywhere in the applicable section using the **line** argument.
- Source addresses:
  - Real—Specify a network object, group, or the **any** keyword.
Dynamic NAT

- Mapped—Specify a different network object or group. You can optionally configure the
  following fallback method:

  Interface PAT fallback—(Routed mode only) The interface keyword enables interface PAT
  fallback. If you specify ipv6, then the IPv6 address of the interface is used. After the mapped
  IP addresses are used up, then the IP address of the mapped interface is used. For this option,
  you must configure a specific interface for the mapped_ifc.

- Destination addresses (Optional):
  - Mapped—Specify a network object or group, or for static interface NAT with port translation
    only, specify the interface keyword. If you specify ipv6, then the IPv6 address of the interface
    is used. If you specify interface, be sure to also configure the service keyword. For this option,
    you must configure a specific interface for the real_ifc. See Static Interface NAT with Port
    Translation, page 4-29 for more information.
  - Real—Specify a network object or group. For identity NAT, simply use the same object or group
    for both the real and mapped addresses.

- Destination port—(Optional.) Specify the service keyword along with the mapped and real service
  objects. For identity port translation, simply use the same service object for both the real and
  mapped ports.

- DNS—(Optional; for a source-only rule.) The dns keyword translates DNS replies. Be sure DNS
  inspection is enabled (it is enabled by default). You cannot configure the dns keyword if you
  configure a destination address. See DNS and NAT, page 5-21 for more information.

- Unidirectional—(Optional.) Specify unidirectional so the destination addresses cannot initiate
  traffic to the source addresses.

- Inactive—(Optional.) To make this rule inactive without having to remove the command, use the
  inactive keyword. To reactivate it, reenter the whole command without the inactive keyword.

- Description—Optional.) Provide a description up to 200 characters using the description keyword.

Examples
The following example configures dynamic NAT for inside network 10.1.1.0/24 when accessing servers
on the 209.165.201.1/27 network as well as servers on the 203.0.113.0/24 network:

hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0

hostname(config)# object network MAPPED_1
hostname(config-network-object)# range 209.165.200.225 209.165.200.254

hostname(config)# object network MAPPED_2
hostname(config-network-object)# range 209.165.202.129 209.165.200.158

hostname(config)# object network SERVERS_1
hostname(config-network-object)# subnet 209.165.201.0 255.255.255.224

hostname(config)# object network SERVERS_2
hostname(config-network-object)# subnet 203.0.113.0 255.255.255.0

hostname(config)# nat (inside, outside) source dynamic INSIDE_NW MAPPED_1 destination
static SERVERS_1 SERVERS_1
hostname(config)# nat (inside, outside) source dynamic INSIDE_NW MAPPED_2 destination
static SERVERS_2 SERVERS_2
The following example configures dynamic NAT for an IPv6 inside network 2001:DB8:AAAA::/96 when accessing servers on the IPv4 209.165.201.1/27 network as well as servers on the 203.0.113.0/24 network:

```
hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96

hostname(config)# object network MAPPED_1
hostname(config-network-object)# range 209.165.200.225 209.165.200.254

hostname(config)# object network MAPPED_2
hostname(config-network-object)# range 209.165.202.129 209.165.200.158

hostname(config)# object network SERVERS_1
hostname(config-network-object)# subnet 209.165.201.0 255.255.255.224

hostname(config)# object network SERVERS_2
hostname(config-network-object)# subnet 203.0.113.0 255.255.255.0

hostname(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_1 destination static SERVERS_1 SERVERS_1
hostname(config)# nat (inside,outside) source dynamic INSIDE_NW MAPPED_2 destination static SERVERS_2 SERVERS_2
```

### Dynamic PAT

The following topics describe dynamic PAT.

- About Dynamic PAT, page 4-18
- Configure Dynamic Network Object PAT, page 4-20
- Configure Dynamic Twice PAT, page 4-22
- Configure Per-Session PAT or Multi-Session PAT, page 4-25

### About Dynamic PAT

Dynamic PAT translates multiple real addresses to a single mapped IP address by translating the real address and source port to the mapped address and a unique port. If available, the real source port number is used for the mapped port. However, if the real port is not available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. If you have a lot of traffic that uses the lower port ranges, you can specify a flat range of ports to be used instead of the three unequal-sized tiers.

Each connection requires a separate translation session because the source port differs for each connection. For example, 10.1.1.1:1025 requires a separate translation from 10.1.1.1:1026.
The following figure shows a typical dynamic PAT scenario. Only real hosts can create a NAT session, and responding traffic is allowed back. The mapped address is the same for each translation, but the port is dynamically assigned.

![Dynamic PAT Diagram]

After the connection expires, the port translation also expires. For multi-session PAT, the PAT timeout is used, 30 seconds by default. For per-session PAT, the xlate is immediately removed. Users on the destination network cannot reliably initiate a connection to a host that uses PAT (even if the connection is allowed by an access rule).

**Note**

For the duration of the translation, a remote host can initiate a connection to the translated host if an access rule allows it. Because the port address (both real and mapped) is unpredictable, a connection to the host is unlikely. Nevertheless, in this case you can rely on the security of the access rule.

**Dynamic PAT Disadvantages and Advantages**

Dynamic PAT lets you use a single mapped address, thus conserving routable addresses. You can even use the ASA interface IP address as the PAT address.

Dynamic PAT does not work with some multimedia applications that have a data stream that is different from the control path. See Default Inspections and NAT Limitations, page 6-6 for more information about NAT and PAT support.

Dynamic PAT might also create a large number of connections appearing to come from a single IP address, and servers might interpret the traffic as a DoS attack. You can configure a PAT pool of addresses and use a round-robin assignment of PAT addresses to mitigate this situation.

**PAT Pool Object Guidelines**

When creating network objects for a PAT pool, follow these guidelines.

**For a PAT pool**

- If available, the real source port number is used for the mapped port. However, if the real port is not available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool that can be used. (8.4(3) and later, not including 8.5(1) or 8.6(1)) If you have a lot of traffic that uses the lower port ranges, you can specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.

- If you use the same PAT pool object in two separate rules, then be sure to specify the same options for each rule. For example, if one rule specifies extended PAT and a flat range, then the other rule must also specify extended PAT and a flat range.
Dynamic PAT

For extended PAT for a PAT pool

- Many application inspections do not support extended PAT. See Default Inspections and NAT Limitations, page 6-6 for a complete list of unsupported inspections.
- If you enable extended PAT for a dynamic PAT rule, then you cannot also use an address in the PAT pool as the PAT address in a separate static NAT with port translation rule. For example, if the PAT pool includes 10.1.1.1, then you cannot create a static NAT-with-port-translation rule using 10.1.1.1 as the PAT address.
- If you use a PAT pool and specify an interface for fallback, you cannot specify extended PAT.
- For VoIP deployments that use ICE or TURN, do not use extended PAT. ICE and TURN rely on the PAT binding to be the same for all destinations.

