

# Class Map Configuration Mode Commands

To configure an AppNav or optimization class map, use the **class-map** global configuration command. To unconfigure settings, use the **no** form of this command.

```
class-map type { appnav | waas } [match-all | match-any] classmap-name [rename new-name]
```

```
no class-map type { appnav | waas } [match-all | match-any] classmap-name
```

## Syntax Description

|                               |  |
|-------------------------------|--|
| <b>appnav</b>                 | Configures an AppNav class map.  |
| <b>waas</b>                   | Configures a WAAS optimization class map.  |
| <b>match-all</b>              | (Optional) Specifies that all match conditions must be satisfied to consider the class map matched (logical AND). Valid only on AppNav class maps. |
| <b>match-any</b>              | (Optional) Specifies that any match condition must be satisfied to consider the class map matched (logical OR).                                    |
| <i>classmap-name</i>          | Class map name (up to 40 alpha-numeric characters and hyphen, beginning with a letter).  |
| <b>rename</b> <i>new-name</i> | (Optional) Renames the class map with the specified new name.  |

## Defaults

For AppNav class maps, match-all is the default when multiple match criteria exist.

## Command Modes

global configuration

## Device Modes

application-accelerator  
appnav-controller

## Usage Guidelines

Use the **class-map** command to add or modify class maps and match conditions to identify specific types of traffic for use in policies. This command invokes the Class Map configuration mode, which is indicated by a different prompt (config-cmap). To return to global configuration mode, enter the **exit** command.

You can delete a class map by using the **no** form of this command. You cannot delete a class map if any policies are using it.

When creating a new class map, you must add at least one condition. If any of the conditions specified match an already existing condition in the class-map, no action is taken.



### Note

You cannot have more than 512 different class maps and 1024 total match conditions.

The WAAS software comes with many class maps and policy rules that help your WAAS system classify and optimize some of the most common traffic on your network. Before you create a new class map or policy rule, we recommend that you review the default class map and policy rules and modify them as appropriate. It is usually easier to modify an existing class map or policy rule than to create a new one. For a list of the default applications, class maps, and policy rules, see the *Cisco Wide Area Application Services Configuration Guide*.

**Note**

We strongly recommend that you use the WAAS Central Manager GUI to centrally configure class maps for your WAAS devices. For more information, see the *Cisco Wide Area Application Services Configuration Guide*.

**Examples**

The following example shows how to configure a WAAS optimization class map:

```
wae(config)# class-map type waas myclass1
wae(config-cmap)# description My class number one
wae(config-cmap)# match protocol mapi tcp source ip 10.10.10.35
wae(config-cmap)# exit
```

The following example shows how to configure an AppNav class map:

```
wae(config)# class-map type appnav myclass1
wae(config-cmap)# match peer 50:3d:e5:9c:8f:aa description SanJose_branch
wae(config-cmap)# exit
```

**Related Commands**

[\(config-cmap\) description](#)

[\(config-cmap\) match peer](#)

[\(config-cmap\) match protocol](#)

[\(config-cmap\) match tcp](#)

## (config-cmap) description

To configure the class map description, use the **description** class map configuration command. To unconfigure the description, use the **no** form of this command.

**description** *description*

**no description** *description*

---

|                           |                    |  |
|---------------------------|--------------------|--|
| <b>Syntax Description</b> | <i>description</i> | Specifies a description of the class map with up to 200 alphanumeric and space characters. |
|---------------------------|--------------------|--|

---



---

|                 |                                |
|-----------------|--------------------------------|
| <b>Defaults</b> | No default behavior or values. |
|-----------------|--------------------------------|

---

|                      |                         |
|----------------------|-------------------------|
| <b>Command Modes</b> | Class map configuration |
|----------------------|-------------------------|

---

|                     |  |
|---------------------|--|
| <b>Device Modes</b> | application-accelerator<br>appnav-controller |
|---------------------|--|

---

|                 |   |
|-----------------|---|
| <b>Examples</b> | The following example shows how to configure a class map description: |
|-----------------|---|

```
wae(config)# class-map type waas myclass1
wae(config-cmap)# description My class number one
```

---

|                         |   |
|-------------------------|---|
| <b>Related Commands</b> | <a href="#">(config-cmap) match peer</a><br><a href="#">(config-cmap) match protocol</a><br><a href="#">(config-cmap) match tcp</a> |
|-------------------------|---|

## (config-cmap) match peer

To configure a match condition based on peer ID for the class map, use the **match peer** class map configuration command. To unconfigure a match condition, use the **no** form of this command.

