Configuring Cisco WAAS with Akamai Connect

This chapter describes how to configure Cisco WAAS with Akamai Connect, which is an integrated solution that combines WAN optimization and intelligent object caching to accelerate HTTP/S applications, video, and content.

Note Throughout this chapter, the term Cisco WAAS device is used to refer collectively to the Cisco Wide Area Application Services (Cisco WAAS) Central Managers and Cisco Wide Area Application Engines (WAEs) in your network. The term WAE refers to WAE and Cisco Wide Area Virtualization Engine (WAVE) appliances, and Cisco Virtual WAAS (vWAAS) instances.

This chapter contains the following sections:

- Benefits of Cisco WAAS with Akamai Connect
- Deployment Options for Cisco WAAS with Akamai Connect
- Operating Considerations for Cisco WAAS with Akamai Connect
- Supported WAAS Platforms for Akamai Connect
- Workflow for Enabling and Using Akamai Connect
- Enabling Akamai Connect and Activating Akamai Connect License
- Enabling Akamai Connected Cache
- Enabling OTT Caching
- Caching Types and Setting Caching Policies
- Using HTTP Proxy for Connections to the Akamai Network
- Cisco Cloud Web Security and Force IMS Features
- Configuring Cache Prepositioning for Akamai Connect
- Configuring HTTP/S Preposition Proxy for Akamai Connect
- Cisco Support for Microsoft Windows Update

Note **Akamai Connect** is the HTTP/S object cache component added to Cisco WAAS, integrated into the existing WAAS software stack and leveraged via the HTTP Application Optimizer. WAAS with Akamai Connect helps to reduce latency for HTTP/S traffic for business and web applications.

Akamai Connected Cache is a component of Akamai Connect, which allows the Cache Engine to cache content that is delivered by an Edge server on the Akamai Intelligent Platform.

Benefits of Cisco WAAS with Akamai Connect

The Akamai Connect feature is an HTTP/HTTPS object cache component that is added to Cisco WAAS. It is integrated into the existing WAAS software stack and is leveraged via the HTTP Application Optimizer.

Akamai Connect helps reduce latency for HTTP/HTTPS traffic for business and web applications, and can improve performance for many applications, including Point of Sale (POS), HD video, digital signage, and in-store order processing. It provides significant and measurable WAN data offload, and is compatible with existing WAAS functions such as DRE (deduplication), LZ (compression), TFO (Transport Flow Optimization), and SSL acceleration (secure/encrypted) for first and second pass acceleration.

The following are some of the benefits of adding Akamai Connect to WAAS:

- Intelligent transparent object caching (by integrating Akamai's cache).
- Seamless integration of Akamai Connect in WAAS software and configuration (with WAAS Central Manager and WAAS CLI).
- Integration with Akamai's Edge Grid Network, which provides low-latency Content Delivery Network transfers (via Akamai Connected Cache).
- Significant and measurable WAN data offload.
- Cache prepositioning (warming) for websites that you specify.
- Hostname rules for cache control of specific websites or domains.
- First and second pass acceleration, because Akamai Connect works with WAAS middle-mile capabilities (including DRE, LZ, TFO, and SSL acceleration).
- Dual-sided or single-sided network deployment.

Deployment Options for Cisco WAAS with Akamai Connect

This section contains the following topics:

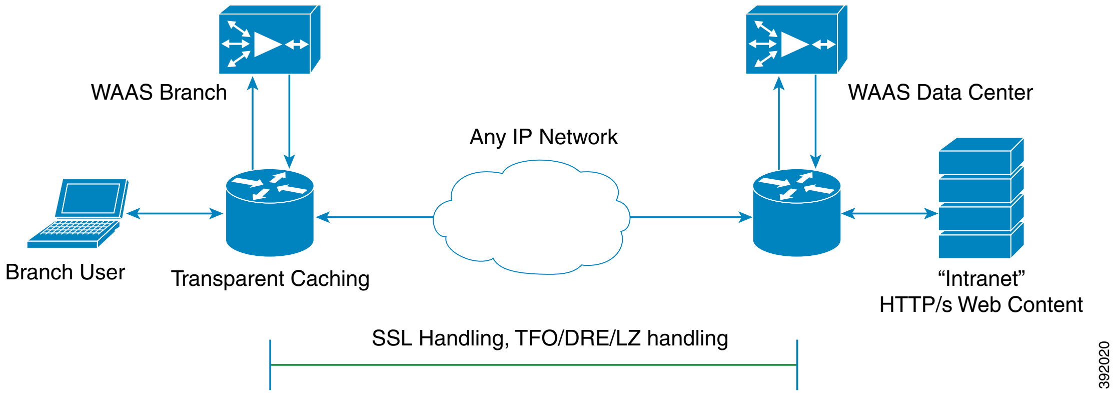
- Dual-Sided Deployment
- Single-Sided Deployment

Dual-Sided Deployment

Dual-sided deployment (Figure 13-1) provides the benefits of WAAS technology plus Akamai caching for HTTP and HTTPS traffic:

- Transparent caching of customer-owned, Intranet web resources
- Caching in branch only.
- Includes prepositioning (for non-SSL content).