For round robin for a PAT pool

- If a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. Note: This “stickiness” does not survive a failover. If the ASA fails over, then subsequent connections from a host may not use the initial IP address.
- Round robin, especially when combined with extended PAT, can consume a large amount of memory. Because NAT pools are created for every mapped protocol/IP address/port range, round robin results in a large number of concurrent NAT pools, which use memory. Extended PAT results in an even larger number of concurrent NAT pools.

Configure Dynamic Network Object PAT

This section describes how to configure network object NAT for dynamic PAT.

Procedure

**Step 1** (Optional.) Create a host or range network object (object network command), or a network object group (object-group network command), for the mapped addresses.

- Instead of using an object, you can optionally configure an inline host address or specify the interface address.
- If you use an object, the object or group cannot contain a subnet; the object must define a host, or for a PAT pool, a range; the group (for a PAT pool) can include hosts and ranges.

**Step 2** Create or edit the network object for which you want to configure NAT.

```
object network obj_name
```

Example

```
hostname(config)# object network my-host-obj1
```

**Step 3** (Skip when editing an object that has the right address.) Define the real IPv4 or IPv6 addresses that you want to translate.

- **host** \{IPv4_address \ IPv6_address\}—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8:0DB8:800:200C:417A.
- **subnet** \{IPv4_address IPv4_mask \ IPv6_addressIPv6_prefix\}—The address of a network. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8:0:CD30::/60.
• **range** `start_address end_address`—A range of addresses. You can specify IPv4 or IPv6 ranges. Do not include masks or prefixes.

Example

```
hostname(config-network-object)# range 10.1.1.1 10.1.1.90
```

**Step 4**

Configure **dynamic PAT** for the object IP addresses. You can only define a single NAT rule for a given object.

```
nat [real_ifc,mapped_ifc] dynamic [mapped_inline_host_ip | mapped_obj | pat-pool mapped_obj [round-robin] [extended] [flat [include-reserve]] | interface [ipv6]] [interface [ipv6]] [dns]
```

Example

```
hostname(config-network-object)# nat (any,outside) dynamic interface
```

Where:

- **Interfaces**—(Required for transparent mode) Specify the real (`real_ifc`) and mapped (`mapped_ifc`) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword **any** for one or both of the interfaces, for example (any,outside).

- **Mapped IP address**—You can specify the mapped IP address as:
  - `mapped_inline_host_ip`—An inline host address.
  - `mapped_obj`—An existing network object that is defined as a host address.
  - `pat-pool`—An existing network object or group that contains multiple addresses.
  - `interface`—(Routed mode only.) The IP address of the mapped interface is used as the mapped address. If you specify `ipv6`, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the `mapped_ifc`. You must use this keyword when you want to use the interface IP address; you cannot enter it inline or as an object.

- **For a PAT pool, you can specify one or more of the following options:**
  - Round robin—The **round-robin** keyword enables round-robin address allocation for a PAT pool. Without round robin, by default all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on.
  - Extended PAT—The **extended** keyword enables extended PAT. Extended PAT uses 65535 ports per `service`, as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80.
  - Flat range—The **flat** keyword enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is **not** available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the `include-reserve` keyword.
**Dynamic PAT**

- **Interface PAT fallback**—(Optional.) The `interface` keyword enables interface PAT fallback when entered after a primary PAT address. After the primary PAT addresses are used up, then the IP address of the mapped interface is used. If you specify `ipv6`, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the `mapped_ifc`. (You cannot specify `interface` in transparent mode.)

- **DNS**—(Optional.) The `dns` keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 5-21 for more information.

### Examples

The following example configures dynamic PAT that hides the 192.168.2.0 network behind address 10.2.2.2:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic 10.2.2.2
```

The following example configures dynamic PAT that hides the 192.168.2.0 network behind the outside interface address:

```
hostname(config)# object network my-inside-net
hostname(config-network-object)# subnet 192.168.2.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic interface
```

The following example configures dynamic PAT with a PAT pool to translate the inside IPv6 network to an outside IPv4 network:

```
hostname(config)# object network IPv4_POOL
hostname(config-network-object)# range 203.0.113.1 203.0.113.254
hostname(config)# object network IPv6_INSIDE
hostname(config-network-object)# subnet 2001:DB8::/96
hostname(config-network-object)# nat (inside,outside) dynamic pat-pool IPv4_POOL
```

### Configure Dynamic Twice PAT

This section describes how to configure twice NAT for dynamic PAT.

**Procedure**

**Step 1** Create host or range network objects (`object network` command), or network object groups (`object-group network` command), for the source real addresses, the source mapped addresses, the destination real addresses, and the destination mapped addresses.

- If you want to translate all source traffic, you can skip adding an object for the source real addresses, and instead specify the `any` keyword in the `nat` command.

- If you want to use the interface address as the mapped address, you can skip adding an object for the source mapped addresses, and instead specify the `interface` keyword in the `nat` command.

- If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the `interface` keyword in the `nat` command.

If you use an object, the object or group cannot contain a subnet. The object must define a host, or for a PAT pool, a range. The group (for a PAT pool) can include hosts and ranges.
**Chapter 4  Network Address Translation (NAT)**

### Dynamic PAT

**Step 2** (Optional.) Create service objects for the destination real ports and the destination mapped ports.

For dynamic NAT, you can only perform port translation on the destination. A service object can contain both a source and destination port, but only the destination port is used in this case. If you specify the source port, it will be ignored.

**Step 3** Configure **dynamic PAT**.

```
nat (real_ifc,mapped_ifc) [line | {after-auto [line]}]
source dynamic {real-obj | any} [mapped_obj [interface [ipv6]] | {pat-pool mapped_obj [round-robin] [extended] [flat [include-reserve]] [interface [ipv6]] | interface [ipv6]}] {destination static {mapped_obj | interface [ipv6]} real_obj} [service mapped_dest_svc_obj real_dest_svc_obj] [dns] {unidirectional} [inactive] [description desc]
```

**Example**

```hostname(config)# nat (inside,outside) source dynamic MyInsNet interface destination static Server1 Server1
description Interface PAT for inside addresses when going to server 1```

Where:

- **Interfaces**—(Required for transparent mode) Specify the real (`real_ifc`) and mapped (`mapped_ifc`) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword `any` for one or both of the interfaces, for example `(any,outside)`.

- **Section and Line**—(Optional.) By default, the NAT rule is added to the end of section 1 of the NAT table (see `NAT Rule Order`, page 4-5). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the `after-auto` keyword. You can insert a rule anywhere in the applicable section using the `line` argument.

