Configuration Guide for Integrated AppNav/AppNav-XE and ISR-WAAS on Cisco 4000 Seires ISRs

Cisco IOS-XE Release 3.13
Revised August 14, 2014

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Configuration Guide for Integrated AppNav/AppNav-XE and ISR-WAAS on Cisco 4000 Seires ISRs

Preface

This preface describes the audience and conventions of the *Configuration Guide for Integrated AppNav/AppNav-XE & ISR-WAAS on Cisco ISR 4451-X*. It also describes the available product documentation and provides information on how to obtain documentation and technical assistance.

Audience

This guide is intended primarily for network administrators, system administrators, and system integrators.

Conventions

This document uses the following conventions:

Convention	Item
boldface font	Commands and keywords.
italic font	Variables for which you supply values.
[enclosed in brackets]	Optional command keywords. You do not have to select any options.
{options enclosed in braces | separated by vertical bar}	Required command keyword to be selected from a set of options. You must choose one option.
`screen` font	Displayed session and system information.
`boldface screen` font	Information you enter.
`italic screen` font	Variables you enter.
Option > Network Preferences	Choosing a menu item.

Note Means *reader take note*.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Related Documentation

The following related documentation is available on Cisco.com:

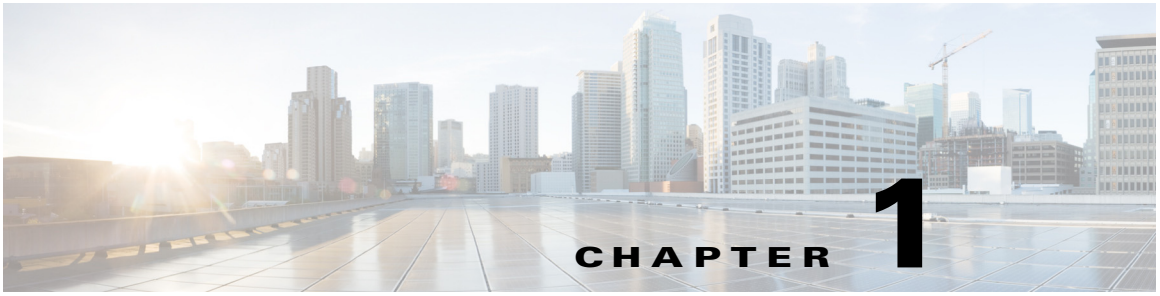
- *Cisco Wide Area Application Services Configuration Guide* (see Cisco WAAS Software Version 5.2.1) at http://www.cisco.com/en/US/products/ps6870/products_installation_and_configuration_guides_list.html
- Cisco Integrated Services Router 4400 Series documents available at http://www.cisco.com/en/US/docs/routers/access/4400/roadmap/isr4400roadmap.html

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



Overview

This guide provides an overview of the ISR-WAAS and AppNav-XE component on Cisco ISR 4451-X in Cisco IOS-XE Release 3.9, and describes the quick start process to easily configure the features. It also provides details of the CLI commands along with examples and troubleshooting tips.

Overview of the WAAS Solution on Cisco ISR 4451-X

The WAAS solution for Cisco ISR 4451-X includes the following:

- ISR-WAAS: Virtualized WAAS in a Cisco IOS-XE container.
- AppNav Controller: Component that intelligently distributes traffic from a router to services.
- AppNav service node auto discovery feature: Feature that automatically discovers service nodes and adds them to an AppNav cluster. See the "About the AppNav Service Node Auto Discovery Feature" section on page 1-3.
- EZConfig: A CLI-based, simplified deployment of the AppNav-XE component and the ISR-WAAS solution on the Cisco ISR 4451-X.
- WAAS Central Manager (WCM): Used to monitor and configure the vWAAS application.

AppNav-XE Component Overview

The AppNav-XE component is made up of a distribution unit called the AppNav Controller and service nodes. The AppNav Controller distributes flows and the service nodes process the flows. Additionally up to four AppNav Controllers can be grouped together to form an AppNav Controller group to support asymmetric flows and high availability. Note that all the routers in the AppNav Controller group need to be the same platform and also have the same memory capacity.

Advantage of Using the AppNav-XE Component

The advantages of using the AppNav-XE component are:

- It can intelligently redirect new flows based on the load on each service node. This includes loads of individual L7 application accelerators.
- For flows that do not require any optimization, service nodes can inform the AppNav Controller to directly pass-through the packets, thereby minimizing the latency and resource utilization.
- There is no impact to traffic when adding or removing service nodes.
- The AppNav-XE component supports VRF so that VRF information is preserved when traffic returns from a service node.
- For special applications such as MAPI (Exchange) and VDI (Citrix), the AppNav-XE component ensures that flows from the same client and destined to the same server and server port are redirected to the same service node.
- You can use an AppNav Controller group to optimize asymmetric flows. An asymmetric flow is when the traffic in one direction goes through one AppNav Controller and the return traffic goes through a different AppNav Controller, but both AppNav Controllers redirect the traffic to the same service node.
- Inter-router high availability, where if one router goes down, traffic can be rerouted to a different router within the AppNav Controller group, keeping the traffic flows uninterrupted.

Interoperability of the AppNav-XE Component

The AppNav component can interoperate with the following features on the router:

- QoS
- NAT (Note that the video application accelerator is disabled and that asymmetric routing and inter-router high availability handled both by the AppNav-XE component and NAT is not supported.)
- AVC 2.0 (FNF, NBAR) (Note that AVC 2.0 does not support symmetric routing and inter-router high availability.)
- IPSec
- GET-VPN
- EzVPN
- DMVPN
- ACL
- VRF
- MPLS (The supported topology is an MPLS network on the WAN side and an IP network on the LAN side.)

- WCCP-AppNav-XE coexistence (WCCP and AppNav-XE can be configured on the same interface only if they act on different flows. Use ACLs for this. WCCP and AppNav XE can be configured on different interfaces—AppNav-XE on WAN and WCCP on LAN (supported on Cisco IOS-XE Release 3.10 and later.)
- PBR/PFR (supported on Cisco IOS-XE Release 3.10.1 and later)

The AppNav-XE component introduces the concept of a virtual interface, which allows users to configure features specific to compressed or uncompressed traffic. For instance, to monitor the traffic that is being redirected to the service node and the traffic that is returning from the service node, you can configure the FNF feature on the AppNav-UnCompress and AppNav-Compress virtual interfaces. Note that these AppNav-XE virtual interfaces appear to the user just as any other interface. However from the above list, the only features that work on the AppNav-XE virtual interfaces are FNF, ACL, and QoS (except for queueing).

About Configuring the AppNav-XE Component

Note the following points regarding configuring the AppNav-XE component:

- You must identify the WAN interfaces for the router that is running the AppNav Controller. The AppNav Controller intercepts packets on both ingress and egress of WAN interface. Only configure the AppNav Controller on WAN interfaces, including all WAN interfaces that will be load balancing.
- Do not use the VRF to access the service node from the AppNav Controller. Neither the service node nor the AppNav Controller IP address should have VRF on the AppNav Controller.
- You can use port channel between the AppNav Controller and the service nodes to increase AppNav Controller-service node bandwidth.
- The **config replace** command cannot be used with AppNav-XE configuration.
- If you use an AppNav Controller group with two or more AppNav Controllers, the AppNav-XE configuration on all the AppNav Controllers must be the same. This also means that the names of the AppNav policy maps and class maps on the AppNav Controllers need to match. Also the VRF names for the traffic seen by the AppNav-XE component need to be the same on all the AppNav Controllers.
- If AppNav-XE is managed by WCM, the authentication key in the service-context configuration cannot be modified using the command line interface (CLI).

For additional information and caveats about configuring the AppNav-XE component, see Chapter 3, “Detailed Configuration”.

About the AppNav Service Node Auto Discovery Feature

The AppNav service node auto discovery feature is targeted for small branch installations. With this feature, the system automatically discovers the service nodes within the same L2 connectivity of the AppNav router and adds them to the service node cluster.

Restriction

The AppNav service node auto discovery feature can only be enabled on one interface on a service node.

To enable the AppNav service node auto discovery feature, do the following:

Procedure

Step 1 Initiate a discovery request on the AppNav-XE component on the router by doing the following:

a. Determine the service node group for which you want to enable the auto discovery.

b. Issue the following commands:

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

Step 2 Initiate a service respond on the service nodes by doing the following:

a. On the WAAS appliance, determine the interface for which you want to enable node discovery. This interface must be in the same subnet as the AppNav Controller.

b. Enable node discovery by issuing the following commands:

```
auto-sn(config)# service-insertion service-node
auto-sn(config-sn)# node-discovery enable GigabitEthernet 0/1
auto-sn(config-sn)# enable
```

Container Overview

The term “container” refers to the KVM hypervisor that runs virtualized applications on the Cisco ISR 4451-X. The term “host” refers to the primary operating system running on a system. For ISR-WAAS on Cisco ISR 4451-X, the host is defined as a Cisco ISR 4451-X running on Cisco IOS XE Release 3.9.

The Virtualization Manager tasks vary depending on the phase of the virtual service deployment. Table 1-1 summarizes this information.

Table 1-1 Virtualization Manager Tasks

Phase	Trigger	Actions	Virtual Service Instance State
Pre-Installation		1. Gather and prepare system resources. 2. Establish internal communication infrastructures.	Host is ready to accept new virtual service.
Installation	Virtualization Manager received a request to install a virtual service package.	1. Unzip and unpack the virtual service definition from its OVA package. 2. Perform SHA2 code signing check using the artifacts in the OVA (.cert, .mf) and a hidden Cisco public key. 3. Validate the machine definition specified in the OVA and perform preliminary resource check (for warnings). 4. Parse the machine definition and create internal objects for manageability. 5. Process tiered resource profiles requests.	• Validated that package is Cisco signed. • Validated integrity of OVA content. • Validated and parsed machine definition and binds it to a virtual service “instance name”.
Configuration	Virtualization infrastructure received a request to configure an instance of the virtual service.	1. Perform validation and necessary network provisioning for configured guest IP address (if applicable). 2. Perform resource check and reservation for selected profile.	Virtual service is configured.
Activation	Virtualization infrastructure received a request to activate the virtual service.	1. Carve out storage resource from host system as needed. 2. Commit CPU, memory, storage, and networking resources as needed. 3. Update the machine definition XML and start the virtual machine. 4. Service to console, aux, logging and tracing ports as needed.	Virtual service is activated.
Post Activation		1. Perform monitoring services. 2. Process lifecycle control services.	

ISR-WAAS Overview

ISR-WAAS is a virtualized WAAS instance running on a Cisco IOS-XE container on a Cisco ISR 4451-X platform. ISR-WAAS provides WAN optimization functionality to the Cisco ISR 4451-X.

The Cisco ISR 4451-X does not support RAID.

ISR-WAAS can run on a Cisco ISR 4451-X router with these minimum requirements:

- 8 GB RAM
- 200 GB hard disk

The Cisco ISR 4451-X requires more resources depending on the ISR-WAAS profile that you install. See Table 1-2.

Table 1-2 Profile Specifications

	Profile Specifications				Router Requirements			
Profile Name	**Connections**	**RAM (GB)**	**Disk (GB)**	**vCPUs**	**RAM**	**# of 200 GB SSD disks**	**CF (GB)**	**Target WAN throughput**
ISR-WAAS-750	750	4	170	2	8	1	16	50 Mbps
ISR-WAAS-1300	1300	6	170	4	16	1	32	100 Mbps
ISR-WAAS-2500	2500	8	360	6	16	2	32	150 Mbps

Licensing Requirements

To deploy both the ISR-WAAS and the AppNav-XE component on the Cisco ISR 4451-X, use the appxk9 package license.

Procedure

Step 1 Enter the licensing command as follows:

```
router(config)# license boot level appxk9
router(config)# end
router# write mem
```

Step 2 Reload the router using the following command:

```
router# reload
```

Step 3 Enter the **show license detail** command as follows:

```
router# show license detail
```

Step 4 To verify that the license is enabled, review the output of the command. Verify that the appxk9 package license is active and in use. The output for “Feature: appxk9” should show “License State: Active, In Use”. Here is an example:

```
router# show license detail
Index 1: Feature: appxk9          Version 1.0
License Type: EvalRightToUse
License State: Active, In Use
    Evaluation total period: 8 weeks 4 days
```

```
    Evaluation period left: 7 weeks 6days
    Period used: 4 days 14 hours
    Transition date: Apr 15 2013 10:27:31
Lock type: Non Node locked
```

Recommendations to Upgrade/Downgrade theCisco ISR 4451-X Running WAAS Software

To upgrade/downgrade an ISR-4451-X running WAAS software for different versions of Cisco IOS XE, Cisco recommends the following steps.