Figure 13-1 Dual-sided Deployment



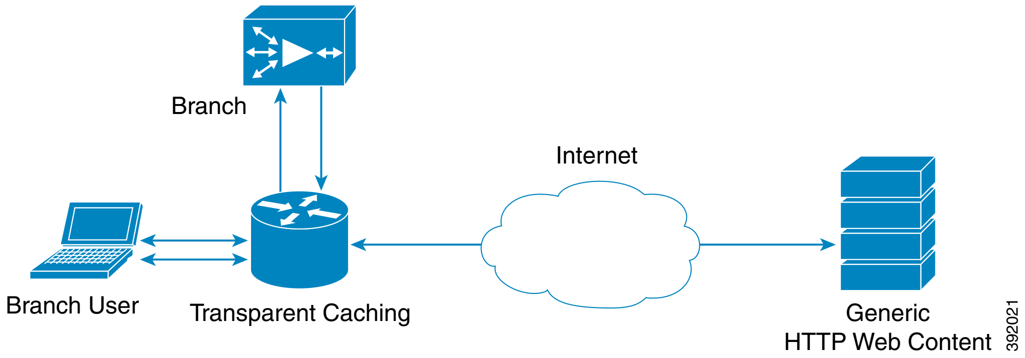
Single-Sided Deployment

Single-sided deployment (Figure 13-2) provides the benefits of WAAS technology plus Akamai caching for HTTP and HTTPS traffic:

the benefits of HTTP object caching.

- Generic web resources that utilize proxy-specific HTTP cache-control headers.
- Caching in branch only.
- Includes prepositioning (for non-SSL content).
- Single-sided deployment is on by default with transparent caching in Standard mode.

Figure 13-2 Single-sided Deployment



Operating Considerations for Cisco WAAS with Akamai Connect

Consider the following when using Cisco WAAS with Akamai Connect:

- You cannot view the contents of the cache, and cannot pin content to make it remain in the cache, for example, for prepositioned content.

- There is no separate cache for HTTPS content. However, data is stored differently for the same site if both HTTP and HTTPS are accessing. (The way the sites are stored in the cache is based on the URL, and this will change between HTTP and HTTPS.)

The Cache Engine has no explicit integration with AppNav. The AppNav status is based on the HTTP application accelerator.

Note The terms *mode*, *profile*, and *policy* are sometimes used interchangeably to describe caching types and processes. This document uses these terms as follows:

Mode—The version of transparent caching (Basic, Standard, Advanced, or Bypass).

Profile—The set of host rules and caching types applied as a group, and which follows the Cache Engine order of precedence.

Policy—The set of rules and the conditions of caching, applied either singly or as a group, to device or device group.

Supported WAAS Platforms for Akamai Connect

The flow of allocated resources to the Akamai Cache Engine is controlled by the WAAS Central Manager, but the overall resource pool and the amount of resources that can be allocated to the Akamai Cache Engine is controlled by the hardware platform, and the number of supported connections and users that the router is designed to service.

This section contains the following topics:

- Supported WAAS Platforms for Akamai Connect up to 6,000 Connections
- Supported WAAS Platforms for Akamai Connect beyond 6,000 Connections

Supported WAAS Platforms for Akamai Connect up to 6,000 Connections

Table 13-1 shows the supported WAAS platforms for Akamai caching up to 6,000 connections, for WAAS Version 5.4.1 and later. For details on supported WAAS platforms beyond 6,000 connections, see Supported WAAS Platforms for Akamai Connect beyond 6,000 Connections.

Table 13-1 Supported WAAS Platforms for Akamai Caching up to 6,000 Connections

Appliance	SM	vWAAS	ISR-WAAS
		vWAAS-150	ISR-G2 and ISR-G3
WAVE-294	SM-700	vWAAS-200	ISR-WAAS-750 (ISR-4451, ISR-4431, ISR-4351, ISR-4331, ISR-4321)
WAVE-594	SM-900	vWAAS-750	ISR-WAAS-1300 (ISR-4451, ISR-4431)

Appliance	SM	vWAAS	ISR-WAAS
WAVE-694	SM-710	vWAAS-1300	ISR-WAAS-2500 (ISR-4451)
	SM-910	vWAAS-2500	
		vWAAS-6000	

Note If you are upgrading from a version earlier than vWAAS Version 5.4, you will need a third disk and possibly more memory added. For more information, see the "Cisco vWAAS with Akamai Connect" chapter in the *Cisco Virtual Wide Area Application Services Configuration Guide*.

Supported WAAS Platforms for Akamai Connect beyond 6,000 Connections

The flow of allocated resources to the Akamai Cache Engine is controlled by the WAAS Central Manager, but the overall resource pool and the amount of resources that can be allocated to the Akamai Cache Engine is controlled by the hardware platform, and the number of supported connections and users that the router is designed to service. For supported WAAS platforms up to 6,000 connections, see Supported WAAS Platforms for Akamai Connect up to 6,000 Connections.