**match peer** *peer-ID* [**description** *description*]

**no match peer** *peer-ID* [**description** *description*]

|                           |                                       |   |
|---------------------------|---------------------------------------|---|
| <b>Syntax Description</b> | <b>peer</b> <i>peer-ID</i>            | Specifies matching based on the device ID of the peer device that is the source or destination of the traffic. The device ID is specified as six hexadecimal bytes separated by colons (aa:bb:cc:dd:ee:ff). |
|                           | <b>description</b> <i>description</i> | Specifies a description of the peer with up to 32 characters.   |

**Defaults** No default behavior or values.

**Command Modes** Class map configuration

**Device Modes** appnav-controller

**Usage Guidelines** This command is available only for configuring AppNav class maps.

**Examples** The following example shows how to configure a peer match condition for a class map:

```
wae(config)# class-map type appnav myclass1
wae(config-cmap)# match peer 50:3d:e5:9c:8f:aa description SanJose_branch
```

**Related Commands**

- [\(config-cmap\) description](#)
- [\(config-cmap\) match protocol](#)
- [\(config-cmap\) match tcp](#)

## (config-cmap) match protocol

To configure a match condition based on layer 7 protocol for the class map, use the **match protocol** class map configuration command. To unconfigure a match condition, use the **no** form of this command. The options for this command differ depending on what type of class map you are configuring.

For an AppNav class map:

```
match protocol {mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql |
  msn-messenger | netlogon} [tcp source [ip ip_address [wildcard_mask]]
```


```
no match protocol {mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql |
  msn-messenger | netlogon} [tcp source [ip ip_address [wildcard_mask]]
```

For a WAAS optimization class map:

```
match protocol {epm-uuid uuid | mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr
  | ms-sql | msn-messenger | netlogon} [tcp source [ip ip_address [wildcard_mask]] [port begin
  [end]]
```

```
no match protocol {epm-uuid uuid | mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api |
  ms-rfr | ms-sql | msn-messenger | netlogon} [tcp source [ip ip_address [wildcard_mask]]
  [port begin [end]]
```

### Syntax Description

|   |   |
|---|---|
| <b>epm-uuid</b> <i>uuid</i>   | Specifies a custom EndPoint Mapper (EPM) service by its Universal Unique ID (UUID). Available only for WAAS optimization class maps.  |
|  |   |
| <b>Note</b>   | If you try to create a class map with an EPM UUID match condition that is already being used, an error message is displayed and the new class map is not created.                             |
| <b>mapi</b>   | Microsoft RPC application keywords:   |
| <b>ms-ad-rep</b>  | Microsoft Exchange MAPI (Exchange Server Store EMSMDB),   |
| <b>ms-exch-nspi</b>   | Microsoft Active Directory Replication (drsuapi),   |
| <b>ms-frs</b>   | Microsoft Active Directory Name Service Provider (NSP),   |
| <b>ms-frs-api</b>   | Microsoft File Replication Services (FRS),  |
| <b>ms-rfr</b>   | Microsoft File Replication API,   |
| <b>ms-sql</b>   | Microsoft Exchange Directory RFR interface,   |
| <b>msn-messenger</b>  | Microsoft SQL,  |
| <b>netlogon</b>   | Microsoft Messenger Service,  |
| <b>tcp source</b>   | Netlogon RPC  |
| <b>ip</b> <i>ip_address</i>   | Specifies the criteria for matching source TCP packets.   |
| <i>wildcard_mask</i>  | Specifies the IP address of the system that is the source of the traffic.   |
| <b>port</b> <i>begin [end]</i>  | A wildcard subnet mask, which matches a range of source IP addresses. Use dotted decimal notation (such as 0.0.0.255 for /24).  |
|   | Specifies the criteria for identifying the port or ports used by the source host. Specify a single port or a begin and end port for a range. Available only for WAAS optimization class maps. |

**(config-cmap) match protocol**

---

**Defaults** No default behavior or values.

---

**Command Modes** Class map configuration

---

**Device Modes** application-accelerator  
appnav-controller

---

**Examples** The following example shows how to configure a protocol match condition for a class map:

```
wae(config)# class-map type waas myclass1
wae(config-cmap)# match protocol mapi tcp source ip 10.10.10.35
```

---

**Related Commands** [\(config-cmap\) description](#)  
[\(config-cmap\) match peer](#)  
[\(config-cmap\) match tcp](#)

## (config-cmap) match tcp

To configure a match condition based on source and/or destination IP address and port for the class map, use the **match tcp** class map configuration command. To unconfigure a match condition, use the **no** form of this command. The options for this command differ depending on what type of class map you are configuring.