For Cisco IOS-XE 3.9

For the Cisco ISR 4451-X running Cisco IOS-XE 3.9 release, follow these steps:

Step 1 Install ISR-WAAS version 5.2 using the **service waas enable** command. The appropriate OVA file should be on the flash drive of the Cisco ISR 4451-X.

Step 2 To upgrade from WAAS 5.2 to WAAS 5.3, use the bin image (copy ftp/http install) process using the CLI or Central Manager.

Step 3 To downgrade back to WAAS 5.2.1, uninstall the current version using the **service waas disable** command and install a new image of WAAS using step 1 above.

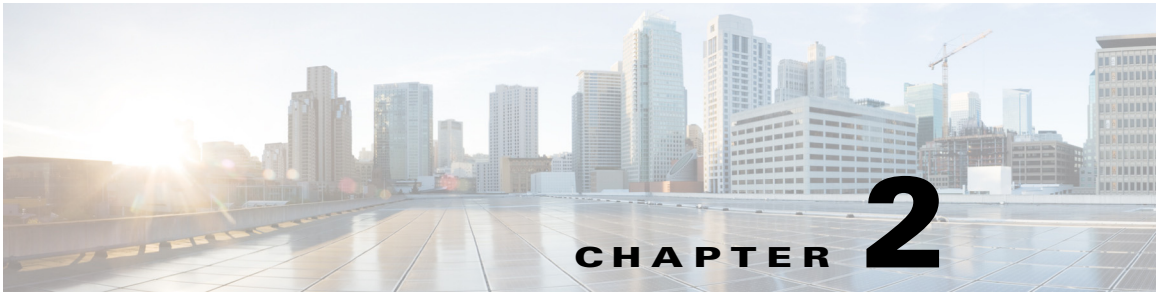
For Cisco IOS-XE 3.10

For the Cisco ISR 4451-X running Cisco IOS-XE 3.10 release, follow these steps:

Step 1 Install ISR-WAAS version 5.3 only. Do not downgrade IOS-XE to earlier versions.

Step 2 If IOS-XE 3.9 version is required, first uninstall WAAS 5.3 using the **service waas disable** command, then downgrade IOS-XE 3.10 to version 3.9.

Step 3 Enable WAAS 5.2 using the appropriate OVA file and the **service waas enable** command.



Quick Start for Branch Users Using Cisco ISR-4451-X

This chapter describes how to get started quickly with ISR-WAAS on the Cisco ISR 4451-X.

Prerequisites and Requirements for Using the EZConfig Program

This ection contains the following subsections:

Prerequisites

Before getting started with ISR-WAAS on the Cisco ISR 4451-X, ensure that you have the following:

- Cisco ISR 4451-X router with 8 GB RAM, 16GB compact flash memory, and 200 GB hard disk
- Valid cisco.com username and password
- CCO image from Cisco.com
- ISR-WAAS package file (OVA) that is shipped on the boot flash.
- Wide Area Virtualization Engine (WAVE) appliance as the peer device
- WAAS Central Manager (WCM)

Requirements

- Cisco recommends that you connect the Cisco ISR 4451-X router to the WCM before using the EZConfig program. If you do not connect them, the EZConfig program still starts the service but displays an error at the end. You can later register the service with the WCM by using the WCM-specific commands **service waas wcm ip address** *ip_address* and **service waas wcm deregister**.
- You also need to configure an SRV record on a DNS server that is reachable from the router. The system uses the SRV record to look up the IP address of the WCM.
- You must manually configure the WAN and LAN interfaces and the WAN connectivity routes before you can use the EZConfig program.
- The EZConfig program attempts to enable the ISR-WAAS solution on the WAN interfaces that you enter. If the WAN interfaces are down or not configured for proper connectivity, the EZConfig program still configures the ISR-WAAS solution but the WAAS optimization functionality will not work until the WAN interfaces are enabled properly.
- If there is an existing configuration for ISR-WAAS (both the AppNav-XE component and the virtual container) on the Cisco ISR 4451-X that is not associated with the name AUTOWAAS, then you must manually clean the configuration before running the EZConfig program. EZConfig uses the name AUTOWAAS for the entire internal configuration naming so that the configuration can be tracked together and shown as being associated with the EZConfig menu.

Enabling ISR-WAAS on a Cisco ISR 4451-X Using the EZConfig Program

The EZConfig program is a single CLI command that launches an interactive mode for enabling ISR-WAAS on the Cisco ISR 4451-X. The program walks you through a series of questions and enables the corresponding AppNav Controller, container, interface, and connected application configurations.

Using the EZConfig Program

To run the EZConfig program, issue the following CLI command on a Cisco ISR 4451-X while logged in with privilege 15:

```
router# service waas enable
```

The system displays a welcome message and prompts you for several input parameters, as explained in subsequent sections.

```
router# service waas enable
**************************************************************************
****  Entering WAAS service interactive mode.                         ****
****  You will be asked a series of questions, and your answers       ****
****  will be used to modify this device's configuration to           ****
****  enable a WAAS Service on this router.                           ****
**************************************************************************

Continue? [y]:

At any time: ? for help, CTRL-C to exit.

Existing/conflicting WAAS configuration found.
Do you want to clean existing configuration so a fresh configuration through this
interactive menu can proceed? [y]: y
% Virtual service AUTOWAAS was not activated
deactivating!!!!!!!!!!!!

removing previous profile extraction

*Nov 14 18:29:12.911: %VIRT_SERVICE-5-INSTALL_STATE: Successfully uninstalled virtual
service AUTOWAAS

Continuing with WAAS service enablement...
```

Selecting the OVA Package

The EZConfig program searches the router storage devices, router flash, and hard disks for ISR-WAAS images with the following Cisco-approved naming convention: **ISR4451-X-WAAS*.ova**. The system only uses images with this naming convention as choices. If the system only finds one OVA package, it automatically uses it.

Here is an example of the system only finding one OVA package:

```
Only one WAAS image found locally (harddisk:/ISR4451-X-WAAS-eft.ova) - using as default

Extracting profile from harddisk:/vWAAS-kvm-5.1.2-SP4-b9.ova, this may take a couple of
minutes ...
```

Here is an example of the system finding multiple OVA packages:

```
Select a WAAS image to install:
1.harddisk:/ISR4451-X-WAAS-eft.ova
2.harddisk:/ISR4451-X-WAAS-test.ova
3.Enter your own image

Select image  [2]:

Extracting profile from harddisk:/ISR4451-X-WAAS-test.ova, this may take a couple of
minutes ...
```

The system sets the OVA image with the latest timestamp as the default. If you press enter without selecting an image, the system uses the default image. If you choose to enter an image name, the system prompts you for the image location and name, as in the example below. However, you can only select an image from the hard disk or the router flash.

```
Select a WAAS image to install:
1.harddisk:/ISR4451-X-WAAS-eft.ova
2.harddisk:/ISR4451-X-WAAS-test.ova
3.Enter your own image

Select image  [2]: 3
```

```
Enter the local WAAS image to install. (blank to return) []:
harddisk:/ISR4451-X-WAAS-myfile.ova
```

Selecting the ISR-WAAS Profile

Each ISR-WAAS image is shipped with multiple profiles. The profiles dictate the resources used by the ISR-WAAS virtual instance and the number of connections supported. The system prompts you to select a profile, as in the following example:

```
These are the available profiles
1.  ISR-WAAS-2500
2.  ISR-WAAS-1300
3.  ISR-WAAS-750

Choose profile  [1]:
```

The system sets the profile with the highest number of connections as the default. You can press enter to select the default profile or select a different profile.

Entering the Host IP Address and the ISR-WAAS Service IP Address

You must enter an internal IP address and subnet mask for the host, as well as an IP address for the ISR-WAAS service. The ISR-WAAS service IP address must be in the same network as the host. The host IP address is the address used by the router to communicate with the container.

Alternatively, you can specify that the ISR-WAAS service IP address be in the same subnet as one of the active router interfaces. This interface IP address can then be borrowed as the host IP address using IP unnumbered. The system inserts a static route to divert traffic to ISR-WAAS.

Note

- The system supports IPv4 and IPv6.
- The host IP address and subnet mask must be in the format “a.b.c.d/nn” or “a.b.c.d a.b.c.d”.

The following example illustrates the EZConfig IP address prompt. The system does not prompt you for the host IP address:

```
The following ip address type supported for WAAS
 1) ipv4
 2) ipv6

Select ip address type (1 or 2):2

Enter the IP address to be configured on the WAAS service: 10:10:10:10::10

The following ip address type supported for Host on Router
 1) ipv4
 2) ipv6

Select ip address type (1 or 2):2

Enter the IP address to be configured on this router: 10:10:10:11::10/24
```

Because the service IP address entered was in the same subnet as one of the interfaces, say GigabitEthernet0/0/1, the system borrows the service IP address for the host from this interface using IP unnumbered. If you are not using IP unnumbered, the system prompts you to enter the ISR-WAAS service IP address, the host IP address, and the subnet mask.

Entering the WAAS Central Manager (WCM) IP Address

The WAAS Central Manager (WCM) manages the WAAS service. It configures the WAAS policy and the application accelerators. The IP address of the WCM must be reachable from the router; otherwise, the WCM registration fails. If this happens, the EZConfig program continues with the remaining configuration and you can manually connect ISR-WAAS to WCM using the **service waas wcm** command.

The system uses a DNS SRV record to look up the WCM IP address. The system sends the IP address to ISR-WAAS for registration with the WCM. In order to successfully look up the IP address, make sure that the following record is available on a DNS server that is reachable from the router:

```
_waascms._tcp.cisco.com
Example:
ip host waas-cm.cisco.com 100.0.0.100
ip host _waascms._tcp.ciscowaas.local srv 1 100 8443 waas-cm.cisco.com
```

If the system cannot reach the DNS server, or if there is no SRV record, and the system cannot obtain the IP address of the WCM, the system prompts you to manually enter it.

Use the **show virtual-service detail** command to check the status of the WCM registration.

The following is an example of the EZConfig WCM IP address prompt:

```
The following ip address type supported for WAAS Central Manager
 1) ipv4
 2) ipv6

Select ip address type (1 or 2):2

Enter the IP address of the WAAS Central Manager (WCM): 10:10:10:10::12
```

After the EZConfig program installs and activates the ISR-WAAS virtual instance, the system displays the virtual instance in the WCM.

Entering the WAAS Interception Interfaces

The EZConfig program displays a list of interfaces on the router. Enter the WAN interfaces where WAAS functionality is enabled using the underlying WAAS interception and routing mechanism AppNav.

Note You cannot use the GigabitEthernet0 interface because it is a management interface used by the router.