- **Source addresses:**
  - **Real**—Specify a network object, group, or the `any` keyword. Use the `any` keyword if you want to translate all traffic from the real interface to the mapped interface.
  - **Mapped**—Configure one of the following:
    - **Network object**—Specify a network object that contains a host address.
    - **pat-pool**—Specify the `pat-pool` keyword and a network object or group that contains multiple addresses.
    - **interface**—(Routed mode only.) Specify the `interface` keyword alone to only use interface PAT. If you specify `ipv6`, then the IPv6 address of the interface is used. When specified with a PAT pool or network object, the `interface` keyword enables interface PAT fallback. After the PAT IP addresses are used up, then the IP address of the mapped interface is used. For this option, you must configure a specific interface for the `mapped_ifc`.

For a PAT pool, you can specify one or more of the following options:

- **Round robin**—The `round-robin` keyword enables round-robin allocation for a PAT pool. Without round robin, by default all ports for a PAT address will be allocated before the next PAT address is used. The round-robin method assigns an address/port from each PAT address in the pool before returning to use the first address again, and then the second address, and so on.

- **Extended PAT**—The `extended` keyword enables extended PAT. Extended PAT uses 65535 ports per `service`, as opposed to per IP address, by including the destination address and port in the translation information. Normally, the destination port and address are not considered when
creating PAT translations, so you are limited to 65535 ports per PAT address. For example, with extended PAT, you can create a translation of 10.1.1.1:1027 when going to 192.168.1.7:23 as well as a translation of 10.1.1.1:1027 when going to 192.168.1.7:80.

-- Flat range—The flat keyword enables use of the entire 1024 to 65535 port range when allocating ports. When choosing the mapped port number for a translation, the ASA uses the real source port number if it is available. However, without this option, if the real port is not available, by default the mapped ports are chosen from the same range of ports as the real port number: 1 to 511, 512 to 1023, and 1024 to 65535. To avoid running out of ports at the low ranges, configure this setting. To use the entire range of 1 to 65535, also specify the include-reserve keyword.

- Destination addresses (Optional):
  - Mapped—Specify a network object or group, or for static interface NAT with port translation only (routed mode), specify the interface keyword. If you specify ipv6, then the IPv6 address of the interface is used. If you specify interface, be sure to also configure the service keyword. For this option, you must configure a specific interface for the real_ifc. See Static Interface NAT with Port Translation, page 4-29 for more information.
  - Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses.

- Destination port—(Optional.) Specify the service keyword along with the mapped and real service objects. For identity port translation, simply use the same service object for both the real and mapped ports.

- DNS—(Optional; for a source-only rule.) The dns keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). You cannot configure the dns keyword if you configure a destination address. See DNS and NAT, page 5-21 for more information.

- Unidirectional—(Optional.) Specify unidirectional so the destination addresses cannot initiate traffic to the source addresses.

- Inactive—(Optional.) To make this rule inactive without having to remove the command, use the inactive keyword. To reactivate it, reenter the whole command without the inactive keyword.

- Description—Optional.) Provide a description up to 200 characters using the description keyword.

Examples

The following example configures interface PAT for inside network 192.168.1.0/24 when accessing outside Telnet server 209.165.201.23, and Dynamic PAT using a PAT pool when accessing any server on the 203.0.113.0/24 network.

```
hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 192.168.1.0 255.255.255.0

hostname(config)# object network PAT_POOL
hostname(config-network-object)# range 209.165.200.225 209.165.200.254

hostname(config)# object network TELNET_SVR
hostname(config-network-object)# host 209.165.201.23

hostname(config)# object service TELNET
hostname(config-service-object)# service tcp destination eq 23

hostname(config)# object network SERVERS
hostname(config-network-object)# subnet 203.0.113.0 255.255.255.0
```
hostname(config)# nat (inside,outside) source dynamic INSIDE_NW interface
destination static TELNET_SVR TELNET_SVR service TELNET TELNET
hostname(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool PAT_POOL
destination static SERVERS SERVERS

The following example configures interface PAT for inside network 192.168.1.0/24 when accessing
outside IPv6 Telnet server 2001:DB8::23, and Dynamic PAT using a PAT pool when accessing any server
on the 2001:DB8:AAAA::/96 network.

hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 192.168.1.0 255.255.255.0

hostname(config)# object network PAT_POOL
hostname(config-network-object)# range 2001:DB8:AAAA::1 2001:DB8:AAAA::200

hostname(config)# object network TELNET_SVR
hostname(config-network-object)# host 2001:DB8::23

hostname(config)# object service TELNET
hostname(config-service-object)# service tcp destination eq 23

hostname(config)# object network SERVERS
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96

hostname(config)# nat (inside,outside) source dynamic INSIDE_NW interface ipv6
destination static TELNET_SVR TELNET_SVR service TELNET TELNET
hostname(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool PAT_POOL
destination static SERVERS SERVERS

Configure Per-Session PAT or Multi-Session PAT

By default, all TCP PAT traffic and all UDP DNS traffic uses per-session PAT. To use multi-session PAT
for traffic, you can configure per-session PAT rules: a permit rule uses per-session PAT, and a deny rule
uses multi-session PAT.

Per-session PAT improves the scalability of PAT and, for clustering, allows each member unit to own
PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit.
At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This
reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state.
Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds.

For “hit-and-run” traffic, such as HTTP or HTTPS, per-session PAT can dramatically increase the
connection rate supported by one address. Without per-session PAT, the maximum connection rate for
one address for an IP protocol is approximately 2000 per second. With per-session PAT, the connection
rate for one address for an IP protocol is 65535/average-lifetime.

For traffic that can benefit from multi-session PAT, such as H.323, SIP, or Skinny, you can disable
per-session PAT by creating a per-session deny rule. These rules are available starting with version
9.0(1).

Before You Begin

By default, the following rules are installed:
xlate per-session permit tcp any4 any4
xlate per-session permit tcp any4 any6
xlate per-session permit tcp any6 any4
xlate per-session permit tcp any6 any6
xlate per-session permit udp any4 any4 eq domain
You cannot remove these rules, and they always exist after any manually-created rules. Because rules are evaluated in order, you can override the default rules. For example, to completely negate these rules, you could add the following:

```
xlate per-session deny tcp any4 any4
xlate per-session deny tcp any4 any6
xlate per-session deny tcp any6 any4
xlate per-session deny tcp any6 any6
xlate per-session deny udp any4 any4 eq domain
xlate per-session deny udp any4 any6 eq domain
xlate per-session deny udp any6 any4 eq domain
xlate per-session deny udp any6 any6 eq domain
```

**Procedure**

**Step 1** Create a permit or deny per-session PAT rule. This rule is placed above the default rules, but below any other manually-created rules. Be sure to create your rules in the order you want them applied.

```
xlate per-session {permit | deny} {tcp | udp} source_ip [operator src_port] destination_ip [operator dest_port]
```

**Example**

```
hostname(config)# xlate per-session deny tcp any4 209.165.201.3 eq 1720
```

For the source and destination IP addresses, you can configure the following:

- **host ip_address**—Specifies an IPv4 or IPv6 host address.
- **ip_address mask**—Specifies an IPv4 network address and subnet mask.
- **ipv6-address/prefix-length**—Specifies an IPv6 network address and prefix.
- **any4** and **any6**—**any4** specifies only IPv4 traffic; and **any6** specifies any6 traffic.

