Detailed Configuration

This chapter describes the detailed configuration for AppNav-XE and contains the following sections:

Configuring the AppNav Controller

To configure the AppNav Controller, follow these procedures:

Configuring AppNav Controller Groups

The AppNav Controller group configures the AppNav Controller. To configure the AppNav Controller group, enter the IP addresses used by the AppNav Controllers.

Restrictions

- The AppNav Controller group must always contain exactly one local IP address. This is the IP address of the local AppNav Controller (the local router). Note that this local IP address must belong to an interface from which all the other AppNav Controllers in the AppNav Controller group and all the service nodes are reachable.
- The AppNav Controller group cannot have more than four AppNav Controllers. This must include exactly one local IP address and optionally up to three non-local IP addresses.
- You can use the IP address from GigE, the VLAN interface, the loopback interface etc., but the interface must not have VRF configured.
- The system only supports configuration of one AppNav Controller group.

Use the following command:

```
(config)# [no] service-insertion appnav-controller-group
group-name
```

Submode command:

```
(config-service-insertion-acg)# [no] appnav-controller
IP_address
```

Optional command:

```
(config-service-insertion-acg)# [no] description
 group_description
```

Configuring a Reload Delay

(This feature is available in Cisco IOS XE release 3.10.2.)

Within an AppNav Controller Group (ACG), when a router has just rebooted, it is important to provide a brief delay before designating the router as active and handing over traffic. The delay enables the rebooted router to synchronize flows with the router currently handling traffic. The synchronization helps to avoid unintentionally resetting connections.

During the delay:

- Another AppNav controller in the ACG handles the traffic.
- On the affected device, AppNav enters a mode in which it drops all TCP packets, including border gateway protocol (BGP) packets.
- On the affected device, AppNav drops EIGRP and OSPF.

Command

Use the following command configure the delay:

```
(config-service-insertion-acg)# [no] service-insertion acg-reload-delay
 [
120-450
]
```

The default delay is 120 seconds. Typically, this is sufficient time for synchronizing flows.

The **no** form of the command cancels the delay upon next reload. Executing the **no** form of the command does not cancel the current delay if already applied.

Note Adding the command to the startup-config batch file ensures that the delay is configured during the reboot process.

Requirements for Use

The delay feature is intended for use in the following scenario:

- AppNav Controller Group configured with multiple controllers (see Configuring AppNav Controller Groups, on page 1)
- At least one AppNav service context enabled

- Router has just rebooted

Configuring Service Node Groups

You must configure a service node under a service node group. The AppNav-XE component intelligently distributes flows to the service node within the service node group.

Beginning with the Cisco IOS XE 3.13 release, a total of 64 service nodes may be included in a cluster. (Earlier releases permitted 32.)

Restriction

You cannot use VRF with either the AppNav Controller or the service node IP address. The IP addresses must be explicitly accessible without VRF. For example, you cannot use the management interface's IP address (with vrf Mgmt-intf) as the AppNav Controller IP address.

Use the following command:

```
(config)# [no] service-insertion service-node-group
 group_name
```

Submode commands:

```
(config-service-insertion-sng)# [no] description
 group_description
(config-service-insertion-sng)# [no] service-node
 IP_address
```

Configuring AppNav Class Maps

Use AppNav classes to determine which traffic should be handled by the AppNav-XE component. Use the appnav type class-map to classify the traffic based on the following set of parameters:

- Access list
- Service node peer device ID
- Special protocols supported by the service node

Table 1: ACL and ACE Platform Limits , on page 3 lists the ACL and ACE platform limits for each of the ASR and CSR platforms.