See the following example:

```
The following IP interfaces are currently available on the router:

Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   192.168.9.1     YES NVRAM  up                    up
GigabitEthernet0/0/1   10.10.9.1       YES NVRAM  up                    up
GigabitEthernet0/0/2   100.0.0.2       YES NVRAM  up                    up
GigabitEthernet0/0/3   unassigned      YES NVRAM  down                  down
```

```
GigabitEthernet0       1.1.220.8       YES NVRAM  up                    up

Enter a WAN interface to enable WAAS interception (blank to skip) []: Gi 0/0/0

Enter additional WAN interface (blank to finish) []:
```

Verifying Input

After you enter all the requested information, the EZConfig program displays a configuration summary so that you can review the inputs and modify them if needed. The following is a sample:

```
***************************
** Configuration Summary: **
***************************

a) WAAS Image and Profile Size:
   bootflash:/ISR4451-X-WAAS-eft.ova  (1331268265) bytes
    ISR-WAAS-750

b) Router IP/mask:
    Using IP unnumbered from interface GigabitEthernet0/0/1

   WAAS Service IP:
   10.10.9.10

c) WAAS Central Manager:
   100.0.0.1

d) Router WAN Interfaces:
   GigabitEthernet0/0/0

Choose letter 'a-d' to edit, 'v' to view config script, 's' to apply config [s]:
```

If you select **s**, which is the default, the system applies the above configuration. If you choose a letter **a** through **d**, the system prompts you to modify the configuration that you chose. If you select **v**, the system displays the configuration as shown below:

```
The following configuration will be applied:
interface VirtualPortGroup31
ip unnumbered GigabitEthernet0/0/1
ip route 10.10.9.10 255.255.255.255 VirtualPortGroup31

virtual-service AUTOWAAS
interface VirtualPortGroup31
ip address 10.10.9.10
exit
profile ISR-WAAS-750
activate

interface GigabitEthernet0/0/2
service-insertion waas
exit

service-insertion service-node-group AUTOWAAS-SNG
description AUTOWAAS
node-discovery enable
service-node 10.10.9.10
exit

service-insertion appnav-controller-group AUTOWAAS-SCG
description AUTOWAAS
appnav-controller 10.10.9.1
```

```
exit

ip access-list extended EPMAP
permit tcp any any eq 135
ip access-list extended NFS
permit tcp any any eq 2049
ip access-list extended HTTPS
permit tcp any any eq 443
ip access-list extended CIFS
permit tcp any any eq 139
permit tcp any any eq 445
ip access-list extended RTSP
permit tcp any any eq 554
permit tcp any any eq 8554
ip access-list extended Citrix-ICA
permit tcp any any eq 1494
ip access-list extended Citrix-CGP
permit tcp any any eq 2598
ip access-list extended HTTP
permit tcp any any eq 80
permit tcp any any eq 3218
permit tcp any any eq 8000
permit tcp any any eq 8080
permit tcp any any eq 8088
ip access-list extended SN_OR_WCM
permit tcp host 10.10.9.10 any
permit tcp any host 10.10.9.10
permit tcp host 100.0.0.1 any
permit tcp any host 100.0.0.1
ip access-list extended AUTOWAAS
permit tcp any any
class-map type appnav match-any SN_or_WCM
match access-group name SN_or_WCM
class-map type appnav match-any NFS
match access-group name NFS
class-map type appnav match-any HTTP
match access-group name HTTP
class-map type appnav match-any HTTPS
match access-group name HTTPS
class-map type appnav match-any CIFS
match access-group name CIFS
class-map type appnav match-any MAPI
match protocol mapi
class-map type appnav match-any RTSP
match access-group name RTSP
class-map type appnav match-any Citrix-ICA
match access-group name Citrix-ICA
class-map type appnav match-any Citrix-CGP
match access-group name Citrix-CGP
class-map type appnav match-any AUTOWAAS
match access-group name AUTOWAAS
policy-map type appnav AUTOWAAS
description AUTOWAAS global policy
class SN_OR_WCM
pass-through
class HTTP
distribute service-node-group AUTOWAAS-SNG
monitor-load http
class MAPI
distribute service-node-group AUTOWAAS-SNG
monitor-load mapi
class HTTPS
distribute service-node-group AUTOWAAS-SNG
monitor-load ssl
```

```
class CIFS
distribute service-node-group AUTOWAAS-SNG
monitor-load cifs
class Citrix-ICA
distribute service-node-group AUTOWAAS-SNG
monitor-load ica
class Citrix-CGP
distribute service-node-group AUTOWAAS-SNG
monitor-load ica
class EPMAP
distribute service-node-group AUTOWAAS-SNG
monitor-load MS-port-mapper
class NFS
distribute service-node-group AUTOWAAS-SNG
monitor-load nfs
class RTSP
distribute service-node-group AUTOWAAS-SNG
monitor-load video
class AUTOWAAS
distribute service-node-group AUTOWAAS-SNG
service-insertion service-context waas/1
service-policy AUTOWAAS
service-node-group AUTOWAAS-SNG
appnav-controller-group AUTOWAAS-SCG
enable

service waas wcm ip address 100.0.0.1
****************************
** Configuration Summary: **
****************************

a) WAAS Image and Profile Size:
harddisk:/ISR4451-X-WAAS-eft.ova (1331268265) bytes
ISR-WAAS-750

b) Router IP/mask:
Using ip unnumbered from interface GigabitEthernet0/0/1
WAAS Service IP:
10.10.9.10

c) WAAS Central Manager:
100.0.0.100

d) Router WAN Interfaces:
GigabitEthernet0/0/0

Choose letter 'a-d' to edit, 'v' to view config script, 's' to apply config [s]: c

Enter the IP address of the WAAS Central Manager (WCM): 100.0.0.1

****************************
** Configuration Summary: **
****************************
a) WAAS Image and Profile Size:
harddisk:/ISR4451-X-WAAS-eft.ova (1331268265) bytes
ISR-WAAS-750

b) Router IP/mask:
Using ip unnumbered from interface GigabitEthernet0/0/1

WAAS Service IP:
10.10.9.10
```

```
c) WAAS Central Manager:
100.0.0.1

d) Router WAN Interfaces:
GigabitEthernet0/0/1

Choose letter 'a-d' to edit, 'v' to view config script, 's' to apply config [s]: b

An internal IP interface and subnet is required to deploy a WAAS service on this router.
This internal subnet must contain two usable IP addresses that can route and communicate
with the WAAS Central Manager (WCM).

Enter the IP address to be configured on the WAAS service: 9.9.9.1

Enter the IP address/mask to be configured on this router: 9.9.9.2/24


****************************
** Configuration Summary: **
****************************

a) WAAS Image and Profile Size:
harddisk:/ISR4451-X-WAAS-eft.ova (1331268265) bytes
ISR-WAAS-750

b) Router IP/mask:
9.9.9.2
255.255.255.0

WAAS Service IP:
9.9.9.1

c) WAAS Central Manager:
100.100.0.1

d) Router WAN Interfaces:
GigabitEthernet0/0/0

Choose letter 'a-d' to edit, 'v' to view config script, 's' to apply config [s]:
```

Applying the Configuration

After verifying the configuration, the EZConfig program displays the progress of the WAAS installation and activation. The system then applies the configuration and displays the status of the WAAS virtual service. See the following example:

```
The configuration will be applied and the status of the WAAS service will be displayed
after deployment
Installing bootflash:/ISR4451-X-WAAS-eft.ova
installing!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!
*Dec 13 04:52:07.227: %VIRT_SERVICE-5-INSTALL_STATE: Successfully installed virtual
service AUTOWAAS
System is attempting to deploy and activate WAAS image, this may take up to 10 minutes
activating!!!!

*Dec 13 04:52:26.718: %VIRT_SERVICE-5-ACTIVATION_STATE: Successfully activated virtual
service AUTOWAAS
*Dec 13 04:52:28.717: %LINK-3-UPDOWN: Interface VirtualPortGroup31, changed state to up
*Dec 13 04:52:29.717: %LINEPROTO-5-UPDOWN: Line protocol on Interface VirtualPortGroup31,
changed state to up
```

```
Waiting for WAAS application to be at a stage to accept WCM IP configuration.

Waiting!
*Dec 13 04:52:31.080: %LINK-3-UPDOWN: Interface AppNav-Compress1, changed state to up
*Dec 13 04:52:32.047: %LINEPROTO-5-UPDOWN: Line protocol on Interface AppNav-Compress1,
changed state to up
*Dec 13 04:52:32.047: %LINEPROTO-5-UPDOWN: Line protocol on Interface AppNav-UnCompress1,
changed state to up
*Dec 13 04:52:32.048: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
state to
up!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!
WAAS service activated!
Note: Please issue "copy running-config startup-config" command to save changes!
```

Disabling the WAAS Service Using the EZConfig Program

The EZConfig program uses the name **AUTOWAAS** for the virtual service, AppNav, class map, and policy map configuration. Whenever you run the EZConfig program, the system checks the configuration for any previously configured virtual instances and any AppNav configurations named AUTOWAAS. If the system finds any, the EZConfig program prompts you to clean up the system before enabling the WAAS service. See the following example:

```
router# service waas enable
**************************************************************************
****  Entering WAAS service interactive mode.                         ****
****  You will be asked a series of questions, and your answers       ****
****  will be used to modify this device's configuration to           ****
****  enable a WAAS Service on this router.                           ****
**************************************************************************

Continue? [y]: y

At any time: ? for help, CTRL-C to exit.

Existing/conflicting WAAS configuration found.

Do you want to clean existing configuration so a fresh configuration through this
interactive menu can proceed? [y]: y

deactivating!!!!!!!!!!

removing previous profile extraction

*Aug 29 00:35:46.126: %VIRT_SERVICE-5-INSTALL_STATE: Successfully uninstalled virtual
service AUTOWAAS

Continuing with WAAS service enablement...
```

Another way to clean up old EZConfig configurations is to use the **service waas disable** command. This command deactivates the virtual instance, uninstalls the OVA image, and removes all configurations with the name AUTOWAAS. See the following example:

```
router# service waas disable
***************************************************
**    WAAS disable service interactive mode.     **
**    You will be asked a series of questions    **
**    and your answers will be used to *REMOVE*  **
**    the WAAS and AppNav Service configuration  **
```

```
**    on this router.                              **
************************************************

Are you sure you want to remove 'AUTOWAAS' service and configuration for WAAS/AppNav?
[yes]: yes
deactivating!!!!!!!!!!

*Aug 29 00:51:12.912: %LINK-3-UPDOWN: Interface VirtualPortGroup31, changed state to down
*Aug 29 00:51:13.913: %LINEPROTO-5-UPDOWN: Line protocol on Interface VirtualPortGroup31,
changed state to down
*Aug 29 00:51:20.268: %LINK-5-CHANGED: Interface VirtualPortGroup31, changed state to
administratively down
*Aug 29 00:51:21.297: %VIRT_SERVICE-5-INSTALL_STATE: Successfully uninstalled virtual
service

AUTOWAAS
WAAS/AppNav configuration and service removed!
Note:Please issue "copy running-config startup-config" command to save changes!
```

Automatic Configuration Entries

In addition to setting the ISR-WAAS IP address and default gateway configuration entries (either using the EZConfig program or setting them manually as described in Chapter 3, “Detailed Configuration”), the system also automatically applies the following host router configurations entries to the ISR-WAAS:

Hostname

The system automatically sets the ISR-WAAS hostname to be “Router-” followed by the router hostname, as in the following example:

```
hostname Router-ISR-WAAS
```

Domain Name

The system automatically sets the ISR-WAAS domain name to the same domain name as the host router.

```
ip domain-name cisco.com
```

Timezone

The system automatically sets the ISR-WAAS timezone configuration to the same timezone setting as the host router.

```
clock timezone PDT -8 0
```

NTP Servers

The host router allows up to six NTP servers but ISR-WAAS only allows up to three NTP servers. The system uses the first three NTP servers configured on the router for ISR-WAAS.

```
ntp server 50.116.38.157
ntp server 199.102.46.72
```

The system also configures the WCM server with the same clock source.

DNS Server

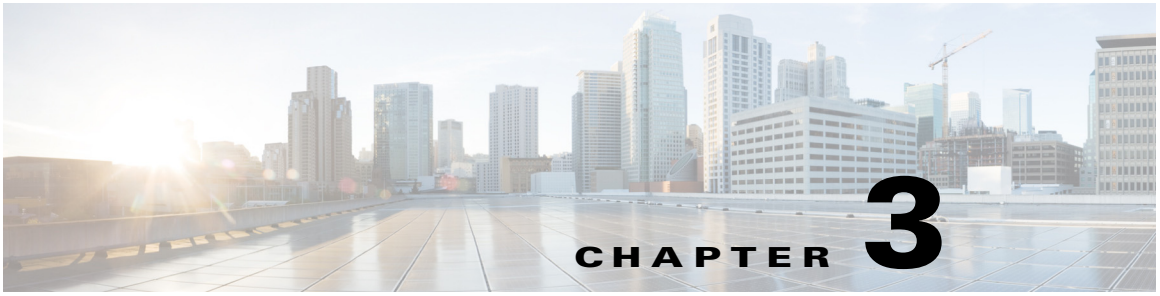
The host router allows up to six DNS servers but ISR-WAAS only allows up to three DNS servers. The system uses the first three DNS servers configured on the router for ISR-WAAS.

```
ip name-server 208.67.222.222
ip name-server 208.67.220.220
```

WAAS Central Manager (WCM) Changes for ISR-WAAS

Below are the changes in the WCM GUI that are unique to ISR-WAAS:

- The following pages under the **Configure > Network** tab are read-only:
  - Network Interfaces
  - Default Gateway
  - DNS
- The **Jumbo MTU** link under the **Configure > Network** tab is unavailable.
- On the **Configure > Interception > Interception Configuration** page, the AppNav Controller is the only supported interception method.
- The CIFS accelerator is unavailable in ISR-WAAS, which leads to the following:
  - The CIFS Accelerator and Windows Print Accelerator fields under the **Configure > Acceleration > Enabled Features** page are unavailable on the device level and appropriate warning message will be provided in the device group level. The WCM ensures that CIFS-related configurations are not pushed to the ISR-WAAS application from DG level to prevent the device-level page from going into override mode.
  - The CIFS Acceleration Report under the **Monitor > Acceleration** tab is unavailable.
  - The ISR-WAAS application is not supported on the Preposition and Dynamic shares pages under the **Home > CIFS File Services** tab.