The **operator** matches the port numbers used by the source or destination. The default is all ports. The permitted operators are:

- **lt**—less than
- **gt**—greater than
- **eq**—equal to
- **neq**—not equal to
- **range**—an inclusive range of values. When you use this operator, specify two port numbers, for example, **range 100 200**.

**Examples**

The following example creates a deny rule for H.323 traffic, so that it uses multi-session PAT:

```
hostname(config)# xlate per-session deny tcp any4 209.165.201.7 eq 1720
hostname(config)# xlate per-session deny udp any4 209.165.201.7 range 1718 1719
```
Static NAT

The following topics explain static NAT and how to implement it.

- About Static NAT, page 4-27
- Configure Static Network Object NAT or Static NAT-with-Port-Translation, page 4-32
- Configure Static Twice NAT or Static NAT-with-Port-Translation, page 4-34

About Static NAT

Static NAT creates a fixed translation of a real address to a mapped address. Because the mapped address is the same for each consecutive connection, static NAT allows bidirectional connection initiation, both to and from the host (if an access rule exists that allows it). With dynamic NAT and PAT, on the other hand, each host uses a different address or port for each subsequent translation, so bidirectional initiation is not supported.

The following figure shows a typical static NAT scenario. The translation is always active so both real and remote hosts can initiate connections.

Figure 4-5 Static NAT

![Static NAT Diagram]

Note

You can disable bidirectionality if desired.

Static NAT with Port Translation

Static NAT with port translation lets you specify a real and mapped protocol (TCP or UDP) and port.

- About Static NAT with Port Address Translation, page 4-27
- Static NAT with Identity Port Translation, page 4-28
- Static NAT with Port Translation for Non-Standard Ports, page 4-29
- Static Interface NAT with Port Translation, page 4-29

About Static NAT with Port Address Translation

When you specify the port with static NAT, you can choose to map the port and/or the IP address to the same value or to a different value.
The following figure shows a typical static NAT with port translation scenario showing both a port that is mapped to itself and a port that is mapped to a different value; the IP address is mapped to a different value in both cases. The translation is always active so both translated and remote hosts can initiate connections.

**Figure 4-6 Typical Static NAT with Port Translation Scenario**

The following static NAT with port translation example provides a single address for remote users to access FTP, HTTP, and SMTP. These servers are actually different devices on the real network, but for each server, you can specify static NAT with port translation rules that use the same mapped IP address, but different ports. For details on how to configure this example, see Single Address for FTP, HTTP, and SMTP (Static NAT-with-Port-Translation), page 5-5.
Static NAT with Port Translation for Non-Standard Ports

You can also use static NAT with port translation to translate a well-known port to a non-standard port or vice versa. For example, if inside web servers use port 8080, you can allow outside users to connect to port 80, and then undo translation to the original port 8080. Similarly, to provide extra security, you can tell web users to connect to non-standard port 6785, and then undo translation to port 80.

Static Interface NAT with Port Translation

You can configure static NAT to map a real address to an interface address/port combination. For example, if you want to redirect Telnet access for the ASA outside interface to an inside host, then you can map the inside host IP address/port 23 to the ASA interface address/port 23. (Note that although Telnet to the ASA is not allowed to the lowest security interface, static NAT with interface port translation redirects the Telnet session instead of denying it).

One-to-Many Static NAT

Typically, you configure static NAT with a one-to-one mapping. However, in some cases, you might want to configure a single real address to several mapped addresses (one-to-many). When you configure one-to-many static NAT, when the real host initiates traffic, it always uses the first mapped address. However, for traffic initiated to the host, you can initiate traffic to any of the mapped addresses, and they will be untranslated to the single real address.
The following figure shows a typical one-to-many static NAT scenario. Because initiation by the real host always uses the first mapped address, the translation of real host IP/1st mapped IP is technically the only bidirectional translation.

**Figure 4-8 One-to-Many Static NAT**

For example, you have a load balancer at 10.1.2.27. Depending on the URL requested, it redirects traffic to the correct web server. For details on how to configure this example, see *Inside Load Balancer with Multiple Mapped Addresses (Static NAT, One-to-Many)*, page 5-4.

**Figure 4-9 One-to-Many Static NAT Example**
Other Mapping Scenarios (Not Recommended)

The ASA has the flexibility to allow any kind of static mapping scenario: one-to-one, one-to-many, but also few-to-many, many-to-few, and many-to-one mappings. We recommend using only one-to-one or one-to-many mappings. These other mapping options might result in unintended consequences.

Functionally, few-to-many is the same as one-to-many; but because the configuration is more complicated and the actual mappings may not be obvious at a glance, we recommend creating a one-to-many configuration for each real address that requires it. For example, for a few-to-many scenario, the few real addresses are mapped to the many mapped addresses in order (A to 1, B to 2, C to 3). When all real addresses are mapped, the next mapped address is mapped to the first real address, and so on until all mapped addresses are mapped (A to 4, B to 5, C to 6). This results in multiple mapped addresses for each real address. Just like a one-to-many configuration, only the first mappings are bidirectional; subsequent mappings allow traffic to be initiated to the real host, but all traffic from the real host uses only the first mapped address for the source.

The following figure shows a typical few-to-many static NAT scenario.

![Figure 4-10 Few-to-Many Static NAT](image)

For a many-to-few or many-to-one configuration, where you have more real addresses than mapped addresses, you run out of mapped addresses before you run out of real addresses. Only the mappings between the lowest real IP addresses and the mapped pool result in bidirectional initiation. The remaining higher real addresses can initiate traffic, but traffic cannot be initiated to them (returning traffic for a connection is directed to the correct real address because of the unique 5-tuple (source IP, destination IP, source port, destination port, protocol) for the connection).

**Note**

Many-to-few or many-to-one NAT is not PAT. If two real hosts use the same source port number and go to the same outside server and the same TCP destination port, and both hosts are translated to the same IP address, then both connections will be reset because of an address conflict (the 5-tuple is not unique).
The following figure shows a typical many-to-few static NAT scenario.

**Figure 4-11 Many-to-Few Static NAT**

Instead of using a static rule this way, we suggest that you create a one-to-one rule for the traffic that needs bidirectional initiation, and then create a dynamic rule for the rest of your addresses.

### Configure Static Network Object NAT or Static NAT-with-Port-Translation

This section describes how to configure a static NAT rule using network object NAT.