This section contains the following topics:

- WAVE and vWAAS Models for Akamai Connect beyond 6,000 Connections
- Configuring HTTP-OC on WAVE-7541/7571/8541
- Configuring HTTP-OC on vWAAS-12000/50000

WAVE and vWAAS Models for Akamai Connect beyond 6,000 Connections

For WAAS Version 6.2.1 and later, the Akamai Connect Cache Engine is supported for scaling beyond 6,000 connections on the following platforms:

- WAVE-7541, WAVE-7571, and WAVE-8541
- vWAAS-12000 and vWAAS-50000

Table 13-2 shows supported vWAAS models for Akamai caching beyond 6,000 connections, and disk and memory requirements for Akamai caching beyond 6,000 connections

Table 13-2 Supported vWAAS Models and Memory/Disk Requirements for Akamai Connect beyond 6,000 Connections

vWAAS Model	Total HTTP Object Cache Connections (K)	Cache Engine Cache Disk (GB)	Additional Resource to be Added
vWAAS-12000	12	750	6GB RAM, 750 GB disk
vWAAS-50000	50	850	850 GB disk

Note For vWAAS with WAAS Version 6.2.x, vWAAS with Akamai Connect beyond 6,000 connections is not supported for Cisco vWAAS on RHEL KVM or KVM on CentOS.

Caution When a WAE—a WAVE model used for Akamai Connect beyond 6,000 connections—is assigned to a device group in the WAAS Central Manager *after* Akamai Connect is already enabled, you must manually reload the device. Akamai Connect will remain in shutdown state until the reload is performed.

Scaling for these platforms is based on memory availability and scale performance. Table 13-3 shows the total HTTP object cache connections, memory required for the cache engine, cache engine cache disk, and additional resources, if needed.

The Akamai Connect Cache Engine connection-handling capacity is determined by the upper limit of memory that is given to the Akamai Connect Cache Engine at startup. The Akamai Connect Cache Engine will allocate memory as needed up to the upper limit. In case of overload, the connection will be optimized by HTTP-AO, without a caching benefit.

Note For vWAAS-12000 and vWAAS-50000, HTTP object cache will scale up to the platform TFO limit. To achieve this, you must augment the platform resources (CPU, RAM, and disk) during provisioning.

For vWAAS-12000, you must allocate at least 6 GB of additional RAM.

For vWAAS-1200 and vWAAS-50000, you must allocate Cache Engine cache disk resources. Cache disk requirements are shown in Table 13-3.

Table 13-3 WAAS Mid to High End Platform Memory/Disk Requirements for HTTP-OC

Cisco WAAS Platform	Total HTTP Object Cache Connections (K)	Cache Engine Cache Disk (GB)	Additional Resource to be Added
WAVE-7541	18	708	N/A
WAVE-7571	45	839	N/A
WAVE-8541	112	675	N/A
vWAAS-12000	12	1500	6GB RAM, 750 GB disk
vWAAS-50000	50	2350	850 GB disk

Configuring HTTP-OC on WAVE-7541/7571/8541

Configuring HTTP-OC on WAVE-7541, WAVE-7571 or WAVE-8541 includes configuring the Device Profile feature and re-partitioning the WAVE disk.

- **Device Profile** (WAAS Version 6.2.1 and later)—The Device Profile feature enables the device mode as “branch” which tunes the resource allocation for various WAAS services as a branch traffic scenario and branch services. Device Profile is enabled at the *individual device* level; it is not enabled for an entire device group.
- **Data Disk Partitions** (WAAS Version 6.2.1 and later)—To accommodate the larger-scale connections available for WAAS Version 6.2.1 with Akamai Connect, the single partition for the RAID5-based disk subsystem is split into multiple partitions.

To configure HTTP-OC on WAVE-7541, WAVE-7571 or WAVE-8541, follow these steps.

Step 1 Upgrade the WAAS Central Manager and WAE devices to WAAS Version 6.2.1 or later.

For complete upgrade instructions, including critical prerequisites before upgrading the WAAS Central Manager to WAAS Version 6.2.1 or later, see the Release Note for Cisco Wide Area Application Services for your WAAS release.

After upgrade, the Device Profile feature—**Device** > *device-name* > **Configure** > **Caching** > **Device Profile**—is initially disabled, and the **Branch** check box in the Device Profile screen section is unchecked. See Step for how to enable Device Profile after an upgrade.

Step 2 After upgrade is complete enable HTTP-OC, from the WAAS Central Manager or from the WAAS CLI.

Step 3 When you enable HTTP-OC for WAVE-7541/7571/8541, you will receive a message to run the **disk delete-data-partitions** EXEC command and to restart the system.

Note The **disk delete-data-partitions** command deletes all data partitions on all logical drives, including CONTENT, PRINTSPOOL, and SYSFS partitions. These partitions include all DRE and SMB object cache files, SYSFS and print spool files. New partitions are created at system restart.

Step 4 Upgrading the system and configuring HTTP-OC from the WAAS Central Manager (to upgrade from the WAAS CLI, see Step 5):

a. Enable Akamai Connect.

b. A message is displayed with two requirements, for disk delete data partitions and Device Profile enable, and includes an approval request for these.

c. After user approval, the WAAS Central Manager will initiate disk delete data partitions process and enable the Device Profile feature.

d. Restart the system.

e. By default, from the WAAS Central Manager, Akamai Connect will be enabled at restart.

Note For WAVE models 7541 and 8541, the Device Profile feature is automatically set/unset when you enable/disable HTTP OC. For WAVE-7571, the Device Profile setting requires you to reboot to change the Device Profile feature status.