Detailed Configuration

This chapter provides detailed configuration information for AppNav-XE on the Cisco ISR 4451-X and contains the following sections:

Configuring the AppNav Controller

To configure the AppNav-XE Controller, follow these configurations tasks:

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.
- When you have two Cisco ISR AppNav controllers but only a single WAN link, then configure AppNav redirection only on an active WAN router. If the secondary AppNav controllers WAN link is not enabled, the AppNAv controller will drop the queries from an active WAN router / AppNav controller and pass-through traffic will be dropped.
- 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 description_text
```

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 sng_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 3-1 lists the ACL and ACE platform limits for the Cisco ISR 4451-X platform.

Table 3-1 ACL and ACE Platform Limits

Platforms	ACLs	ACEs (IPv4)	ACEs per ACL(IPv4)	ACEs (IPv6)	ACEs per ACL(IPv6)
Cisco ISR 4451-X	4K	20K	20K	10K	10K

•

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 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 3-2 lists the limits for AppNav policy maps, class maps, and match filters per class.

Table 3-2 Limits for AppNav Policy Maps, Class maps, and Match Filters Per Class

Policy/Class/Filter Capacity	Cisco ISR 4451-X
Unique policy maps	4096 (16000 from Cisco IOS-XE Release 3.10 for RP2, ESP40, ESP100, ESP200 models only)
Unique class maps	4096
Number of classes per policy map	256
Number of filters per class map	32

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 virtual_instance_name/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 AppNav. 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 interface_name
(config-if)# [no] service-insertion virtual_instance_name
```

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

This section contains the following sub-sections:

Enabling the AppNav Service Node Auto Discovery Feature

Note Configuring the AppNav service node auto discovery feature is only applicable if you do not use the EZConfig program. If you do use the EZConfig program, you do not need to configure the AppNav service node auto discovery feature.

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

Procedure

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.

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

Step 2 Enable the feature by entering the following:

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

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

```
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 service requestor:

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

Note If interface is not specified, the default is GigabitEthernet0/0

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

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

Disabling the AppNav Service Node Auto Discovery Feature

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 service node, as follows:

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

Configuring the Container

Note Configuring the container is only applicable if you do not use the EZConfig program. If you use the EZConfig program, you do not need to configure the container.

Copying the ISR-WAAS OVA Package to the Cisco ISR 4451-X

Before you can install ISR-WAAS, you must copy the OVA package to the Cisco ISR 4451-X.

Restriction

- You can only deploy one instance of the ISR-WAAS application on a Cisco ISR 4451-X.

Use the following CLI command to copy the ISR-WAAS OVA package to the Cisco ISR 4451-X:

```
router# copy tftp: harddisk:
Address or name of remote host []? 1.1.220.10
Source filename []? ISR4451-X-WAAS-eft.ova
Destination filename [ISR4451-X-WAAS-eft.ova]?
Accessing tftp://1.1.220.10/ISR4451-X-WAAS-eft.ova...
Loading ISR4451-X-WAAS-eft.ova from 1.1.220.10 (via GigabitEthernet0):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 1331268265/4096 bytes]

1331268265 bytes copied in 165.024 secs (8067119 bytes/sec)
```

Installing the ISR-WAAS OVA Package

Use the **virtual-service install name** *name* **package** *package* command to install the ISR-WAAS OVA package onto the Cisco ISR 4451-X.

Notes

- It can take two to three minutes to get a package installed successfully.
- Ensure that the output of the **show virtual-service list** command shows the status as "installed" before proceeding further.

```
router# virtual-service install name ISR4451-X-WAAS package
harddisk:ISR4451-X-WAAS-eft.ova
Package "harddisk:/ISR4451-X-WAAS-eft.ova" is currently being installed for virtual
service "ISR-WAAS". Once the install is finished, please activate the VM to run the VM.

router# show virtual-service list
System busy installing virtual-service 'ISR-WAAS'. The request may take several minutes...
Virtual Service List:

Name                     Status              Package Name
------------------------------------------------------------------------
ISR-WAAS                    Installing         ISR4451-X-WAAS-eft.ova

*Sep 16 00:55:07.588: %IOSXE_VMAN-3-RSPMSGHDLR: Failed to deliver response message:
License Register fails

router# show virtual-service list
Virtual Service List:

Name                     Status              Package Name
------------------------------------------------------------------------
ISR-WAAS                    Installed          ISR4451-X-WAAS-eft.ova
```

Here is the output of the **show virtual-service detail** command at this stage after installation.

```
router# show virtual-service detail
Virtual Service AUTOWAAS Detail:

Package metadata:
Package name          : ISR4451-X-WAAS-eft.ova
Application name      : ISR-WAAS
Application version   : 1.0
Application description : WAAS
Certificate type      : N/A
Signing method        : SHA512
Licensing name        : V-WAAS
Licensing version     : 1.0
OVA path              : /vol/harddisk//ISR4451-X-WAAS-eft.ova
State                 : Activated
Detailed guest status :

Activated profile name: ISR-WAAS-750
Disk reservation      : 270784 MB
Memory reservation    : 4096 MB
CPU reservation       : 0% system CPU
VCPUs                 : 2

Attached devices:
Type              Name        Alias
---------------------------------------------
HDD               vdc
```

```
HDD               vdb
HDD               vda
Serial/Trace                  serial3
Serial/Syslog                 serial2
Serial/aux                    serial1
Serial/shell                  serial0
NIC               ieobc_2     ieobc
NIC               dp_2_31     net2

Network interfaces:
MAC address             Attached to interface
------------------------------------------------------
54:0E:00:0B:0C:03       ieobc_2
30:F7:0D:53:C6:1F       VirtualPortGroup31

Guest interface:
Interface: eth0
  ip address: 33.1.1.2/24

Guest routes:
Address/Mask                          Next Hop                          Intf.
------------------------------------------------------------------------------
0.0.0.0/0                             33.1.1.1                          eth0


Resource admission (without profile) : passed
 Disk space    :
 Memory        : 3072MB
 CPU           : Not specified
 VCPUs         : 1
```

Configuring the Virtual Port Group Interface

Configure a virtual port group interface. Use the IP address of the host end of the bridge between the host and virtual service when activated. Here is an example of a part of the running config output:

```
interface VirtualPortGroup4
 ip address 33.1.1.1 255.255.255.0
```

Listing and Selecting a Profile

The output of the **show virtual-service profile name ISR-WAAS** command displays the various profiles that are available in the installed ISR-WAAS OVA package. Using the **detail** keyword with this command displays the resource requirements associated with the profile description.

```
router# show virtual-service profile name ISR-WAAS
Virtual Service ISR-WAAS profiles:

Name                    Description                                        Allowed
-------------------------------------------------------------------------------------
ISR-WAAS-750               ISR-WAAS profile for 750 TCP connections              Yes
ISR-WAAS-200               ISR-WAAS profile for 200 TCP connections              Yes

router# show virtual-service profile name ISR-WAAS detail
Virtual Service ISR-WAAS Profile Details:

Profile name : ISR-WAAS-750
Description : ISR-WAAS profile for 750 TCP connections
```

```
License name : V-WAAS
License version : 1.0
Resource admission : Yes
Resource requirements :
   Disk space    : Not specified
   Memory        : 4096MB
   CPU           : Not specified
   VCPUs         : 2

Profile name : ISR-WAAS-200
Description : ISR-WAAS profile for 200 TCP connections

License name : V-WAAS
License version : 1.0
Resource admission : Yes
Resource requirements :
   Disk space    : Not specified
   Memory        : 2048MB
   CPU           : Not specified
   VCPUs         : 1
```

Creating the ISR-WAAS Application Container

To create the ISR-WAAS application container, associate the virtual service with a profile name and a virtual port group. Configure the IP address of the virtual service end of the bridge that is created between the host and the virtual service when activated. Here is an example:

```
router(config)# virtual-service ISR-WAAS
router(config-virt-serv)# profile ISR-WAAS-750
router(config-virt-serv)# interface VirtualPortGroup31
router(config-virt-serv-intf)# ip address 33.1.1.2
```

Activating the ISR-WAAS Application Container

The following is an example of the commands used to activate the ISR-WAAS application container:

Note Ensure that the output of the **show virtual-service list** command displays the status of the virtual service as “Activated” to confirm that the service has started successfully.

```
router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

router(config)# virtual-service ISR-WAAS

router(config-virt-serv)# activate

router(config-virt-serv)# end

router# show virtual-service list
System busy activating virtual-service 'ISR-WAAS'. The request may take several minutes...
Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ISR-WAAS                   Activating         ISR4451-X-WAAS-eft.ova

router#
```

```
*Sep 16 01:04:06.196: %VIRT_SERVICE-5-ACTIVATION_STATE: Successfully activated virtual
service ISR-WAAS

router#
*Sep 16 01:04:08.196: %LINK-3-UPDOWN: Interface VirtualPortGroup4, changed state to up
*Sep 16 01:04:09.197: %LINEPROTO-5-UPDOWN: Line protocol on Interface VirtualPortGroup4,
changed state to up

router# show virtual-service list
Virtual Service List:

Name                   Status             Package Name
------------------------------------------------------------------------
ISR-WAAS                  Activated     ISR4451-X-WAAS-eft.ova
```

Verifying the ISR-WAAS Application Container Activation

Issue the following command to verify that the ISR-WAAS application container activated correctly:

```
router# show virtual-service detail name ISR-WAAS
Virtual Service ISR-WAAS Detail:

Package metadata:
Package name          : ISR4451-X-WAAS-eft.ova
Application name      : ISR-WAAS
Application version   : 1.0
Application description : WAAS
Certificate type      : N/A
Signing method        : SHA512
Licensing name        : V-WAAS
Licensing version     : 1.0
OVA path              : /vol/harddisk//ISR4451-X-WAAS-eft.ova
State                 : Activated
Detailed guest status :
  Request failed
Activated profile name: ISR-WAAS-750
Disk reservation      : 270784 MB
Memory reservation    : 4096 MB
CPU reservation       : 0% system CPU
VCPUs                 : 2

Attached devices:
Type              Name        Alias
---------------------------------------------
HDD               vdc
HDD               vdb
HDD               vda
Serial/Trace                  serial3
Serial/Syslog                 serial2
Serial/aux                    serial1
Serial/shell                  serial0
NIC               ieobc_1     ieobc
NIC               dp_1_4      net2

Network interfaces:
MAC address             Attached to interface
-----------------------------------------------------
54:0E:00:0B:0C:02       ieobc_1
30:F7:0D:53:C6:1F       VirtualPortGroup4

Guest interface:
Interface: eth0
```

```
  ip address: 33.1.1.1 /24

Guest routes:
Address/Mask                          Next Hop                          Intf.
------------------------------------------------------------------------------
0.0.0.0/0                             33.1.1.2                           eth0


Resource admission (without profile) : passed
 Disk space    :
 Memory        : 3072MB
 CPU           : Not specified
 VCPUs         : 1
```

Stopping the ISR-WAAS Application

To stop the ISR-WAAS application, enter the following commands and then issue the **virtual-service list** command to ensure that the application is deactivated:

```
router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

router(config)# virtual-service ISR-WAAS
router(config-virt-serv)# no activate
router(config-virt-serv)# end

router# show virtual-service list
Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ISR-WAAS                   Deactivated        ISR4451-X-WAAS-eft.ova
```

Uninstalling the ISR-WAAS Application

When you uninstall the ISR-WAAS application, the system releases all the disk storage that was reserved for this virtual service and any associated saved data is lost.

The following commands demonstrate how to uninstall the ISR-WAAS application. Use the **no activate** command to stop the virtual service before you uninstall it.

Note Stopping the ISR-WAAS application can take some time. The **show virtual-service list** command will show the status as “Deactivating” when the application is de-activating. Ensure that the application is deactivated before you uninstall the application

```
router(config)# virtual-service ISR-WAAS
router(config-virt-serv)# no activate

router# virtual-service uninstall name ISR-WAAS
router#
*Sep 16 01:25:44.996: %VIRT_SERVICE-5-INSTALL_STATE: Successfully uninstalled virtual
service ISR-WAAS
router#

router# show virtual-service list
Virtual Service List:
```

Removing the AppNav-XE Configuration

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

Procedure

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

```
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:

```
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:

```
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:

```
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
```

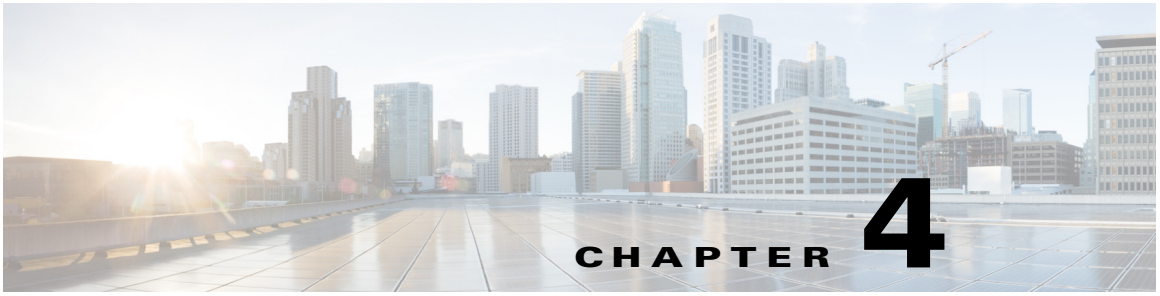
Configuring Port Channel Support for AppNav-XE

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.