**Procedure**

**Step 1**  
(Optional.) Create a network object (`object network` command), or a network object group (`object-group network` command), for the mapped addresses.

- Instead of using an object, you can configure an inline address or specify the interface address (for static NAT-with-port-translation).
- If you use an object, the object or group can contain a host, range, or subnet.

**Step 2**  
Create or edit the network object for which you want to configure NAT.

```
object network obj_name
```

**Example**

```
hostname(config)# object network my-host-obj1
```

**Step 3**  
(Skip when editing an object that has the right address.) Define the real IPv4 or IPv6 addresses that you want to translate.

- **host** `{IPv4_address \ IPv6_address}`—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8:0:800:200C:417A.
- **subnet** `{IPv4_address IPv4_mask \ IPv6_address/IPv6_prefix}`—The address of a network. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8:0:CD30::/60.
- **range** `start_address end_address`—A range of addresses. You can specify IPv4 or IPv6 ranges. Do not include masks or prefixes.

**Example**

```
hostname(config-network-object)# subnet 10.2.1.0 255.255.255.0
```

**Step 4**  
Configure **static NAT** for the object IP addresses. You can only define a single NAT rule for a given object.
Chapter 4      Network Address Translation (NAT)

Static NAT

nat [(real_ifc,mapped_ifc)] static {mapped_inline_ip | mapped_obj | interface [ipv6]} [net-to-net] {dns | service (tcp | udp) real_port mapped_port} [no-proxy-arp]

Example
hostname(config-network-object)#
nat (inside,outside) static MAPPED_IPS service tcp 80 8080

Where:

- Interfaces—(Required for transparent mode) Specify the real (real_ifc) and mapped (mapped_ifc) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword any for one or both of the interfaces, for example (any,outside).

- Mapped IP address—You can specify the mapped IP address as one of the following. Typically, you configure the same number of mapped addresses as real addresses for a one-to-one mapping. You can, however, have a mismatched number of addresses. See Static NAT, page 4-27.
  - mapped_inline_host_ip—An inline IP address. The netmask, prefix, or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 172.20.1.1 as the mapped address, then the mapped range will include 172.20.1.1 through 172.20.1.6.
  - mapped_obj—An existing network object or group.
  - interface—(Static NAT-with-port-translation only; routed mode only.) The IP address of the mapped interface is used as the mapped address. If you specify ipv6, then the IPv6 address of the interface is used. For this option, you must configure a specific interface for the mapped_ifc. You must use this keyword when you want to use the interface IP address; you cannot enter it inline or as an object. Be sure to also configure the service keyword.

- Net-to-net—(Optional.) For NAT 46, specify net-to-net to translate the first IPv4 address to the first IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method is used. For a one-to-one translation, you must use this keyword.

- DNS—(Optional.) The dns keyword translates DNS replies. Be sure DNS inspection is enabled (it is enabled by default). See DNS and NAT, page 5-21 for more information.

- Port translation—(Static NAT-with-port-translation only.) Specify service with either tcp or udp and the real and mapped ports. You can enter either a port number or a well-known port name (such as ftp).

- No Proxy ARP—(Optional.) Specify no-proxy-arp to disable proxy ARP for incoming packets to the mapped IP addresses. For information on the conditions which might require the disabling of proxy ARP, see Mapped Addresses and Routing, page 5-12.

Examples
The following example configures static NAT for the real host 10.1.1.1 on the inside to 10.2.2.2 on the outside with DNS rewrite enabled.

hostname(config)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside,outside) static 10.2.2.2 dns

The following example configures static NAT for the real host 10.1.1.1 on the inside to 10.2.2.2 on the outside using a mapped object.
hostname(config)# object network my-mapped-obj
hostname(config-network-object)# host 10.2.2.2

hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside, outside) static my-mapped-obj

The following example configures static NAT-with-port-translation for 10.1.1.1 at TCP port 21 to the outside interface at port 2121.

hostname(config)# object network my-ftp-server
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside, outside) static interface service tcp 21 2121

The following example maps an inside IPv4 network to an outside IPv6 network.

hostname(config)# object network inside_v4_v6
hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0
hostname(config-network-object)# nat (inside, outside) static 2001:DB8::/96

The following example maps an inside IPv6 network to an outside IPv6 network.

hostname(config)# object network inside_v6
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96
hostname(config-network-object)# nat (inside, outside) static 2001:DB8:BBBB::/96

**Configure Static Twice NAT or Static NAT-with-Port-Translation**

This section describes how to configure a static NAT rule using twice NAT.

**Procedure**

**Step 1** Create host or range network objects (`object network` command), or network object groups (`object-group network` command), for the source real addresses, the source mapped addresses, the destination real addresses, and the destination mapped addresses.

- If you want to configure source static interface NAT with port translation only, you can skip adding an object for the source mapped addresses, and instead specify the `interface` keyword in the `nat` command.
- If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the `interface` keyword in the `nat` command.

If you do create objects, consider the following guidelines:

- The mapped object or group can contain a host, range, or subnet.
- The static mapping is typically one-to-one, so the real addresses have the same quantity as the mapped addresses. You can, however, have different quantities if desired. For more information, see Static NAT, page 4-27.

**Step 2** (Optional.) Create service objects for the:

- Source or Destination real ports
- Source or Destination mapped ports
A service object can contain both a source and destination port; however, you should specify either the source or the destination port for both service objects. You should only specify both the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. For example, if you want to translate the port for the source host, then configure the source service.

**Step 3** Configure static NAT.

```
nat (real_ifc,mapped_ifc) [line | {after-object [line]}]
source static real Obj [mapped_obj | interface [ipv6]]
{destination static {mapped_obj | interface [ipv6]} real_obj}
{service real_src_mapped_dest_svc_obj mapped_src_real_dest_svc_obj}
{net-to-net} {dns} {unidirectional | no-proxy-arp | inactive} {description desc}
```

Example

```
hostname(config)# nat (inside,dmz) source static MyInsNet MyInsNet_mapped
destination static Server1 Server1 service REAL_SRC_SVC MAPPED_SRC_SVC
```

Where:

- **Interfaces**—(Required for transparent mode) Specify the real (real_ifc) and mapped (mapped_ifc) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword any for one or both of the interfaces, for example (any,outside).

- **Section and Line**—(Optional.) By default, the NAT rule is added to the end of section 1 of the NAT table (see NAT Rule Order, page 4-5). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the after-auto keyword. You can insert a rule anywhere in the applicable section using the line argument.

- **Source addresses:**
  - Real—Specify a network object or group. Do not use the any keyword, which would be used for identity NAT.
  - Mapped—Specify a different network object or group. For static interface NAT with port translation only, you can specify the interface keyword (routed mode only). If you specify ipv6, then the IPv6 address of the interface is used. If you specify interface, be sure to also configure the service keyword (in this case, the service objects should include only the source port). For this option, you must configure a specific interface for the mapped_ifc. See Static Interface NAT with Port Translation, page 4-29 for more information.