Step 5 Upgrading the system and configuring HTTP-OC from the CLI (to upgrade from the WAAS Central Manager, see Step 4):

a. Enable Akamai Connect.

b. A message is displayed to run the **disk delete-data-partitions** EXEC command.

c. By default, from the CLI, Akamai Connect is not enabled at system restart.

d. Enable Akamai Connect.

e. Here are operating considerations when Akamai Connect is enabled:

- You can disable the Device Profile feature if you disable Akamai Connect on the device. To disable the Device Profile feature, uncheck the **Branch** check box.
- For WAVE models 7541 and 8541, the Device Profile feature is automatically set/unset when you enable/disable HTTP OC. For WAVE-7571, the Device Profile setting requires you to reboot to change the Device Profile feature status.

- You can disable the Device Profile feature if you disable Akamai Connect on the device. To disable the Device Profile feature, uncheck the **Branch** check box.
- To enable the Device Profile feature from the CLI, use the **device mode** global configuration command, to configure the device to function as application accelerator (**application-accelerator**) or WAAS Central Manager (**central-manager**).

Note For the WAVE-7571, you can also use the device mode command to configure the device to function as a branch device (profile-branch), to configure pre-allocation resrouces for various WAAS services to be branch traffic scenario and branch services.

The branch profile enabled connection count used for computing memory for pre-allocation is 3/4 of the TFO limit.

f. To show device profile settings, use the **show device-mode current** EXEC command.

g. To show the configured device mode (the mode that is configured but has not yet taken effect), use the **show device-mode configured** EXEC command.

Configuring HTTP-OC on vWAAS-12000/50000

As noted in the procedure Configuring HTTP-OC on WAVE-7541/7571/8541, the process includes running the **disk delete-data-partitions** EXEC command and restarting the system. The **disk delete-data-partitions** command deletes all data partitions on all logical drives, including CONTENT, PRINTSPOOL, and SYSFS partitions. These partitions include all DRE and SMB object cache files, SYSFS and print spool files. New partitions are created at system restart.

To configure HTTP-OC on vWAAS-12000 or vWAAS-50000 and to avoid object and DRE caching being lost due to execution of the the **disk delete-data-partitions** command, you must downgrade from WAAS Version 6.2.x to WAAS Version 5.x, and then upgrade to WAAS Version 6.2.x. Use one of the following downgrade/upgrade procedures:

- Downgrade/Upgrade with Additional Akamai Cache Disk Removed/Reinstalled
- Downgrade/Upgrade with Additional Akamai Cache Disk Remaining In Place

Downgrade/Upgrade with Additional Akamai Cache Disk Removed/Reinstalled

To configure HTTP-OC on vWAAS-12000 or vWAAS-50000 with the additional Akamai Cache disk removed and then reinstalled, follow these steps:

Step 1 The device is at WAAS Version 6.2.x, with Akamai Connect enabled.

Step 2 Disable Akamai Connect.

Step 3 Power down the device.

Step 4 Remove the additional Akamai Cache disk.

Step 5 Power on the device.

Step 6 Downgrade from WAAS Version 6.2.x WAAS Version 5.x.

Step 7 Upgrade the WAAS Central Manager and WAE devices to WAAS Version 6.2.x.

Step 8 After upgrade is complete, power off the device.

Step 9 Re-install the additional Akamai Cache disk.

Step 10 Power on the device.

Step 11 Enable Akamai Connect.

Step 12 Enable HTTP-OC from the WAAS Central Manager or from the WAAS CLI.

Step 13 A message is displayed regarding the required additional memory and disk resources (shown in Table 13-3).

Step 14 Power down the vWAAS VM and add the necessary resources the vWAAS VM.

Step 15 Power up the vWAAS VM.

Step 16 HTTP-OC is enabled on the vWAAS.

Downgrade/Upgrade with Additional Akamai Cache Disk Remaining In Place

To configure HTTP-OC on vWAAS-12000 or vWAAS-50000 with the additional Akamai Cache disk remaining in place, follow these steps:

Step 1 Upgrade the WAAS Central Manager and WAE devices to WAAS Version 6.2.1 or later.

Step 2 Enable Akamai Connect.

Step 3 Downgrade from WAAS Version 6.2.x to WAAS Version 5.x.

Step 4 Upgrade the WAAS Central Manager and WAE devices to WAAS Version 6.2.x.

Step 5 Run the **disk delete-data-partitions** EXEC command and restart the system.

- From the CLI, a message is displayed to run the **disk delete-data-partitions** EXEC command and restart the system.
- The WAAS Central Manager does not display this message.

Note After the upgrade, You must run the **disk delete-data-partitions** command to enable Akamai Connect.

Note The **disk delete-data-partitions** command deletes all data partitions on all logical drives, including CONTENT, PRINTSPOOL, and SYSFS partitions. These partitions include all DRE and SMB object cache files, SYSFS and print spool files. New partitions are created at system restart.

Step 6 Enable Akamai Connect.