Monitoring the AppNav-XE and ISR-WAAS Components

This chapter describes how to monitor the AppNav-XE and ISR-WAAS components and contains the following sections:

AppNav Controller Show Commands

You can use **show** commands to check status and display data.

Checking the Status of the AppNav Controller

Use the following command to check on the general status of the AppNav Controller. The command also lists all the interfaces that have “service-insertion waas” configured.

```
router# show service-insertion status

Hostname: Branch-router
Device ID:30f7.0d54.5510
Platform Type:cisco (ISR4452/K9) 2RU
IOS Version: 15.3(20130102:194350)
AppNav Controller Version: 1.0.0
AppNav Enabled Interfaces:
GigabitEthernet0/0/1
```

Checking the Membership of the AppNav Controller Group

Use the following command to check the membership of the AppNav Controller group. It also lists all the service nodes configured and registered with the AppNav Controller.

```
router# show service-insertion appnav-controller-group
All AppNav Controller Groups in service context
Appnav Controller Group                          : acg
Member Appnav Controller Count                   : 2
Members:
      IP Address
        21.0.0.36
        21.0.0.160

AppNav Controller                                : 21.0.0.36
Local AppNav Controller                     : Yes
Current status of AppNav Controller              : Alive
Time current status was reached                  : Wed Sep  5 15:50:06 2012
Cluster protocol ICIMP version                   : 1.1
Cluster protocol Incarnation Number              : 1
Cluster protocol Last Sent Sequence Number       : 0
Cluster protocol Last Received Sequence Number : 0
Current AC View of AppNav Controller
      IP Address
        21.0.0.36
        21.0.0.160

Current SN View of AppNav Controller
      IP Address
      21.0.0.149

AppNav Controller                                  : 21.0.0.160
Local AppNav Controller                            : No
Current status of AppNav Controller                : Alive
Time current status was reached                    : Thu Dec 6 20:17:53 2012
Cluster protocol ICIMP version                     : 1.1
Cluster protocol Incarnation Number                : 1
Cluster protocol Last Sent Sequence Number         : 1355098374
Cluster protocol Last Received Sequence Number     : 1355089899

Current AC View of AppNav Controller
    IP Address
    21.0.0.36
    21.0.0.160

Current SN View of AppNav Controller
    IP Address
    21.0.0.149
```

Displaying Detailed Information About Service Node Groups and Service Nodes

Use the **show service-insertion service-node-group [***sng_name* **| all]** command to display detailed information about service node groups and individual service nodes. You can also use this command to check the status of individual application accelerators.

The output of this command shows the following:

- Cluster protocol information. The *last sent sequence number* and the *last received sequence number* values should be increasing continuously.

- Number of service nodes and associated service contexts.
- Status of each service node, which can be either Alive or Dead
- Load state, which displays the health of the application accelerators. The load state can be one of the following:
  - green—application accelerator is functional and accepting new flows
  - yellow—application accelerator is functional but not accepting new flows
  - red—application accelerator is not functional
- Overall availability of the service node group for each application accelerator

```
router# show service-insertion service-node-group
Service Node Group name :sng1
    Service Context :             waas/1
    Member Service Node count :   1

Service Node (SN) :                   21.0.0.149
Auto discovered :                     No
SN belongs to SNG :                   sng1
Current status of SN :                Alive
Time current status was reached :     Thu Dec 6 20:17:11 2012

Cluster protocol DMP version :                        1.1
Cluster protocol incarnation number :                 2
Cluster protocol last sent sequence number :          1355101043
Cluster protocol last received sequence number:       1348909100

Health Markers:
       AO          Load State           Since
       tcp         GREEN               0d 5h 39m 38s
       epm         GREEN               0d 5h 39m 38s
       cifs        GREEN               0d 5h 39m 38s
       mapi        GREEN               0d 5h 39m 38s
       http        GREEN               0d 5h 39m 38s
       video       GREEN               0d 5h 39m 38s
       nfs         GREEN               0d 5h 39m 38s
       ssl         YELLOW              0d 5h 39m 38s
       ica         RED                 0d 0h 0m 0s

SNG Availability per Accelerator
       AO          Available           Since
       tcp         Yes                 0d 5h 39m 38s
       epm         Yes                 0d 5h 39m 38s
       cifs        Yes                 0d 5h 39m 38s
       mapi        Yes                 0d 5h 39m 38s
       http        Yes                 0d 5h 39m 38s
       video       Yes                 0d 5h 39m 38s
       nfs         Yes                 0d 5h 39m 38s
       ssl         No                  0d 0h 0m 0s
       ica         No                  0d 0h 0m 0s
```

Displaying Class Maps and Policy Maps

The following commands reflect the running configuration and are useful for checking classifications without having to scan through an entire running configuration.

To display all type AppNav class maps and their matching criteria, or a specific AppNav class map and its matching criteria, use the following command:

```
router# show class-map type appnav [AppNav_class_name]
```

To display all type AppNav policy maps and their class and action mappings, or a specified policy map and its class or action mappings, use the following command:

```
router# show policy-map type appnav [AppNav_policy_name]
```

The **show policy-map target service-context [***service_context_name***]** command displays policy map information for service contexts. Use this command to view the flow level stats of all the class maps and policy maps that are configured under a service context. If you do not specify a service context name, the command displays all the configured class maps and policy maps.

Here are two examples:

```
router# show policy-map target service-context waas/1
Service-policy appnav input: p1

      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
        Aggregate: 0 packets, 0 bytes
       monitor-load http
      Class-map: class-default (match-any)
         Match: any

router# show policy-map target service-context
    Service-policy appnav input: p1
      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
       monitor-load http

      Class-map: class-default (match-any)
         Match: any

    Service-policy appnav input: p3

    Class-map: c3 (match-all)
      Match: access-group 101
      distribute service-node-group sng3
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes

    Class-map: class-default (match-any)
       Match: any
```

Displaying Service Context Information

To display information about service contexts, use the **show service-insertion service-context [***service_context_name***]** command. The output of this command displays the status of the specified service context, including the following:

- Current and last states of the Cluster Membership Manager (CMM) and FSM
- State of the cluster
- Views of the stable and current AppNav Controller and service nodes

Here is an example:

```
router# show service-insertion service-context waas/1

Service Context                              : waas/1
Cluster protocol ICIMP version               : 1.1
Cluster protocol DMP version                 : 1.1
Time service context was enabled             : Thu Sep  8 08:38:41 2011
Current FSM state                            : Operational
Time FSM entered current state               : Thu Sep  8 08:48:26 2011
Last FSM state                               : Converging
Time FSM entered last state                  : Thu Sep  8 08:48:16 2011
Cluster operational state                    : Operational

Stable AppNav Controller View:
        2.58.2.40
Stable SN View:
        2.43.139.170     2.58.2.40
Current AppNav Controller View:
        2.58.2.40
Current SN View:
        2.43.139.170     2.58.2.40
```

Displaying Data Path Statistics

Displaying AppNav Controller Group Statistics

To see the number of “keepalives” sent to the other AppNav Controllers and received from the other AppNav Controllers and other statistics related to the AppNav Controller group, use the following command:

```
router# show service-insertion statistics appnav-controller-group
Appnav Controller Group              : acg
Number of AppNav Controllers         : 2
Members:
    IP Address
    21.0.0.36
    21.0.0.160

Aggregate Appnav Controller statistics
--------------------------------------
Time since statistics were last reset/cleared    : 0d 5h 47m 14s

Aggregate number of keepalives sent to ACs       : 168484
Aggregate number of keepalives received from ACs : 166372
Aggregate number of invalid keepalives received :
           Total                                 : 0
           Incompatible ICIMP version            : 0
```

```
            Authentication Failed                    : 0
            Stale keepalive                          : 0
            Malformed keepalive                      : 0
            Unknown keepalive                        : 0
            Inactive keepalive                       : 0
Aggregate number of times liveliness lost with ACs : 1
Aggregate number of times liveliness gained with ACs: 2
```

Displaying Per Service Node and Service Node Group Statistics

To show the connections, packets, and bytes sent to each service node, use the following command:

```
router# show service-insertion statistics service-node [IP_address]
```

To show the aggregated connections, packets, and bytes sent to each service node group, use this command:

```
router# show service-insertion statistics service-node-group [NAME]
```

Here is an example:

```
router# show service-insertion statistics service-node

Statistics for Service Node 21.0.0.149
------------------------------------------------------
Time since statistics were last reset/cleared: 0d 18h 7m 54s
Number of probe requests sent to SN: 326024
Number of probe responses received from SN: 326014
Number of invalid probe responses received:
    Total   : 0
    Incompatible DMP version: 0
    Authentication failed: 0
    Stale response: 0
    Malformed response: 0
    Unknown response: 0
Number of times liveliness lost with SN: 0
Number of times liveliness regained with SN:1

Cluster IPC statistics
----------------------
Time since statistics were last reset/cleared: 0d 18h 8m 24s
Number of load updates received from CMM: 4
Number of erroneous load updates: 0
Time since last load update was received: 0d 14h 32m 43s

Load stats for Service Node 21.0.0.149
------------------------------------------

Accelerator state transition statistics
---------------------------------------
Time since Accl load stats were last cleared: 0d 18h 8m 24s
Accl    Current    Previous   Red     Yellow     Green
tcp     GREEN      RED        0       0          1
epm     GREEN      RED        0       0          1
cifs    GREEN      RED        0       0          1
mapi    GREEN      RED        0       0          1
http    GREEN      RED        0       0          1
video   GREEN      RED        0       0          1
nfs     GREEN      RED        0       0          1
ssl     YELLOW     RED        0       1          0
ica     RED        RED        0       0          0

Traffic distribution statistics for service node 21.0.0.149
```

```
-------------------------------------------------------
Time since distribution stats were last cleared: 0d 18h 8m 24s

Packet and byte counts
----------------------
Redirected Bytes: 2338
Redirected Packets: 50
Received Bytes: 3350
Received Packets: 50

Occurences
-----------
Initial Redirects: 2
Initial Redirects Accepted: 2
Initial Redirect -> Passthrough: 0
Redirect -> Passthrough: 0
```

The important statistics are as follows:

- Probe Requests: The number of heartbeats sent to the service node.
- Probe Responses: The number of heartbeats received from the service node.
- Redirected Bytes: The number of bytes redirected to the service node.
- Redirected Packets: The number of data packets redirected to the service node.
- Received Bytes: The number of bytes received from the service node.
- Received Packets: The number of data packets received from the service node.
- Initial Redirects: The number of times that the SYN packet (the first packet for requesting connection in a TCP flow) was redirected to the service node.
- Initial Redirects Accepted: The number of times that the service node decided to optimize on SYN packet.
- Initial Redirects -> Passthrough: The number of times that the service node decided to pass-through on SYN packet.
- Redirect -> Passthrough: The number of times that the service node decided to pass-through a flow after it was initially accepted for optimize (e.g. due to lack of peer).