- **Destination addresses** (Optional):
  - Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the interface keyword. If you specify ipv6, then the IPv6 address of the interface is used. If you specify interface, be sure to also configure the service keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the real_ifc.
  - Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses.

- **Ports**—(Optional.) Specify the service keyword along with the real and mapped service objects. For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is service real_obj mapped_obj. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is service mapped_obj real_obj. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source
port/mapped destination port; the second service object contains the mapped source port/real
destination port. For identity port translation, simply use the same service object for both the real
and mapped ports (source and/or destination ports, depending on your configuration).

- Net-to-net—(Optional.) For NAT 46, specify **net-to-net** to translate the first IPv4 address to the first
IPv6 address, the second to the second, and so on. Without this option, the IPv4-embedded method
is used. For a one-to-one translation, you must use this keyword.

- DNS—(Optional; for a source-only rule.) The **dns** keyword translates DNS replies. Be sure DNS
inspection is enabled (it is enabled by default). You cannot configure the **dns** keyword if you
configure a destination address. See **DNS and NAT**, page 5-21 for more information.

- Unidirectional—(Optional.) Specify **unidirectional** so the destination addresses cannot initiate
traffic to the source addresses.

- No Proxy ARP—(Optional.) Specify **no-proxy-arp** to disable proxy ARP for incoming packets to
the mapped IP addresses. See **Mapped Addresses and Routing**, page 5-12 for more information.

- Inactive—(Optional.) To make this rule inactive without having to remove the command, use the
**inactive** keyword. To reactivate it, reenter the whole command without the **inactive** keyword.

- Description—Optional.) Provide a description up to 200 characters using the **description** keyword.

### Examples

The following example shows the use of static interface NAT with port translation. Hosts on the outside
access an FTP server on the inside by connecting to the outside interface IP address with destination port
65000 through 65004. The traffic is untranslated to the internal FTP server at 192.168.10.100:6500
through 65004. Note that you specify the source port range in the service object (and not the destination
port) because you want to translate the source address and port as identified in the command; the
destination port is “any.” Because static NAT is bidirectional, “source” and “destination” refers primarily
to the command keywords; the actual source and destination address and port in a packet depends on
which host sent the packet. In this example, connections are originated from outside to inside, so the
“source” address and port of the FTP server is actually the destination address and port in the originating
packet.

```plaintext
hostname(config)# object service FTP_PASV_PORT_RANGE
hostname(config-service-object)# service tcp source range 65000 65004

hostname(config)# object network HOST_FTP_SERVER
hostname(config-network-object)# host 192.168.10.100

hostname(config)# nat (inside,outside) source static HOST_FTP_SERVER interface
service FTP_PASV_PORT_RANGE FTP_PASV_PORT_RANGE
```

The following example shows a static translation of one IPv6 network to another IPv6 when accessing
an IPv6 network, and the dynamic PAT translation to an IPv4 PAT pool when accessing the IPv4 network:

```plaintext
hostname(config)# object network INSIDE_NW
hostname(config-network-object)# subnet 2001:DB8:AAAA::/96

hostname(config)# object network MAPPED_IPv6_NW
hostname(config-network-object)# subnet 2001:DB8:BBBB::/96

hostname(config)# object network OUTSIDE_IPv6_NW
hostname(config-network-object)# subnet 2001:DB8:CCCC::/96

hostname(config)# object network OUTSIDE_IPv4_NW
hostname(config-network-object)# subnet 10.1.1.0 255.255.255.0
```
Identity NAT

You might have a NAT configuration in which you need to translate an IP address to itself. For example, if you create a broad rule that applies NAT to every network, but want to exclude one network from NAT, you can create a static NAT rule to translate an address to itself. Identity NAT is necessary for remote access VPN, where you need to exempt the client traffic from NAT.

The following figure shows a typical identity NAT scenario.

![Identity NAT Diagram](image)

The following topics explain how to configure identity NAT.

- Configure Identity Network Object NAT, page 4-37
- Configure Identity Twice NAT, page 4-39

Configure Identity Network Object NAT

This section describes how to configure an identity NAT rule using network object NAT.

**Procedure**

**Step 1** (Optional.) Create a network object (`object network` command), or a network object group (`object-group network` command), for the mapped addresses.

- Instead of using an object, you can configure an inline address.
- If you use an object, the object must match the real addresses you want to translate.

**Step 2** Create or edit the network object for which you want to configure NAT. The object must be a different one than what you use for the mapped addresses, even though the contents must be the same in each object.

```
object network obj_name
```

**Example**

```bash
hostname(config)# object network MAPPED_IPV4_POOL
hostname(config-network-object)# range 10.1.2.1 10.1.2.254

hostname(config)# nat (inside,outside) source static INSIDE_NW MAPPED_IPV6_NW
destination static OUTSIDE_IPV6_NW OUTSIDE_IPV6_NW

hostname(config)# nat (inside,outside) source dynamic INSIDE_NW pat-pool MAPPED_IPV4_POOL
destination static OUTSIDE_IPV4_NW OUTSIDE_IPV4_NW
```
hostname(config)# object network my-host-obj1

**Step 3**  (Skip when editing an object that has the right address.) Define the real IPv4 or IPv6 addresses that you want to translate.

- **host** \{IPv4\_address | IPv6\_address\}—The IPv4 or IPv6 address of a single host. For example, 10.1.1.1 or 2001:DB8::0DB8:800:200C:417A.

- **subnet** \{IPv4\_address IPv4\_mask | IPv6\_address IPv6\_prefix\}—The address of a network. For IPv4 subnets, include the mask after a space, for example, 10.0.0.0 255.0.0.0. For IPv6, include the address and prefix as a single unit (no spaces), such as 2001:DB8::CD30::/60.

- **range** start\_address end\_address—A range of addresses. You can specify IPv4 or IPv6 ranges. Do not include masks or prefixes.

Example

hostname(config-network-object)# subnet 10.2.1.0 255.255.255.0

**Step 4**  Configure **identity NAT** for the object IP addresses. You can only define a single NAT rule for a given object.

```
nat \{(real\_ifc, mapped\_ifc)\} static \{mapped\_inline\_ip | mapped\_obj\}
\{no-proxy-arp\} \{route-lookup\}
```

Example

hostname(config-network-object)# nat (inside, outside) static MAPPED_IPS

Where:

- **Interfaces**—(Required for transparent mode) Specify the real (real\_ifc) and mapped (mapped\_ifc) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword any for one or both of the interfaces, for example (any,outside).