Workflow for Enabling and Using Akamai Connect

Table 13-4 Workflow for Enabling and Using Akamai Connect

Task	Section
1. Confirm that your WAAS configuration has all requisite components to work with Akamai Connect.	• Supported WAAS Platforms for Akamai Connect • Prerequisites for WAAS with Akamai Connect
2. Enable Akamai Connect.	• Enabling Akamai Connect
3. Register and activate Akamai Connect.	• Activating the Akamai Connect License
4. Enable Akamai Connected Cache.	• Enabling Akamai Connected Cache
5. (Optional) Enable OTT caching.	• Enabling OTT Caching
6. (If needed) De-register and re-register a WAAS device.	• Deregistering and Reregistering a WAAS Device
7. (If needed) Replace an expired Akamai Connect license.	• Replacing an Inactive or Expired Akamai Connect License
8. Set one caching policy for all sites.	• Caching Types and Setting Caching Policies
9. Set an individual caching policy for specific sites.	• Caching Types and Setting Caching Policies
10. Set Force IMS policies.	• Cisco Cloud Web Security and Force IMS Features
11. Configure cache prepositioning	• Configuring Cache Prepositioning for Akamai Connect
12. View cache prepositioning status.	• Viewing Cache Prepositioning Task Status
13. (Optional) Copy cache prepositioning tasks.	• Copying Cache Prepositioning Tasks
14. (Optional) Configure HTTP/S preposition proxy.	• Configuring HTTP/S Preposition Proxy for Akamai Connect
15. View Akamai Connected Cache Statistics.	• Akamai Connected Cache Charts in Chapter 15, “Monitoring Your WAAS Network”
16. (If needed) Set up HTTP proxy connections to the Akamai network.	• Using HTTP Proxy for Connections to the Akamai Network

Enabling Akamai Connect and Activating Akamai Connect License

This section contains the following topics:

- Prerequisites for WAAS with Akamai Connect
- Enabling Akamai Connect
- Activating the Akamai Connect License
- Deregistering and Reregistering a WAAS Device

- Replacing an Inactive or Expired Akamai Connect License

Prerequisites for WAAS with Akamai Connect

Before you enable Akamai Connect, confirm that your WAAS configuration has the following:

- The WAAS Central Manager and WAAS appliances are updated to software version 5.5.1 or later.
- A verified NTP service that is within 30 seconds of the NTP standard server (NTP.org). For how to configure the NTP server, see Configuring an NTP Server in Chapter 10, “Configuring Other System Settings.”
- A working public DNS server configured on the WAAS devices and the WAAS Central Manager. For how to configure the DNS server, see Configuring the DNS Server in Chapter 6, “Configuring Network Settings.”
- The ability for the WAAS Central Manager to reach Akamai’s Luna system via HTTPS on port 443. (The custom hostname is in your activation file.)
- The ability for WAAS devices to make a connection to the Akamai Management Gateway (AMG) to get the authentication key. The WAAS device configured for Akamai Connect needs the correct network connectivity to access the AMG every day to get correct credentials and updated metadata. WAAS will make an HTTPS connection on port 443 to the AMG to get this information.

  If the WAAS devices cannot go direct to the Internet, you can configure them to use the WAAS Central Manager as a proxy.

  Note The Akamai Connected Cache feature will stop functioning if WAAS loses communication with the AMG for more than 48 hours.

Enabling Akamai Connect

To enable Akamai Connect, follow these steps.

Step 1 From the WAAS Central Manager menu, from either the Device Groups or Devices tab, choose **Configure** > **Caching** > **Akamai Connect**.

The Akamai Connect window appears, with two tabs: Cache Settings and Cache Prepositioning.

Step 2 Choose the **Cache Settings** tab.

Note If you are configuring the Akamai Connect feature for a device group, the device group should have only devices that support Akamai Connect. For more information, see Supported WAAS Platforms for Akamai Connect.

Step 3 Check the **Enable Akamai Connect** check box to turn on the Akamai Connect Cache Engine. When the End-User License Agreement (EULA) dialog box appears, click **Accept**.

Step 4 Click **Submit**.

Note When you create settings for the first time, either at the device or the group level, the Akamai license upload file is displayed, and you can select the license file supplied and click **Submit**. For more information on activating the Akamai Connect license, see Activating the Akamai Connect License.

Note Turning on the Cache Engine starts active caching in Standard mode. If you want Advanced or Bypass mode, you must specify it. This step is described in Setting Caching Policies.

Step 5 Continue to **Edit Settings** screen section, described in , or **Advanced Cached Settings** screen section, described in .

Note To edit *any* settings, including advanced settings and cache preposition, the Akamai Connect feature must remain enabled.

Activating the Akamai Connect License

Before you begin the registration process to activate Akamai Connect, confirm the readiness of your WAAS configuration, as described in Prerequisites for WAAS with Akamai Connect.

Note For information on the status of an active Akamai Connect license, see Akamai Connect Diagnostics Using the Central Manager in Chapter 16, “Troubleshooting Your WAAS Network.”

To receive and activate the Akami Connect activation file, follow these steps:

Step 1 Purchase a license for Akamai Connect from your Cisco account representative or reseller.

Step 2 The account representative or reseller enters the order into the Cisco Commerce Workspace (CCW) system. The order ***must*** specify an email address for eDelivery of the Activation file.

Step 3 CCW contacts the Akamai Luna Portal to request a license or licenses for the number and type of Akamai licenses entered.

Step 4 Akamai generates and sends the license(s) to the CCW system in the form of a single activation file.