Displaying Service Context Statistics

To display statistics about the service context, use the **show service-insertion statistics service-context** [*name*] command. The output of this command displays the time spent in each FSM state by the CMM and the amount of time that each service context has been in each FSM state.

Here is an example:

```
Router# show service-insertion statistics service-context
Time spent in various FSM states

Converging     :       0d 0h 0m 31s
Initializing   :       0d 0h 0m 0s
Operational    :       1d 19h 27m 53s
Degraded       :       0d 0h 0m 0s
Internal Error :       0d 0h 0m 0s
Admin Disabled :       0d 0h 0m 0s

Number of entries into Converging State:      3
Number of entries into Initializing State:    1
Number of entries into Operational State:     3
Number of entries into Degraded State:        0
```

```
Number of entries into Internal Error State:  0
Number of entries into Admin Disabled State:  0
```

Displaying Flow Statistics

To query the flows in the flow table and to optionally filter the output by using specific criteria, use the following command:

```
router# show service-insertion statistics connection [[summary] | [vrf-name name]
[client-ip IP_address] [client-port port_number] [server-ip IP_address] [server-port
port_number] [detail]]
```

As part of the flow query, the following information for every flow is available:

- Client IP address, client TCP port and server IP address, server TCP port number
- Service node IP address, passthrough
- VRF name

Here is an example:

```
router# show service-insertion statistics connection
Collecting Records. Please wait...
Client                Server                SN-IP       VRF-Name
51.0.222.4:64234      11.0.0.3:80           21.0.0.104  br_vrf
51.0.222.4:22415      11.0.0.3:80           21.0.0.104
51.0.222.4:15264      11.0.0.3:80           21.0.0.104
51.0.222.4:37759      11.0.0.3:80           21.0.0.104
51.0.222.4:55408      11.0.11.2:23          Passthrou
```

If you include the *detail* keyword, the report also displays the following on a per flow basis:

- Presence of session (3T) or App (2T) association
- Application ID
- Peer ID

The following is an example:

```
router# show service-insertion statistics connection detail
Collecting Records. Please wait...

Client: 192.168.80.4:60973
Server: 192.168.180.4:135
Service Node IP: 172.16.0.2
Flow association: 2T:No,3T:No
VRF-Name:
Application ID: 0
Peer-ID: 00:21:5e:76:65:08

Client: 192.168.80.4:60959
Server: 192.168.180.4:1092
Service Node IP: 172.16.0.2
Flow association: 2T:Yes,3T:Yes
VRF-Name:
Application ID: 78
Peer-ID: 00:21:5e:76:65:08
```

If you include the *summary* keyword, the report displays only the number of 2T and 3T entries, the number of optimized flows, the number of passthrough flows, and the number of flow synchronization failures due to VRF config mismatch on the AppNav Controllers.

The following is an example:

```
router# show service-insertion statistics connection summary
Number of 2T optimized flows    = 0
Number of 3T optimized flows    = 0
Number of optimized flows       = 3
Number of pass-through flows    = 1
Flow sync failures due to vrf mismatch = 0
```

You can also use the **show platform software** command. It works exactly the same as the **show service-insertion statistics** command, but it can also be used to query the flows on the standby FP.

```
router# show platform software appnav-controller <f0 | f1 | fp active | fp standby>
connections …
```

Displaying Application and Session Statistics

To query the application and session entries and to optionally filter the output by using specific criteria, use the following command:

```
router# show service-insertion statistics sessions [[vrf-name name] [client-ip
IP_address][server-ip IP_address] [server-port port_number] [detail]]
```

Application entries do not have client or service node IP addresses.

Here is an example:

```
router# show service-insertion statistics sessions

Collecting Records. Please wait...
Client                Server                SN-IP      VRF-Name
N/A                   192.168.180.4:1092    N/A
192.168.80.4:0        192.168.180.4:1092    172.16.0.2
```

If you include the *detail* keyword, the report also displays the application ID and the time since the last activity.

Here is an example:

```
Router# show service-insertion statistics sessions detail
Collecting Records. Please wait...
Client: 192.168.80.4:0
Server: 192.168.180.4:1098
Service Node IP: 172.16.0.2
VRF-Name:
Application ID: 78
Time since last activity : 0hr 36min 30sec

Client: N/A
Server: 192.168.180.4:1098
Service Node IP: N/A
VRF-Name:
Application ID: 78
Time since last activity : 0hr 36min 30sec
```

You can also use the **show platform software** command. It works exactly the same as the **show service-insertion statistics** command, but it can also be used to query the application and session entries on the standby FP.

```
router# show platform software appnav-controller <f0 | f1 | fp active | fp standby>
sessions …
```

Displaying Classification Statistics

Use the **show policy-map target service-context [***service_context_name***]** command to view the flow level statistics of all the class maps and policy maps that are configured under a service context. If you do not enter a service context name, the system displays all the configured class maps and policy map output.

The following are examples:

```
router# show policy-map target service-context waas/1

Service-policy appnav input: p1
Class-map: c1 (match-all)
      Match: access-group 101
      distribute service-node-group sng
        Distributed: 313450 packets, 135820480 bytes
        Passed through: 0 packets, 0 bytes
       Aggregate: 313450 packets, 135820480 bytes

    monitor-load http

    Class-map: c2 (match-all)
      Match: access-group 102
    Pass-through
        Distributed: 0 packets, 0 bytes
        Passed through: 40 packets, 30000 bytes
       Aggregate: 40 packets, 30000 bytes

    Class-map: class-default (match-any)
      Match: any

router# show policy-map target service-context

Service-policy appnav input: p1
      Class-map: c1 (match-all)
         Match: access-group 101
       distribute service-node-group sng1
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes
       monitor-load http

      Class-map: class-default (match-any)
         Match: any

    Service-policy appnav input: p3
Class-map: c3 (match-all)
      Match: access-group 101
      distribute service-node-group sng3
         Distributed: 0 packets, 0 bytes
         Passed through: 0 packets, 0 bytes
       Aggregate: 0 packets, 0 bytes

    Class-map: class-default (match-any)
       Match: any
```

Displaying Pass Through Reason Statistics

To view the passthrough reason statistics aggregated for all the classes of a policy associated with the specified service context, use the following command:

```
router# show policy-map target service-context context_name passthru-reason
```

To view the passthrough reason statistics for a particular class of a policy associated with the specified service context, use the following command:

```
router# show policy-map target service-context context_name class class_name
passthru-reason
```

Here is an example:

```
router# show policy-map target service-context waas/1 class c4 passthru-reason

Service-policy appnav input: p4

Class-map: c4 (match-all)
    Match: access-group 101
    distribute service-node-group sng4
        Distributed: 11 packets, 222 bytes
        Passed through: 100 packets, 22000 bytes
    Aggregate: 111 packets, 22222 bytes
Collected by SC:

Passthrough Reasons               Packets           Bytes
------------------------          -------  -        ------
PT Flow Learn Failure                      0        0
PT SNG Overload                            0        0
PT Appnav Policy                           0        0
PT Cluster Degrade                         0        0
PT ZBFW                                    0        0
PT NAT ALG                                 0        0
PT Unknown                                 0        0

Indicated by SN:

Passthrough Reasons               Packet             Bytes
------------------------          -------  -         ------
PT No Peer                                 100      22000
PT Rjct Capabilities                       0        0
PT Rjct Resources                          0        0
PT Rjct No License                         0        0
PT App Config                              0        0
PT Global Config                           0        0
PT Asymmetric                              0        0
PT In Progress                             0        0
PT Intermediate                            0        0
PT Overload                                0        0
PT Internal Error                          0        0
PT App Override                            0        0
PT Server Black List                       0        0
PT AD Version Mismatch                     0        0
PT AD AO Incompatible                      0        0
PT AD AOIM Progress                        0        0
PT DM Version Mismatch                     0        0
PT Peer Override                           0        0
PT Bad AD Options                          0        0
PT Non-optimizing Peer                     0        0
PT SN Interception ACL                     0        0
PT IP Fragment Unsupported                 0        0
                                  -----------       -------
PT Overall                                 100      22000
```

Displaying Alarms

Use the following command to display the alarms seen on the AppNav Controller. The **detail** option gives a brief explanation of each alarm and the **support** option gives a longer explanation along with a recommended action.

```
router# show service-insertion alarms [critical | major | minor] [detail [support]]
```

The following is an example:

```
router# show service-insertion alarms detail

Critical Alarms:
----------------
    Alarm Instance     Alm ID  Module     AC/SN IP Addr  AO      SNG
    1 degraded_cluster 29002   cmm        N/A            N/A     N/A

    Cluster protocol detected inconsistency in AC view of peer ACs. Device will
pass-through all new connections.

Major Alarms:
-------------
    Alarm Instance     Alm ID  Module     AC/SN IP Addr  AO      SNG
    1 ac_unreachable   29006   cmm        192.168.1.11   N/A     N/A

    Cluster protocol on device cannot communicate with peer AC ("192.168.1.11").

    2 sn_unreachable   29007   cmm        192.168.2.31   N/A     N/A

    Cluster protocol on device cannot communicate with peer SN ("192.168.2.31").

    3 sng_unavailable  30001   fdm        N/A            N/A     sng1

    Service Node Group ("sng1") has become unavailable.

    4 sng_ao_unavailable30000 fdm        N/A            sslsng

    Service Node Group ("sng") has become unavailable for accelerator - ("ssl").

Minor Alarms:
-------------
None
```

AppNav Service Node Auto Discovery Show Commands

Use the following commands to show information about the AppNav service node auto discovery feature.

show service-insertion service-node-group *sng_name*

```
router# show service-insertion service-node-group sng

Service Node Group name : sng
Service Context : waas/1
Member Service Node count : 2

Service Node (SN) : 20.20.20.20
Auto discovered : Yes
SN belongs to SNG : sng
Current status of SN : Alive
```

```
Time current status was reached : Thu Dec 6 00:51:48 2012

Cluster protocol DMP version : 1.1
Cluster protocol incarnation number : 13
Cluster protocol last sent sequence number : 1355026900
Cluster protocol last received sequence number: 131214317

Health Markers:
       AO          Load State         Since
       tcp         YELLOW             0d 14h 3m 53s
       epm         RED                0d 0h 0m 0s
       cifs        RED                0d 0h 0m 0s
       mapi        RED                0d 0h 0m 0s
       http        RED                0d 0h 0m 0s
       video       RED                0d 0h 0m 0s
       nfs         RED                0d 0h 0m 0s
       ssl         RED                0d 0h 0m 0s
       ica         RED                0d 0h 0m 0s

Service Node (SN) : 1.2.3.4
Auto discovered : No
SN belongs to SNG : sng
Current status of SN : Dead
Time current status was reached : Thu Dec 6 14:19:52 2012

Cluster protocol DMP version : 0.0
Cluster protocol incarnation number : 0
Cluster protocol last sent sequence number : 1355026901
Cluster protocol last received sequence number: 0

Health Markers:
       AO          Load State         Since
       tcp         RED                0d 0h 0m 0s
       epm         RED                0d 0h 0m 0s
       cifs        RED                0d 0h 0m 0s
       mapi        RED                0d 0h 0m 0s
       http        RED                0d 0h 0m 0s
       video       RED                0d 0h 0m 0s
       nfs         RED                0d 0h 0m 0s
       ssl         RED                0d 0h 0m 0s
       ica         RED                0d 0h 0m 0s

SNG Availability per Accelerator
       AO          Available          Since
       tcp         No                 0d 0h 0m 0s
       epm         No                 0d 0h 0m 0s
       cifs        No                 0d 0h 0m 0s
       mapi        No                 0d 0h 0m 0s
       http        No                 0d 0h 0m 0s
       video       No                 0d 0h 0m 0s
       nfs         No                 0d 0h 0m 0s
       ssl         No                 0d 0h 0m 0s
       ica         No                 0d 0h 0m 0s
```

show service-insertion service-node-group *sng_name* auto-discovered

```
router# show service-insertion service-node-group sng auto-discovered

    MAC Address       Resp Elapsed Minutes    IP Address
    50:57:a8:e1:af:1  0                       20.20.20.20
```

show mdns req

```
router# show mdns request
```

```
MDNS Outstanding Requests
==================================================
Request name : _appnav_waas_node._udp.local
Request type : PTR
Request class : IN
```

show mdns stat

```
router# show mdns stat

mDNS Statistics
mDNS packets sent : 852
mDNS packets received : 510
mDNS packets dropped : 0
```