- **Mapped IP addresses**—Be sure to configure the same IP address for both the mapped and real address. Use one of the following:
  - **mapped\_inline\_host\_ip**—An inline IP address. The netmask, prefix, or range for the mapped network is the same as that of the real network. For example, if the real network is a host, then this address will be a host address. In the case of a range, then the mapped addresses include the same number of addresses as the real range. For example, if the real address is defined as a range from 10.1.1.1 through 10.1.1.6, and you specify 10.1.1.1 as the mapped address, then the mapped range will include 10.1.1.1 through 10.1.1.6.
  - **mapped\_obj**—A network object or group that includes the same addresses as the real object.

- **No Proxy ARP**—(Optional.) Specify no-proxy-arp to disable proxy ARP for incoming packets to the mapped IP addresses. For information on the conditions which might require the disabling of proxy ARP, see Mapped Addresses and Routing, page 5-12.

- **Route lookup**—(Routed mode only; interfaces specified.) Specify route-lookup to determine the egress interface using a route lookup instead of using the interface specified in the NAT command. See Determining the Egress Interface, page 5-14 for more information.


Example

The following example maps a host address to itself using an inline mapped address:

```
hostname(config)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside, outside) static 10.1.1.1
```

The following example maps a host address to itself using a network object:

```
hostname(config)# object network my-host-obj1-identity
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# object network my-host-obj1
hostname(config-network-object)# host 10.1.1.1
hostname(config-network-object)# nat (inside, outside) static my-host-obj1-identity
```

Configure Identity Twice NAT

This section describes how to configure an identity NAT rule using twice NAT.

Procedure

Step 1

Create host or range network objects (object network command), or network object groups (object-group network command), for the source real addresses (you will typically use the same object for the source mapped addresses), the destination real addresses, and the destination mapped addresses.

- If you want to perform identity NAT for all addresses, you can skip creating an object for the source real addresses and instead use the keywords any any in the nat command.
- If you want to configure destination static interface NAT with port translation only, you can skip adding an object for the destination mapped addresses, and instead specify the interface keyword in the nat command.

If you do create objects, consider the following guidelines:

- The mapped object or group can contain a host, range, or subnet.
- The real and mapped source objects must match. You can use the same object for both, or you can create separate objects that contain the same IP addresses.

Step 2

(Optional.) Create service objects for the:

- Source or Destination real ports
- Source or Destination mapped ports

A service object can contain both a source and destination port; however, you should specify either the source or the destination port for both service objects. You should only specify both the source and destination ports if your application uses a fixed source port (such as some DNS servers); but fixed source ports are rare. For example, if you want to translate the port for the source host, then configure the source service.

Step 3

Configure identity NAT.

```
nat [(real_ifc, mapped_ifc)] [line | {after-object [line]}]
source static (nw_obj nw_obj | any any)
[destination static (mapped_obj | interface [ipv6]) real_obj]
[service real_src_mapped_dest_svc_obj mapped_src_real_dest_svc_obj]
[no-proxy-arp] [route-lookup] [inactive] [description desc]
```
Example

```
hostname(config)# nat (inside, outside) source static MyInsNet MyInsNet
destination static Server1 Server1
```

Where:

- **Interfaces**—(Required for transparent mode.) Specify the real \( real\_ifc \) and mapped \( mapped\_ifc \) interfaces. Be sure to include the parentheses. In routed mode, if you do not specify the real and mapped interfaces, all interfaces are used. You can also specify the keyword \texttt{any} for one or both of the interfaces, for example \texttt{(any, outside)}.

- **Section and Line**—(Optional.) By default, the NAT rule is added to the end of section 1 of the NAT table (see \texttt{NAT Rule Order, page 4-5}). If you want to add the rule into section 3 instead (after the network object NAT rules), then use the \texttt{after-auto} keyword. You can insert a rule anywhere in the applicable section using the \texttt{line} argument.

- **Source addresses**—Specify a network object, group, or the \texttt{any} keyword for both the real and mapped addresses.

- **Destination addresses**—(Optional):
  - Mapped—Specify a network object or group, or for static interface NAT with port translation only, specify the \texttt{interface} keyword (routed mode only). If you specify \texttt{ipv6}, then the IPv6 address of the interface is used. If you specify \texttt{interface}, be sure to also configure the \texttt{service} keyword (in this case, the service objects should include only the destination port). For this option, you must configure a specific interface for the \texttt{real\_ifc}.
  - Real—Specify a network object or group. For identity NAT, simply use the same object or group for both the real and mapped addresses.

- **Ports**—(Optional.) Specify the \texttt{service} keyword along with the real and mapped service objects. For source port translation, the objects must specify the source service. The order of the service objects in the command for source port translation is \texttt{service real_obj mapped_obj}. For destination port translation, the objects must specify the destination service. The order of the service objects for destination port translation is \texttt{service mapped_obj real_obj}. In the rare case where you specify both the source and destination ports in the object, the first service object contains the real source port/mapped destination port; the second service object contains the mapped source port/real destination port. For identity port translation, simply use the same service object for both the real and mapped ports (source and/or destination ports, depending on your configuration).

- **No Proxy ARP**—(Optional.) Specify \texttt{no-proxy-arp} to disable proxy ARP for incoming packets to the mapped IP addresses. See \texttt{Mapped Addresses and Routing, page 5-12} for more information.

- **Route lookup**—(Optional; routed mode only; interfaces specified.) Specify \texttt{route-lookup} to determine the egress interface using a route lookup instead of using the interface specified in the NAT command. See \texttt{Determining the Egress Interface, page 5-14} for more information.

- **Inactive**—(Optional.) To make this rule inactive without having to remove the command, use the \texttt{inactive} keyword. To reactivate it, reenter the whole command without the \texttt{inactive} keyword.

- **Description**—(Optional.) Provide a description up to 200 characters using the \texttt{description} keyword.

---

**Monitoring NAT**

To monitor object NAT, use the following commands:

- `show nat`
Shows NAT statistics, including hits for each NAT rule.

- **show nat pool**
  Shows NAT pool statistics, including the addresses and ports allocated, and how many times they were allocated.

- **show running-config nat**
  Shows the NAT configuration. You cannot see object NAT rules using `show running-config object`. When you use show running-config without modifiers, objects that include NAT rules are shown twice, first with the basic address configuration, then later in the configuration, the object with the NAT rule. The complete object, with the address and NAT rule, is not shown as a unit.

- **show xlate**
  Shows current NAT session information.