Step 5 The CCW system sends an email, with the activation file attached, to the email address specified in the order. The order of priority for selecting the email address in a CCW order is::

- Priority1: eDelivery email address
- Priority2: end customer email address
- Priority3: shipping contact email address

Note If you do not provide an email address in your order, you will not receive an activation file.

Step 6 Enable Akamai Caching on each WAE. There are two paths available to reach the Akamai Connect screen. You can use either one to enable Akamai Connect to use any of the transparent caching methods, Akamai Connected Cache, or OTT. If this is the first time you are navigating to the Akamai Connect screen, you will be prompted to provide the activation file for licensing.

- From the WAAS Central Manager choose **Device/Device Group > Configure > Caching > Akamai Connect**.
  OR
- From the WAAS Central Manager choose **Home > Admin > Licenses > Akamai Connect**. This path can be used later to add more licenses, if needed.

  The Akamai Connect screen is displayed.

Step 7 At the **Upload Akamai Connect License file** field, click **Browse**, highlight the activation file and click **OK**.

Step 8 Click **Upload**. The authentication data in the activation file is transmitted to the Akamai Luna portal.

Step 9 After the device message is sent to the Luna portal:

- The Luna portal sends the Entitlement Code to the WAAS Central Manager and the Akamai Management Gateway (AMG).
- The WAAS Central Manager sends the Entitlement Code to WAAS.
- The AMG rolls out the Entitlement Code to Edge Servers on the Akamai Grid Network.

The Entitlement Code is maintained on Luna, on the AMG, and on the WAAS device. WAAS connects to the AMG using a proxy/DNS server that can resolve the address **amg.terra.akamai.com**.

Step 10 The list titled **Status of devices with Akamai Connect feature configured** displays the following types of status for one, some, or all devices.

- Akamai Device Status - ActivationInProgress, Active
- Operational Status - Disconnected, Connected, or Running
- Connectivity to Akamai - Activating, Activated, or Connected

The device registration, operational status, and connectivity to Akamai proceed through a set of status indicators for the three status categories: Akamai Device Status/Operational Status/Connectivity to Akamai:

- **ActivationInProgress /Disconnected /Activating**
- **ActivationInProgress /Connected /Activating**
- **Active /Connected /Activated**
- **Active /Connected /Connected**

Note The activation process for WAAS devices may take between 15-60 minutes to complete, and for this time period, the **Connectivity to Akamai** status displays as **Activating**. During this time, device(s) may not be able to communicate with the Akamai Network, because they are not recognized by the AMGs until the activation process is complete, and the **Connectivity to Akamai** status displays as **Connected**.

Step 11 For the last steps in the registration process, Luna sends the Connected Cache credentials to the AMG and to the Edge Servers on the Akamai Grid network. The AMG forwards Connected Cache credentials on to WAAS. With the Connected Cache credentials on both WAAS and the Edge Servers, the Connected

Cache is enabled, and caching requests can be served by the Edge servers. This authenticated connection can then service requests for Connected Cache and OTT caching from the Akamai Grid network Edge Servers.

Step 12 The registration of each WAE begins. The WAAS Central Manager provides information to the Akamai Luna Portal for each device that will be running Akamai Connect.

Note **Connected** Operational Status can take several minutes to complete. Rollout of the activation to the Edge servers can take up to 45 minutes to complete. A device may take from a few minutes to up to two hours to show an **Active** Activation Status, depending on when the request was made, traffic conditions, and other variables.

Step 13 Each WAE that has been sent the entitlement code will try to make an SSL connection to the AMG using **amg.terra.akamai.com**. The Luna Portal will push out the Akamai Connected Cache credentials to the AMG and Akamai Grid Network (to the Akamai Edge Servers).

- The AMG will push the Akamai Connected Cache credentials out to each of the WAEs that are configured for Akamai Connected Cache. If OTT is enabled, the OTT metadata needed to help cache YouTube objects is also processed at this time.
- The Akamai Connected Cache credentials are sent by the WAE Cache Engine when going to the origin server. If the WAE Cache Engine has valid credentials according to the Akamai Edge Server, the Akamai Edge Server then provides objects to the WAE Cache Engine that are not normally cacheable to other devices.

Step 14 The WAE Cache Engine will request new credentials daily and will be good for two days. The connections are always established from WAE or WAAS Central Manager over TCP 443 to the AMG.

- For security, firewalls are usually deployed by performing statefull insepction on traffic from within the company to the outside. They are also configured to block unknown traffic from the outside to the inside. Since connection should not initiate from AMG to any WAAS Central Manager or WAE at any time, there should not be an issue. If there is, then a hole will need to be made to allow the WAAS Central Manager or WAE to speak to any device on port 443.

Note The Devices listing on the **All Devices** screen includes a column titled **Akamai Connect**, which shows the status of each device: Active, Not Supported, Connected, Disconnected.

Step 15 As needed, configure HTTP proxy or external HTTP proxy, as described in Using HTTP Proxy for Connections to the Akamai Network.

Deregistering and Reregistering a WAAS Device

When you deregister a WAAS device from the WAAS Central Manager, the WAAS Central Manager will trigger the removal of the device record on the Akamai side, thereby invalidating the entitlement key used by the Cache Engine to talk to AMG devices. On the WAAS side, the Cache Engine will continue operating in transparent cache mode.