Container Show Commands

Displaying Virtual Service Information

The **show virtual-service** CLI command provides details about the running application, the profiles supported, storage used by the application, and CPU utilization. See the following example:

```
router# show virtual-service ?
  detail       Detail information about appliance
  list         List the appliance
  profile      information about appliance profile
  storage      Storage information about appliance
  utilization  Utilization information about appliance
  version      Version information about appliance
  |            Output modifiers
```

Displaying Details for a Virtual Service

Use the following CLI command to display details for a virtual service:

```
router# show virtual-service detail
Virtual Service AUTOWAAS Detail:

Package metadata:
Package name          : ISR4451-X-WAAS-eft.ova
Application name      : ISR-WAAS
Application version   : 1.0
Application description : WAAS
Certificate type      : N/A
Signing method        : SHA512
Licensing name        : V-WAAS
Licensing version     : 1.0
```

```
OVA path              : /vol/harddisk//ISR4451-X-WAAS-eft.ova
State                 : Activated
Detailed guest status :

Activated profile name: ISR-WAAS-750
Disk reservation      : 270784 MB
Memory reservation    : 4096 MB
CPU reservation       : 0% system CPU
VCPUs                 : 2

Attached devices:
Type              Name        Alias
---------------------------------------------
HDD               vdc
HDD               vdb
HDD               vda
Serial/Trace                  serial3
Serial/Syslog                 serial2
Serial/aux                    serial1
Serial/shell                  serial0
NIC               ieobc_2     ieobc
NIC               dp_2_31     net2

Network interfaces:
MAC address             Attached to interface
-----------------------------------------------------
54:0E:00:0B:0C:03       ieobc_2
30:F7:0D:53:C6:1F       VirtualPortGroup31

Guest interface:
Interface: eth0
  ip address: 33.1.1.2/24

Guest routes:
Address/Mask                          Next Hop                          Intf.
-------------------------------------------------------------------------------
0.0.0.0/0                             33.1.1.1                          eth0


Resource admission (without profile) : passed
 Disk space    :
 Memory        : 3072MB
 CPU           : Not specified
 VCPUs         : 1
```

Displaying a List of Virtual Services

The container infrastructure provides commands to view the status and details of installed applications. To view the list of existing applications, use the following command:

```
router# show virtual-service list
System busy installing virtual-service 'ISR-WAAS'. The request may take several minutes...
Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ISR-WAAS                   Installing         ISR4451-X-WAAS-eft.ova
```

Displaying Storage Volume Information for a Virtual Service

Use the following CLI command to display storage volume information for a virtual service:

```
router# show virtual-service storage volume list
Virtual-Service storage volume list

Name                      Capacity    In Use      Virtual-Service
--------------------------------------------------------------------------------
vda.AUTOWAAS              4097 MB     Yes         AUTOWAAS
vdb.AUTOWAAS              163841 MB   Yes         AUTOWAAS
```

Displaying Statistics for a Virtual Service

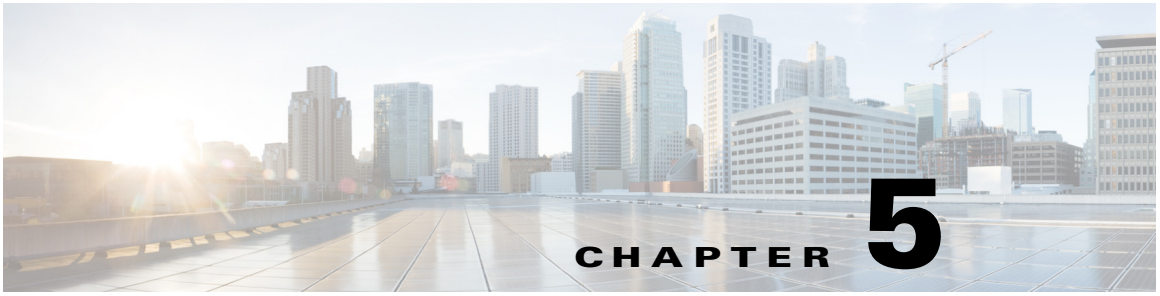
Use the following CLI command to display statistics for a virtual service:

```
router# show virtual-service utilization statistics CPU
--------------------------
/cgroup
--------------------------
 directory  share system % User % Num CPUs CPU core %
 .           1024 70.83     29.17  8         9.57 14.59 7.87 23.65 7.39 17.06 6.51 13.35
./libvirt   8192 75.75     24.25  8        18.05 28.26 11.68 10.70 10.38 9.51 6.06 5.37
./iosbinos  1024 69.21     30.79  8        5.61 10.12 6.53 25.90 6.37 21.43 7.35 16.68

. ./libvirt ./iosbinos
61.33 18.22 20.45
--------------------------
/cgroup/libvirt
--------------------------
 directory  share system % User % Num CPUs CPU core %
 .           8192 75.75     24.25  8         18.05 28.26 11.68 10.70 10.38 9.51 6.06 5.37
./AUTOWAAS  1024 75.75     24.25  8        18.05 28.26 11.68 10.70 10.38 9.51 6.06 5.37

. ./AUTOWAAS
-0.00 100.00
--------------------------
/cgroup/libvirt/AUTOWAAS
--------------------------
 directory  share system % User % Num CPUs CPU core %
 .           1024 75.75     24.25  8         18.05 28.26 11.68 10.70 10.38 9.51 6.06 5.37

.
100.00
--------------------------
/cgroup/iosbinos
--------------------------
 directory  share system % User % Num CPUs CPU core %
 .           1024 69.21     30.79  8         5.61 10.12 6.53 25.90 6.37 21.43 7.35 16.68
100.00
```



Troubleshooting

This chapter describes some common problems and how to troubleshoot them and contains the following sections:

Using Debug Commands

AppNav-XE Debug Commands

Clearing AppNav-XE Statistics

To clear all the AppNav-XE statistics or just certain statistics, use the following command:

```
router# clear service-insertion statistics ?

all                       Clear all service-insertion statistics
appnav-controller         Clear appnav-controller statistics
appnav-controller-group   Clear appnav-controller-group statistics
service-context           Clear service-context statistics
service-node              Clear service-node statistics
service-node-group        Clear service-node-group statistics
```

Debugging the Cisco IOS-XE Control Plane

Use the following debug commands to trace control plane activities:

```
router# debug appnav-controller ?
   auto-discovery Debugging AppNav Controller service node auto discovery feature
   cm-reg         Debugging AppNav Controller CM registration with the WCM server
   cmm            Debugging AppNav Controller Cluster Management (CMM)
   fdm            Debugging AppNav Controller Flow Distribution Module (FDM)
   ha             Enable AppNav Controller high availability (HA) redundancy
                  checkpoint and ISSU infrastructure debugs
   vi             Debugging AppNav Controller Virtual Interface (VI), including the
                  status at the time of creation and links to the compress and
                  uncompress interface

router# debug appnav-controller cmm ?
  all      Enable all CMM debugs
  cli      Enable CMM CLI debugs
  events   Enable CMM state machine event debugs
  misc     Enable CMM misc debugs
  packets  Reception and transmission of packets (can be filtered based on IP address)
  shell    Enable CMM misc debugs
  timers   Enable CMM misc debugs

router# debug appnav-controller fdm ?
  all     Enable all FDM debugs
  events  Enable debugging for important events being handled by FDM
  infra   Enable debugging for FDM infrastructure events
```

The following debug commands are the most useful:

- **debug appnav-controller cmm events**
- **debug appnav-controller fdm events**
- **debug appnav-controller ha**

Debugging the Cisco IOS-XE Infrastructure

This section contains the following information:

Showing Packet Drop Statistics

Use the following command to display unplanned packet drops:

```
router# show platform hardware qfp active statistics drop
-----------------------------------------------------------------------------
Global Drop Stats                          Packets                  Octets
-----------------------------------------------------------------------------
AppNavBadRoute                                38                    2888
Ipv4AclLookupMiss                             42                    3034
Ipv4NoRoute                                 4408                 1293334
UnconfiguredIpv4Fia                           19                    1710
```

The following are the reasons for which packets may drop:

- AppNavInvSNpkt—Malformed or unsupported packet from the service node.

- AppNavInternalErr—Logic error within the AppNav-XE component. Uncommon.
- AppNavBadRoute—A non-AppNav-XE packet appeared at the AppNav-XE virtual or tunnel interface. Very common when routing protocols are enabled.
- AppNavNoTunnel—There is no tunnel facility available for the service node-bound packet.
- AppNavNoSvcCtx—There is no service context matching the flows from the service node.
- AppNavInvFOState—The flow state is no longer valid. This is usually due to changes in the configuration.
- AppNavUnexpctdpkt—The AppNav-XE component did not expect to process more packets because it has been shut down.

Showing Data Path CPU Utilization

To display the data path CPU utilization, use the following command:

```
router# show platform hardware qfp active datapath utilization

    CPP 0: Subdev 0      5 secs       1 min       5 min       60 min
Input: Priority (pps)    0            0           0           0
               (bps)     0            72          88          48

Non-Priority (pps)       226455       225968      198785      72441
             (bps)       1879325304   1875408168  1648044616  599951168

        Total (pps)      226455       225968      198785      72441
              (bps)      1879325304   1875408240  1648044704  599951216

Output: Priority (pps)   229023       228474      245267      90057
                (bps)    1619093520   1641710256  2389617496  949076160

Non-Priority (pps)       209522       208053      293300      104501
                (bps)    180090080    178161632   3124680344  1191566064

            Total (pps)  438545       436527      538567      194558
                  (bps)  1799183600   1819871888  5514297840  2140642224

Processing: Load (pct)   26           26          19          8
```

Showing Data Path Memory Utilization

Use the following command to show statistics about the data path memory use.

Note The value for **In Use DRAM memory** must be less than 90 percent of the value for **Total DRAM memory**; otherwise, the AppNav-XE component stops optimizing new flows.

```
router# show platform hardware qfp active infrastructure exmem statistics

QFP exmem statistics
Type: Name: DRAM, QFP: 0
    Total: 268435456
    InUse: 99933184
    Free: 168502272
    Lowest free water mark: 168502272

Type: Name: IRAM, QFP: 0
    Total: 134217728
    InUse: 8087552
```

```
    Free: 126130176
    Lowest free water mark: 126130176

Type: Name: SRAM, QFP: 0
    Total: 32768
    InUse: 15088
    Free: 17680
    Lowest free water mark: 17680
```

Debugging the Data Plane

The output of the following debug command is displayed as a log file named /tmp/fp/trace/cpp_cp_*Fx*-0.log under the FP shell, where *Fx* is either F0 or F1 depending on the active FP module. You need a shell license to access the FP shell.

If you do not have shell access, you can use the **test platform software trace slot fp act cpp-control-process rotate** command to force the log to flush to bootflash:tracelogs.

```
router# debug platform hardware qfp active feature appnav-controller datapath ?
classify    Debug QFP flow classification such as traces, policy, peer ID, and
               classification action (which service node group)
drop        Enable drop debugging and shows traces of packet drop due to errors
fdl         Debug QFP flow distribution such as selecting a service node within a service
               node group
ha          Debug QFP high availability (HA) and AppNav Controller issues. Shows traces
               related to syncing flows between AppNav Controllers and between active and
               standby FPs
interop     Debug QFP feature interoperations such as FNF, NBAR, and NAT
pkt-path    Debug QFP packet processing and packet interception
proxy       Debug QFP proxy issues related to interface with the control place, such as
               statistics reporting and configuration
```

Each of the above categories (other than **drop**, which has no level) has the following four levels:

- Error—Displays error level debugs and detects potential issues.
- Warn—Displays warnings and errors.
- Info—Displays information, warnings, and errors.
- All—The lowest level of debugging. Displays all debugs.

To limit the number of debug messages, we recommend that you only enable the error debug level first and then slowly reduce the debug level.