### History for NAT

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Object NAT</td>
<td>8.3(1)</td>
<td>Configures NAT for a network object IP address(es). We introduced or modified the following commands: <strong>nat (object network configuration mode)</strong>, <strong>show nat</strong>, <strong>show xlate</strong>, <strong>show nat pool</strong>.</td>
</tr>
<tr>
<td>Twice NAT</td>
<td>8.3(1)</td>
<td>Twice NAT lets you identify both the source and destination address in a single rule. We modified or introduced the following commands: <strong>nat</strong>, <strong>show nat</strong>, <strong>show xlate</strong>, <strong>show nat pool</strong>.</td>
</tr>
</tbody>
</table>
### History for NAT

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity NAT configurable proxy ARP and route lookup</td>
<td>8.4(2)/8.5(1)</td>
<td>In earlier releases for identity NAT, proxy ARP was disabled, and a route lookup was always used to determine the egress interface. You could not configure these settings. In 8.4(2) and later, the default behavior for identity NAT was changed to match the behavior of other static NAT configurations: proxy ARP is enabled, and the NAT configuration determines the egress interface (if specified) by default. You can leave these settings as is, or you can enable or disable them discretely. Note that you can now also disable proxy ARP for regular static NAT. For pre-8.3 configurations, the migration of NAT exempt rules (the <code>nat 0 access-list</code> command) to 8.4(2) and later now includes the following keywords to disable proxy ARP and to use a route lookup: <code>no-proxy-arp</code> and <code>route-lookup</code>. The <code>unidirectional</code> keyword that was used for migrating to 8.3(2) and 8.4(1) is no longer used for migration. When upgrading to 8.4(2) from 8.3(1), 8.3(2), and 8.4(1), all identity NAT configurations will now include the <code>no-proxy-arp</code> and <code>route-lookup</code> keywords, to maintain existing functionality. The <code>unidirectional</code> keyword is removed. We modified the following command: <code>nat static [no-proxy-arp] [route-lookup]</code>.</td>
</tr>
<tr>
<td>PAT pool and round robin address assignment</td>
<td>8.4(2)/8.5(1)</td>
<td>You can now specify a pool of PAT addresses instead of a single address. You can also optionally enable round-robin assignment of PAT addresses instead of first using all ports on a PAT address before using the next address in the pool. These features help prevent a large number of connections from a single PAT address from appearing to be part of a DoS attack and makes configuration of large numbers of PAT addresses easy. We modified the following commands: <code>nat dynamic [pat-pool mapped_object [round-robin]]</code> and <code>nat source dynamic [pat-pool mapped_object [round-robin]]</code>.</td>
</tr>
<tr>
<td>Round robin PAT pool allocation uses the same IP address for existing hosts</td>
<td>8.4(3)</td>
<td>When using a PAT pool with round robin allocation, if a host has an existing connection, then subsequent connections from that host will use the same PAT IP address if ports are available. We did not modify any commands. <em>This feature is not available in 8.5(1) or 8.6(1).</em></td>
</tr>
</tbody>
</table>
### Feature Name

| Flat range of PAT ports for a PAT pool | 8.4(3) | If available, the real source port number is used for the mapped port. However, if the real port is not available, by default the mapped ports are chosen from the same range of ports as the real port number: 0 to 511, 512 to 1023, and 1024 to 65535. Therefore, ports below 1024 have only a small PAT pool.
If you have a lot of traffic that uses the lower port ranges, when using a PAT pool, you can now specify a flat range of ports to be used instead of the three unequal-sized tiers: either 1024 to 65535, or 1 to 65535.
We modified the following commands: `nat dynamic [pat-pool mapped_object [flat [include-reserve]]]` and `nat source dynamic [pat-pool mapped_object [flat [include-reserve]]].` *This feature is not available in 8.5(1) or 8.6(1).* |

| Extended PAT for a PAT pool | 8.4(3) | Each PAT IP address allows up to 65535 ports. If 65535 ports do not provide enough translations, you can now enable extended PAT for a PAT pool. Extended PAT uses 65535 ports per service, as opposed to per IP address, by including the destination address and port in the translation information.
We modified the following command: `nat dynamic [pat-pool mapped_object [extended]]` and `nat source dynamic [pat-pool mapped_object [extended]].` *This feature is not available in 8.5(1) or 8.6(1).* |
### History for NAT

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Description</th>
</tr>
</thead>
</table>
| Automatic NAT rules to translate a VPN peer’s local IP address back to the peer’s real IP address | 8.4(3)            | In rare situations, you might want to use a VPN peer’s real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer’s real public IP address if, for example, your inside servers and network security is based on the peer’s real IP address. You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected. You can view the rules using the `show nat` command. Because of routing issues, we do not recommend using this feature unless you know you need it; contact Cisco TAC to confirm feature compatibility with your network. See the following limitations:  
  - Only supports Cisco IPsec and AnyConnect Client.  
  - Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied.  
  - Does not support load-balancing (because of routing issues).  
  - Does not support roaming (public IP changing).  
We introduced the following command: `nat-assigned-to-public-ip interface` (tunnel-group general-attributes configuration mode). |
| NAT support for IPv6                                                        | 9.0(1)            | NAT now supports IPv6 traffic, as well as translating between IPv4 and IPv6. Translating between IPv4 and IPv6 is not supported in transparent mode. We modified the following commands: `nat` (global and object network configuration modes), `show nat`, `show nat pool`, `show xlate`. |
| NAT support for reverse DNS lookups                                         | 9.0(1)            | NAT now supports translation of the DNS PTR record for reverse DNS lookups when using IPv4 NAT, IPv6 NAT, and NAT64 with DNS inspection enabled for the NAT rule. |
### Per-session PAT

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per-session PAT</td>
<td>9.0(1)</td>
<td>The per-session PAT feature improves the scalability of PAT and, for clustering, allows each member unit to own PAT connections; multi-session PAT connections have to be forwarded to and owned by the master unit. At the end of a per-session PAT session, the ASA sends a reset and immediately removes the xlate. This reset causes the end node to immediately release the connection, avoiding the TIME_WAIT state. Multi-session PAT, on the other hand, uses the PAT timeout, by default 30 seconds. For “hit-and-run” traffic, such as HTTP or HTTPS, the per-session feature can dramatically increase the connection rate supported by one address. Without the per-session feature, the maximum connection rate for one address for an IP protocol is approximately 2000 per second. With the per-session feature, the connection rate for one address for an IP protocol is $65535/\text{average-lifetime}$. By default, all TCP traffic and UDP DNS traffic use a per-session PAT xlate. For traffic that requires multi-session PAT, such as H.323, SIP, or Skinny, you can disable per-session PAT by creating a per-session deny rule. We introduced the following commands: <code>xlate per-session</code>, <code>show nat pool</code>.</td>
</tr>
</tbody>
</table>

### Transactional Commit Model on NAT Rule Engine

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Platform Releases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transactional Commit Model on NAT Rule Engine</td>
<td>9.3(1)</td>
<td>When enabled, a NAT rule update is applied after the rule compilation is completed; without affecting the rule matching performance. We added the <code>nat</code> keyword to the following commands: <code>asp rule-engine transactional-commit</code>, <code>show running-configuration asp rule-engine transactional-commit</code>, <code>clear configure asp rule-engine transactional-commit</code>.</td>
</tr>
</tbody>
</table>