When you reregister a WAAS device with the WAAS Central Manager, one of two things will happen:

- The WAAS Central Manager will auto-assign the device to device groups (that are so marked). If any of these device groups have Akamai Connect/HTTP cache settings, the WAAS Central Manager will trigger registration with Akamai.

- If no device group is configured with Akamai Connect/HTTP cache settings, the registration is done individually.

After the device is registered, it will get a new entitlement key.

Replacing an Inactive or Expired Akamai Connect License

If your license has become inactive or expired, follow these steps to replace your license:

Step 1 When a license is inactive or expired, a notification is displayed in one of two WAAS Central Manager screens:

- At the **Home** > **Admin** > **Licenses** > **Akamai Connect** screen: “Akamai Connect License is Inactive. Please remoeve current license and import valid license.”
- At the **Home** > **Monitor** > **Troubleshoot** > **Akamai Diagnostics** screen: “Akamai Connect License is Inactive. Please remove existing license and import new one using Akamai License page.”

Step 2 Remove the inactive or expired license.

Step 3 To upload a new license file, at the **Home** > **Admin** > **Licenses** > **Akamai Connect** screen, click Choose File to browse to the new license file and click **Upload**.

Step 4 If you import an expired license, you will see the message: “Unable to communicate to Akamai server (Error: License is inactive or expired). See Central Manager log file for detailed error information.”

Step 5 To obtain a new license, contact your Cisco account representative or reseller.

Enabling Akamai Connected Cache

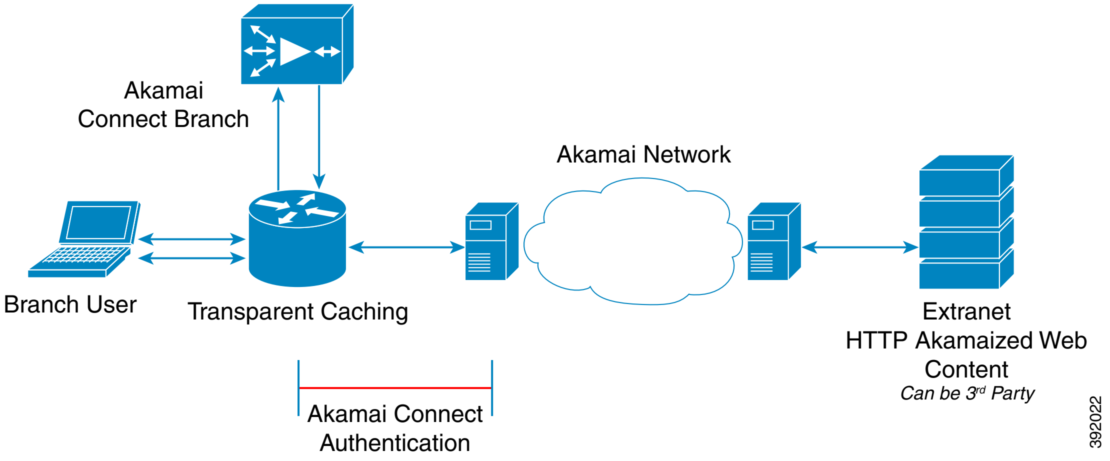
This section contains the following topics:

- About Akamai Connected Cache
- Akamai Connected Cache Requirements
- Procedure for Enabling Akamai Connected Cache

About Akamai Connected Cache

Akamai Connected Cache (Figure 13-3) allows the Cache Engine to cache content that is delivered by an Edge server on the Akamai Intelligent Platform. This is content that is served by the worldwide Akamai Content Delivery Network (CDN); it is typically not cacheable by enterprise cache engines, but can be cached in Akamai Cache Engine based on interactions with network edge elements that are serving it.

Figure 13-3 Akamai Connected Cache



Akamai Connected Cache provides the following features:

- Object caching is done on the client-side WAAS device only.
- Prepositioning can be leveraged to cache HTTP websites delivered via the Akamai Intelligent Platform.
- During the enabling/registration of HTTP object cache, each WAE Cache Engine contacts the Akamai network to obtain credentials.
- The WAAS/Akamai Cache Engine determines which sites can be "Akamaized" by Akamai Connected Cache from the HTTP headers in the first reply. The Cache Engine and the Akamai Edge Server then exchange credentials and agree that Akamai Connected Cache can occur. This is done again via HTTP headers in HTTP request and responses.
- The Akamai Edge Servers can provide objects it is handling, the object that will not change, to allow WAEs with Akamai Cache Engine and the correct credentials to cache these objects. Users or other caches without valid credentials will not be allowed to cache.
- The Akamai Edge Server provides additional headers to allow the WAAS/Akamai Cache Engine to cache the objects for the objects it handles. The Cache Engine forwards this back to the corresponding client. The headers passed between the Cache Engine and the client are similar to what the client or enterprise proxy server would see if the WAE was not in the path.

Akamai Connected Cache Requirements

Akamai Connected Cache is enabled by default when you check the **Enable Akamai Connect** check box at the Akamai Connect **Cache Settings** tab (**Configure** > **Caching** > **Akamai Connect**).