You can also use the following command to check on packets dropped by the router. The command lists all the packets that were dropped with a reason. If you see AppNav drop reasons, you can enable the debug drop command to see the actual packet drops inside the trace logs.

```
router# show platform hardware qfp active statistics drop

Global Drop Stats         Packets        Octets
-----------------------------------------------------
The Global drop stats were all zero
```

AppNav Service Node Auto Discovery Debug Commands

Use the following debug commands to trace the AppNav service node auto discovery feature:

- **debug appnav-controller auto-discovery**
- **debug mdns all**

- **debug mdns packet**

Container Debug Commands

The following debug commands are available to debug the application configuration, installation, activation, and status:

- **debug virtual-service virtualPortGroup**
- **debug virtual-service timeout**
- **debug virtual-service messaging**
- **debug virtual-service all**

EZConfig Debug Commands

Use the following commands to debug the operation of the EZConfig program:

- **debug service-integration api**
- **debug service-integration configuration**
- **debug service-integration menu**

Common Problems

Traffic Not Redirected

If traffic is not redirected properly, ensure that "service-insertion waas" is present on interfaces on which the traffic is supposed to be intercepted. Issue the **show service-insertion status** command to verify this.

Traffic Passed Through Instead of Redirected

The **show service-insertion statistics connection** command indicates whether traffic is passed through or redirected. If traffic is passed through instead of being redirected, use the **show policy-map target service-context** *context_name* **passthru-reason** command to find out the reason. For details, see the "Displaying Pass Through Reason Statistics" section on page 4-10.

You can also monitor the service node counters. See the "Displaying Per Service Node and Service Node Group Statistics" section on page 4-6.

The term "Initial Redirect" indicates that flows are being redirected to the service nodes. If the flows are not being redirected to the service nodes, maybe the policy did not cover the traffic type.

The "Initial Redirect -> Passthrough" counter indicates that the service node has decided to pass-through the flow. This is likely due to policies on the service node.

The "Redirect -> Passthrough" counter indicates that the service node later decided to pass-through the flow. This is likely due to lack of a peer WAAS device. Two WAAS devices are needed along the path to optimize a flow.

Traffic Not Optimized

The following can cause traffic to not be optimized:

- One or both disks failed
- Configuration mismatch between the AppNav-XE component and the WAAS policy
- The AppNav-XE component not redirecting the traffic
- A resource is not available
- ISR-WAAS configuration
- ISR-WAAS crashes

Degraded Cluster

If connections are passed through and you are using an AppNav Controller group that has two or more AppNav Controllers, it is possible that the cluster state is degraded instead of operational. This means that the AppNav Controller view is not the same on each of the AppNav Controllers.

To check the cluster state and the stable AppNav Controller view on each of the AppNav Controllers, use the following command:

```
router# show service-insertion service-context

Service Context                       : waas/1
Cluster protocol ICIMP version        : 1.1
Cluster protocol DMP version          : 1.1
Time service context was enabled      : Fri Dec 7 19:28:11 2012
Current FSM state                     : Degraded
```

```
Time FSM entered current stat         : Fri Dec 7 21:58:29 2012
Last FSM state                        : Converging
Time FSM entered last state           : Fri Dec 7 21:58:19 2012
Cluster operational state             : Degraded

Stable AppNav controller View:
    21.0.0.145
    21.0.0.160

Stable SN View:
    21.0.0.149
```

The reason for the difference in AppNav Controller views on the AppNav Controllers may be due to a mismatch in the AppNav Controller group configuration on the AppNav Controllers or due to a connectivity problem between the AppNav Controllers.

It is also useful to check the alarms on each of the AppNav Controllers by using the following command that also suggests corrective actions:

```
router# show service-insertion alarms detail support

Critical Alarms:
----------------
    Alarm Instance     Alm ID  Module     AC/SN IP Addr  AO      SNG
    1 degraded_cluster 29002   cmm        N/A            N/A     N/A

    Cluster protocol detected inconsistency in AC view of peer ACs. Device will
pass-through all new connections.

    Cluster view is degraded.

        Explanation:

                Cluster membership manager has detected a discrepancy in the AC view of peer
ACs. Optimization will be turned off on this device for cluster consistency.

        Action:

                Check the network for partial connectivity issues.

Major Alarms:
-------------
    Alarm Instance     Alm ID  Module     AC/SN IP Addr  AO      SNG
    1 ac_unreachable   29006   cmm        192.168.1.11   N/A     N/A

    Cluster protocol on device cannot communicate with peer AC ("192.168.1.11").

        AppNav controller is unreachable.


        Explanation:

                Cluster protocol detected failure of the peer AC. This could happen due to
several reasons - configuration mismatch or network issues preventing communication between
the ACs or the AC actually being down.

    Action:
        Other alarms will indicate if this is a configuration issue. If so, correcting the
configuration mismatch will cause this alarm to go away. Otherwise, check the network to
see if the devices are able to communicate with each other.

Minor Alarms:
-------------
None
```

Service Node Excluded

If no traffic is redirected to a particular service node and you are using an AppNav Controller group with two or more AppNav Controllers, it is possible that the service node is excluded. This happens when the service node view is not the same on each of the AppNav Controllers.

To check the stable service node view on each of the AppNav Controllers, use the **show service-insertion service-context** command.

The reason for the difference in service node views could be due to a mismatch in the service node group configuration on the AppNav Controllers or due to a connectivity problem between one or more of the AppNav Controllers and the excluded service node.

To check if any service nodes are excluded or unreachable, look for the SN_excluded and SN_unreachable alarms by using the **show service-insertion alarms detail support** command on each of the AppNav Controllers.

Flows Not Synced Between AppNav Controllers

This could be due to a mismatch in the VRF names for the traffic seen by the AppNav Controllers in the ACG.

Check the output of the **show service-insertion statistics connection summary** command for the counter for Flow Sync Failures due to vrf mismatch.

```
router# show service-insertion statistics connection summary
Number of 2T entries=0
Number of 3T entries=0
Number of optimized flows=0
Number of pass-through flows=0
Flow sync failures due to vrf-mismatch=3
```

Connection Hangs

A connection might be considered “hung” for various reasons. In many cases, it helps to use telnet to simulate a connection to the server. For example, enter **telnet** *HTTP_server* **80**.

If the connection hangs during the TCP 3-way handshake, verify that both the connection and the route to the service node are properly set up.

If the connection hangs after the connection was established, verify the connection along the path. Make sure that the MTU along the path is correct.

Use the **show service-insertion statistics connection** command on the AppNav Controller and the **show statistics connection** command on the service node to cross check the connections between the AppNav Controller and the service node.

Use the **show platform hardware qfp active statistics drop** command to check for packet drops.

Connection Resets

You can usually see the reason for the connection reset by issuing the **show statistics connection closed** command and the **show statistics connection closed conn-id** *connection_ID* command on the service node. Capturing packets is also useful in analyzing the reason for the connection reset.

If you use ISR-WAAS as the service node, verify the DRE peer ID by using the **show dre** command on the service node and making sure that both of the ISR-WAAS DRE peer IDs are not the same. **vm init** is required for ISR-WAAS during configuration.

Use the **show platform hardware qfp active statistics drop** command to check for dropped packet.

Application Accelerator Status Shows as Red with No Load

Some older service nodes may not support all application accelerators.

Individual application accelerators, such as the video application accelerator, require a separate license.

The AppNav-XE Component Fails to Initialize

If the system displays an ERROR_NOTIFY syslog message when you enable the **service-insertion waas** command on the interface, it could be that the AppNav-XE component failed to initialize due to low memory. Check the amount of memory by using the following command:

```
router# show platform hardware qfp active infrastructure exmem statistics
QFP exmem statistics

Type: Name: DRAM, QFP: 0
  Total: 268435456
  InUse: 102283264
  Free: 166152192
  Lowest free water mark: 166152192
Type: Name: IRAM, QFP: 0
  Total: 134217728
  InUse: 8186880
  Free: 126030848
  Lowest free water mark: 126030848
Type: Name: SRAM, QFP: 0
  Total: 32768
  InUse: 15088
  Free: 17680
  Lowest free water mark: 17680
```

If the available memory is less than 10 percent of the total memory, the AppNav-XE component may not be able to initialize, which results in no flows being redirected.

If the output of the **show policy-map target service-context waas/1** command is blank, instead of listing the AppNav policy being used, it may indicate that the system was unable to initialize.

Flow Limit Reached

Both the AppNav Controller and the service nodes have a limit on the number of flows that they can support. On the AppNav Controller, the limit is 2 million flows. Beyond that, all flows are passed through. If you exceed the limit, the system displays the following error message:

```
03/10 00:53:51.720 [errmsg]: (warn): %CFT_CLIENT-4-MAX_FCS_TOUCH_WARN: CFT number of
flow-context threshold is reached, can't allocate more memory for flow-context.
```

The flow limit may be reached in advance due to available memory. In this case, the system displays the following syslog message:

```
*Aug 24 00:29:17.205: %CFT_CLIENT-4-CFT_MEMORY_BOUNDARY_TOUCH_WARN: F0: cpp_cp:  CFT
reached maximum configured memory utilization. Can't allocate more memory for
flow-context.
```

In both cases, when the existing flows are completed and the number of flows dips below the threshold, flows are optimized again.

Application Installation Errors

The following errors can happen during the application installation:

- Code signing failure.
- Not able to extract the libvirt data.
- Resource not available on box.
- Profile selection failed due to the profile database not being populated correctly or not being synced to Cisco IOS-XE.
- The OVA package is not compatible with Cisco IOS-XE or the AppNav-XE component.
- ISR-WAAS is not able to get the boot strap configuration.
- The Cisco IOS-XE image does not support KVM, ISR-WAAS, or the AppNav-XE component.

These errors might occur due to an incompatibility issue between the ISR-WAAS OVA and the Cisco IOS-XE image. Ensure that you have the Cisco IOS-XE Release 3.9 image and the latest compatible OVA package. When selecting an OVA profile to use during activation, make sure that the Cisco ISR 4451-X has enough memory and storage resources for the profile selected.

Degraded Performance

If your performance is degraded, first check the CPU utilization on the application using the following command:

```
router# show virtual-service utilization name ISR4451-X-WAAS_application_name
```

If the application CPU is at 100 percent, check the bandwidth and compression ratio in the ISR4451-XWAAS application. If the bandwidth is within the profile limits, enable ISR-WAAS debugging. If the application CPU is not at 100 percent, check the router CPU using the following command:

```
router# show processes cpu
```

If the platform CPU is at 100 percent, follow the standard platform debug process. If the application is bound by bandwidth and there is CPU headroom on the Cisco ISR 4451-X, consider upgrading the application profile by reinstalling the ISR-WAAS application and selecting a higher profile.

Another reason for degraded performance is application disk failure. Check the ISR-WAAS installation log and ISR-WAAS alarms on WCM to see if any disk failure has occurred.

End Users Cannot Reach the Server

If ISR-WAAS crashes, traffic should not be impacted because the AppNav-XE component stops redirecting traffic.

Note Crashes in ISR-WAAS application accelerators do not impact connectivity because traffic goes into bypass mode.

Other AppNav-XE Known Issues

If the AppNav Controller does not respond to a WAAS TCP trace, the system forwards the TCP trace to the service node and the service node generates a response along with a list of service nodes along the path.

Accessing the ISR-WAAS Application

You can access the ISR-WAAS application by using the following CLI command:

```
router# virtual-service connect name AUTOWAAS console
Connected to appliance. Exit using ^c^c^c
Cisco Wide Area Application Engine Console

Username: admin
Password: xyz
System is initializing. Please wait...
Please use 'show disks details' to monitor system status.
NO-HOSTNAME#
```

For more information, see the *WAAS Configuration Guide*.

Example of ISR-WAAS Running Configuration

A sample ISR-WAAS configuration is displayed below:

Note This sample does not include the policy configuration.

```
router# show run no-policy
! waas-universal-k9 version 5.1.0 (build b15 Aug 17 2012)
!
device mode application-accelerator
!
interception-method appnav-controller
!
!
hostname router-ISR4451-X-WAAS
!
!
primary-interface Virtual 1/0
!
interface Virtual 1/0
ip address 2.79.27.2 255.255.255.0
exit
!
ip default-gateway 2.79.27.1
!
!
```

```
no auto-register enable
!
! ip path-mtu-discovery is disabled in WAAS by default
!
username admin password 1 ****
username admin privilege 15
!
!
authentication login local enable primary
authentication configuration local enable primary
!
tfo tcp optimized-send-buffer 2048
tfo tcp optimized-receive-buffer 2048
!
!
no dre auto-bypass enable
!
no service-insertion pass-through offload enable
!
!
kernel kdb
central-manager address 2.79.5.12
cms enable
!
!
!
stats-collector logging enable
stats-collector logging rate 30
!
service-insertion service-node
enable
exit
!
! End of WAAS configuration
```