For an AppNav class map:

```

match tcp { any |
  destination [ip ip_address [wildcard_mask]] [port begin [end]] [source ip ip_address
  [wildcard_mask] [port begin [end]] |
  source ip ip_address [wildcard_mask] [destination ip ip_address [wildcard_mask] [port begin
  [end]] [protocol { mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql |
  msn-messenger | netlogon }]}

no match { any |
  destination [ip ip_address [wildcard_mask]] [port begin [end]] [source ip ip_address
  [wildcard_mask] [port begin [end]] |
  source ip ip_address [wildcard_mask] [destination ip ip_address [wildcard_mask] [port begin
  [end]] [protocol { mapi | ms-ad-rep | ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql |
  msn-messenger | netlogon }]}

```

For a WAAS optimization class map:

```

match tcp { any |
  destination [ip ip_address [wildcard_mask]] [port begin [end]] [source ip ip_address
  [wildcard_mask] [port begin [end]] |
  source [ip ip_address [wildcard_mask]] [port begin [end]] [destination ip ip_address
  [wildcard_mask] [port begin [end]] [protocol { epm-uuid uuid | mapi | ms-ad-rep |
  ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql | msn-messenger | netlogon }]}

no match tcp { any |
  destination [ip ip_address [wildcard_mask]] [port begin [end]] [source ip ip_address
  [wildcard_mask] [port begin [end]] |
  source [ip ip_address [wildcard_mask]] [port begin [end]] [destination ip ip_address
  [wildcard_mask] [port begin [end]] [protocol { epm-uuid uuid | mapi | ms-ad-rep |
  ms-exch-nspi | ms-frs | ms-frs-api | ms-rfr | ms-sql | msn-messenger | netlogon }]}

```

### Syntax Description

|   |   |
|---|---|
| <b>any</b>                              | Matches any TCP packets.  |
| <b>destination</b>                      | Specifies the destination criteria for matching TCP packets.  |
| <b>ip</b> <i>ip_address</i>             | Specifies the IP address of the system that is the source or destination of the traffic.  |
| <i>wildcard_mask</i>                    | A wildcard subnet mask, which matches a range of IP addresses. Use dotted decimal notation (such as 0.0.0.255 for /24).                                     |
| <b>port</b> <i>begin</i> [ <i>end</i> ] | Specifies the criteria for identifying the port or ports used by the source or destination host. Specify a single port or a begin and end port for a range. |
| <b>source</b>                           | Specifies the source criteria for matching TCP packets.   |
| <b>protocol</b>                         | Specifies the criteria for matching TCP packets based on layer 7 protocol.  |

|                             |  |
|-----------------------------|--|
| <b>epm-uuid</b> <i>uuid</i> | Specifies a custom EndPoint Mapper (EPM) service by its Universal Unique ID (UUID). Available only for WAAS optimization class maps. |
| <b>mapi</b>                 | Microsoft RPC application keywords:  |
| <b>ms-ad-rep</b>            | Microsoft Exchange MAPI (Exchange Server Store EMSMDB),  |
| <b>ms-exch-nspi</b>         | Microsoft Active Directory Replication (drsuapi),  |
| <b>ms-frs</b>               | Microsoft Active Directory Name Service Provider (NSP),  |
| <b>ms-frs-api</b>           | Microsoft File Replication Services (FRS),   |
| <b>ms-frs-api</b>           | Microsoft File Replication API,  |
| <b>ms-rfr</b>               | Microsoft Exchange Directory RFR interface,  |
| <b>ms-sql</b>               | Microsoft SQL,   |
| <b>msn-messenger</b>        | Microsoft Messenger Service,   |
| <b>netlogon</b>             | Netlogon RPC   |

**Defaults** No default behavior or values.

**Command Modes** Class map configuration

**Device Modes** application-accelerator  
appnav-controller

**Examples** The following example shows how to configure a TCP match condition for a class map:

```
wae(config)# class-map type appnav myclass2
wae(config-cmap)# match tcp source port 4000 4004 destination ip 10.10.20.50
```

**Related Commands** [\(config-cmap\) description](#)  
[\(config-cmap\) match peer](#)  
[\(config-cmap\) match protocol](#)