Table 1: ACL and ACE Platform Limits

Platforms	ACLs	ACEs (IPv4)	ACEs per ACL(IPv4)	ACEs (IPv6)	ACEs per ACL(IPv6)
ASR1K ESP-5	4K	25K	15K	8K	4K
ASR1K ESP-10	4K	50K	30K	15K	8K
ASR1K ESP-20/40	4K	100K	60K	30K	16K

Platforms	ACLs	ACEs (IPv4)	ACEs per ACL(IPv4)	ACEs (IPv6)	ACEs per ACL(IPv6)
ASR1K ESP-80/60	4K	400K	80K	200K	8K
CSR 1000V	350 IPv4 or 175 IPv6	1K	550	550	275

To create or modify a class map to be used for matching connections to a specified class, use the **class-map** command in global configuration mode. To remove an existing class map, use the **no** form of this command. The **class-map** command enters class-map configuration mode in which you can enter an optional description command and one or more of the match commands to configure the match criteria for this class.

The syntax for defining a class map is as shown below:

```
(config)# [no] class-map type appnav [match-all | match-any]
appnav_class_name
```

If you do not specify a match, the default is match-all.

Submode commands:

```
(config-cmap)# [no] description
 description_text
(config-cmap)# [no] match access-group
{ACL_number
 | name
ACL_name}
(config-cmap)# [no] match peer
device_ID
(config-cmap)# [no] match protocol
 app_def
```

Match Access-Group Command

The **match access-group** command specifies a numbered access-list or named access list whose contents are used as the match criteria against packets to determine if they belong to this class. The Access List (ACL) number can range from 1 to 2699.

Match Peer Command

The **match peer** command identifies a peer service node that may be performing optimization at the client side of a connection and must be specified in 01:23:45:67:89:ab format. The match peer clause is only useful if the AppNav-XE component is acting as core, that is, receiving a connection that has already been through a peer WAAS device.

Match Protocol Command

The **match protocol** command gets one of the following protocols:

- CITRIX
- MAPI
- MS-AD-REP
- MS-EXCH-NSPI

- MS-FRS
- MS-FRSAPI
- MS-RFR
- MS-SQL
- MSN-MESSENGER
- NETLOGON

The protocol is only used along with additional information provided by the service node to associate the packet with specific applications. The match protocol filter should not be confused with the **monitor-load** keyword in AppNav policy described below.

Configuring AppNav Policy Maps

After you configure the AppNav class maps, you can assign actions to them by using an AppNav policy map.

Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class

Table 2: Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class , on page 5 lists the limits for AppNav policy maps, class maps, and match filters per class.

Table 2: Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class

Policy/Class/Filter Capacity	ASR 1000	CSR 1000V
Unique policy maps	4096 (16000 from Cisco IOS-XE Release 3.10 for RP2, ESP40, ESP100, ESP200 models only)	30
Unique class maps	4096	256
Number of classes per policy map	1000	32
Number of filters per class map	32	8

To create or modify a policy map that defines the service policy for the candidate optimization traffic, use the **policy-map** command in global configuration mode.

```
(config)# [no] policy-map type appnav
 appnav_policy_name
```

Submode commands:

```
(config-pmap)# [no] description
 description_text
(config-pmap)# [no] class
appnav_class_name
```

The **class** command above enters the policy-map-class configuration submode:

```
(config-pmap-c)# [no] distribute service-node-group
```

```
SNG_name
(config-pmap-c)# [no] monitor-load

application_accelerator_name
(config-pmap-c)# [no] pass-through
```

Distribute Command

The **distribute** command is the most common action in this class. The system sends the traffic that matches the class map to the service node group identified by the specified *SNG_name* parameter. If no service node group is available, or if no distribute is specified, the default action is to pass-through the traffic.

To configure primary and backup service node groups, use two **distribute** command statements:

```
(config-pmap-c)# distribute service-node-group

primary_SNG_name
(config-pmap-c)# distribute service-node-group

backup_SNG_name
```

If the service nodes in the primary service node group are not available, the system will use the backup service node group.

Monitor-Load Command

The **monitor-load** command determines which load values should be monitored. When you monitor an application accelerator, the AppNav Controller checks for overload on that application accelerator and does not send new flows to a service node that is overloaded. Flows are sent to a different service node in the service node group.

This command is optional; if you use it, the system monitors the application accelerator indicated by the *application_accelerator_name* parameter. If you do not use this command, the system monitors the TFO accelerator status. If you specify an application accelerator, it replaces the existing monitor-load if one exists.

The supported application accelerators are:

- MS-port-mapper (monitor Microsoft Endpoint Port Mapper load)
- cifs (monitor SMB or CIFS accelerator load)
- http (monitor HTTP accelerator load)
- ica (monitor ICA accelerator load)
- mapi (monitor MAPI accelerator load)
- nfs (monitor NFS accelerator load)
- ssl (monitor SSL accelerator load)
- video (monitor video accelerator load)

Pass-Through Command

Use the **pass-through** command to explicitly indicate that no redirection is to take place. You cannot use the **pass-through** command with the **distribute** or **monitor-load** commands. If you use the **pass-through** command, the system blocks any **distribute** or the **monitor-load** command actions and displays an error message. If you use either the **distribute** or the **monitor-load** command, then the system blocks any **pass-through** command actions.

Configuring Service Contexts

A service context is used to tie the AppNav Controller group, service node group, and AppNav policy map together.

Note If AppNav-XE is managed by WCM, the authentication key in the service-context configuration cannot be modified using the command line interface (CLI)

Use the following command to create a service context:

```
(config)# service-insertion service-context waas/
interface_ID
```

interface_ID is a number that is unique across all service contexts. It determines the naming of the automatically-created virtual interfaces called AppNav-Compress*interface_ID* and AppNav-UnCompress*interface_ID* .

Submode commands:

```
(config-service-insertion-context)# [no] appnav-controller-group
 ACG_name
(config-service-insertion-context)# [no] authentication sha1 key
 authentication_key
(config-service-insertion-context)# [no] service-node-group
 SNG_name
(config-service-insertion-context)# [no] service-policy
appnav_policy_name
(config-service-insertion-context)# [no] vrf { name
VRF_name
 | default | global}
(config-service-insertion-context)# [no] enable
```

AppNav Controller Group Command

ACG_name is the name of the AppNav Controller group to which this service context belongs. You can only configure one AppNav Controller group for each service context.

Authentication SHA1 Key Command

authentication-key is the shared authentication key used during AppNav Controller to service node registration. You must configure the key identically on service nodes in the same service context. Currently, the AppNav Controller group only supports one authentication key. All service contexts must use authentication or no service contexts can use authentication.

Service Node Group Command

SNG_name is the name of one or more service node groups that are part of the service context. The list is used to cross check the ones used in the AppNav policy. Note that the same service node group cannot be shared between two service contexts.

Service Policy Command

appnav_policy_name is the name of the AppNav policy for the service context.

VRF Name Command

VRF_name is the name of the VRF on the LAN interface for the traffic seen by the AppNav-XE component. You can enter more than one VRF name. You can define up to 64 VRF names, but there is no limit to the number of VRFs supported. VRF global is the same as the other VRF definitions except that it identifies traffic with no VRF. The VRF names are listed one after another such as the following:

```
vrf name v1
vrf name v2
vrf name v3
vrf global
```

If you do not configure a VRF in the service context, the system automatically applies the default configuration of vrf default. The purpose of vrf default is to match traffic that does not match a configured VRF name or vrf global.

The following logic is used to pick the right service context for a packet: The system compares the VRF on the LAN interface traversed by the packet against the VRF names (or vrf global) that is configured in the service contexts. If there is a match, the system picks the corresponding service context. If there is no match, the system picks a service context with vrf default, if available. If there is no such service context, then the system passes through the packet.

Enabling AppNav Interception

Currently, the only service supported by the AppNav-XE component is WAAS.

To enable the AppNav-XE component, identify your WAN interface and then use the **service-insertion** command.

```
(config)# interface
 if_name
(config-if)# [no] service-insertion waas
```

Note Both the incoming and outgoing TCP traffic of the interface are subject to AppNav processing according to their VRF and the service policy associated with the service context identified by the VRF.

Configuring the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

This section contains the following subsections:

Enabling the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

To configure the AppNav service node auto discovery feature, perform the following steps:

Procedure

SUMMARY STEPS

1. In Cisco IOS-XE, enter the following command. For the *SNG_name* parameter, enter the name of the service node group for which you want to enable the AppNav service node auto discovery feature. Ensure that the WAAS device is in the same subnet as the AppNav-XE component.
2. Enable the feature by entering the following:
3. On the WAAS device, enter the following command:
4. Select the interface to use and make sure it is in the same subnet as the AppNav-XE service requestor: If interface is not specified, the default is GigabitEthernet0/0.
5. Configure and enable the AppNav service node auto discovery feature by entering the following:

DETAILED STEPS

Step 1 In Cisco IOS-XE, enter the following command. For the *SNG_name* parameter, enter the name of the service node group for which you want to enable the AppNav service node auto discovery feature. Ensure that the WAAS device is in the same subnet as the AppNav-XE component.

Example:

```
router(config)# service-insertion service-node-group
 SNG_name
```

Step 2 Enable the feature by entering the following:

Example:

```
router(config-service-insertion-sng)# node-discovery enable
```

Step 3 On the WAAS device, enter the following command:

Example:

```
WAAS(config)# service-insertion service-node
```

Step 4 Select the interface to use and make sure it is in the same subnet as the AppNav-XE service requestor: If interface is not specified, the default is GigabitEthernet0/0.

Example:

```
WAAS(config)# node-discovery enable GigabitEthernet 0/1
```

Step 5 Configure and enable the AppNav service node auto discovery feature by entering the following:

Example:

```
WAAS(config)# enable
```

Disabling the AppNav Service Node Auto Discovery Feature (For Cisco CSR 1000V Series Only)

You can disable the AppNav service node auto discovery feature by doing either of the following:

- Go to Cisco IOS-XE and disable the entire service node auto discovery feature for the entire system by entering the following:

```
router(config)# service-insertion service-node-group sng
router(config-service-insertion-sng)# no node-discovery enable
```

- Disable the service response feature on the WAAS node, as follows:

```
router(config)# service-insertion service-node
router(config)# no enable
```

Removing the AppNav-XE Configuration

To remove the AppNav-XE configuration, follow these steps:

Procedure

SUMMARY STEPS

1. From configuration mode, remove the interception from the WAN interface. Use these CLI commands:
2. Disable the AppNav service context. Use these CLI commands:
3. Remove the AppNav service context, service node group, and AppNav Controller group. Use these CLI commands:
4. Remove the AppNav policy map, class map, and access list. Use these CLI commands:

DETAILED STEPS

Step 1 From configuration mode, remove the interception from the WAN interface. Use these CLI commands:

Example:

```
router(config)# interface GigabitEthernet0/0/1
router(config-if)# no service-insertion waas
router(config-if)# exit
```

Step 2 Disable the AppNav service context. Use these CLI commands:

Example:

```
router(config)# service-insertion service-context  waas/1
router(config-service-insertion-context)# no enable
router(config-service-insertion-context)# exit
```

Step 3 Remove the AppNav service context, service node group, and AppNav Controller group. Use these CLI commands:

Example:

```
router(config)# no service-insertion service-context  waas/1
router(config)# no service-insertion service-node-group ISR-WAAS-SNG
router(config)# no service-insertion appnav-controller-group ISR-WAAS-SCG
```

Step 4 Remove the AppNav policy map, class map, and access list. Use these CLI commands:

Example:

```
router(config)# no policy-map type appnav ISR-WAAS
router(config)# no class-map type appnav match-any ISR-WAAS
router(config)# no ip access-list extended ISR-WAAS
router(config)# end
```

Configuring Port Channel Support for AppNav-XE

You can configure port channel support for AppNav-XE by indicating to the dataplane to swap IP addresses in the packets so that they can be distributed between different port channels.

To do this, use the following command:

```
(config)# service-insertion swap src-ip
(config)# [no] service-insertion swap src-ip
```

This command also enables AppNav-XE to handle packets from the Service Node whose ip addresses are swapped.