Akamai Connected Cache requires registration and an authentication key to operate. For how to disable/enable Akamai Connected Cache see Procedure for Enabling OTT Caching.

Procedure for Enabling Akamai Connected Cache

You can configure the Akamai Connected Cache - Cache Engine settings at the **device group level** (to apply a configuration to all registered devices) or the **device level** (to apply a configuration to a particular registered device).

To enable Akamai Connected Cache, follow these steps. For more information on Akamai Connected Cache, see About Akamai Connected Cache.

Step 1 To enable Akamai caching, check the **Akamai Connected Cache** check box. The default is enabled. When you enable Akamai connected cache, it is enabled for all suitable Akamaized content.

Step 2 Click **Submit.**

Step 3 After you enable Akamai Connected Cache, you can set a caching policy for all sites, or an individual caching policy for specific sites, as described in Setting Caching Policies.

Step 4 After you enable Akamai Connected Cache, you can configure cache prepositioning, as described in Configuring Cache Prepositioning for Akamai Connect.

Enabling OTT Caching

This section contains the following topics:

- Overview of OTT Caching
- Procedure for Enabling OTT Caching

Overview of OTT Caching

This section contains the following topics:

- About OTT Caching
- Sites that Support OTT Caching
- Workflow for OTT Caching with WAAS and Akamai Connect

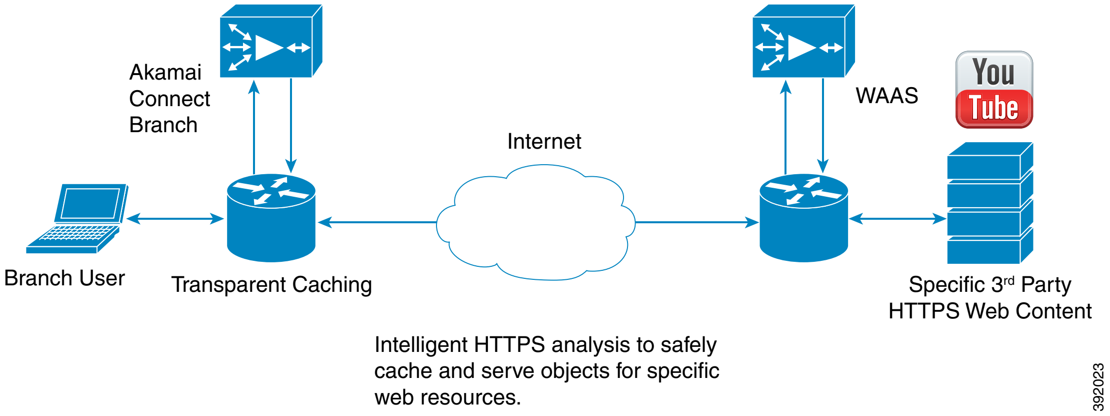
About OTT Caching

Over the Top (OTT) caching caches dynamic content by examining the URL related to a session and a site to determine if the object is identical to the one previously stored in the Cache Engine cache. OTT is used for streamed content, particularly video content, and for sites that use dynamic URLs based on session or authentication methods. Figure 13-4 shows an example of OTT caching.

OTT is a caching feature that Akamai has engineered to allow WAAS to cache and serve some popular sites that are normally not cacheable. This caching feature requires special metadata that is created and distributed by Akamai. OTT uses metadata logic to determine a unique cache key per video; this allows dynamic URLs to be cached.

Note OTT is disabled by default, but enabled after HTTP application accelerator object cache is enabled. For how to enable or disable OTT, see Procedure for Enabling OTT Caching.

Figure 13-4 Example of OTT Caching



Sites that Support OTT Caching

Sites that support OTT caching include the following:

- Apple
- Google
- Lynda
- Microsoft Updates
- Office 365
- Pearson
- Salesforce
- Schoology
- Vimeo
- Youku
- YouTube
  - Since YouTube is delivered via HTTPS, you need to follow the same process as you do for SAAS optimization. For more information, see Configuring SSL Acceleration for SaaS Applications in Chapter 12, “Configuring Application Acceleration.” The domains that must be matched are *.youtube.com, *.ytimg.com, *.googlevideo.com, and *.ggpht.com.

Workflow for OTT Caching with WAAS and Akamai Connect

The following is a workflow for OTT caching WAAS and Akamai Connect for YouTube:

1. During the registration process the WAE Cache Engine provides metadata for YouTube.

2. A client goes to the YouTube site. (Note that the one client request actually requests the video in chunks, even with a dynamic ID. Each chunk not only contains a part of a video, but has an associated audio/video quality, such as 360p, 480p, or 720p.
3. The Akamai Cache Engine uses metadata logic to determine a unique cache key per dynamic ID. The Cache Engine stores this for one day, even though YouTube usually expires the dynamic ID in approximately six hours.
4. Each time the video is played, the request reaches out to the origin server and fetches the dynamic ID. It then compares this with the dynamic ID and cache key pair in the cache.

   If the dynamic ID associated with the video has changed, the video will be served from the origin, and this will result in a miss. A new cache key is generated for that ID and is then stored.

   If a match is found, the video is served out of cache.

Procedure for Enabling OTT Caching

To enable OTT caching, follow these steps.

Step 1 To enable Over the Top (OTT) caching, check the **Over the Top Cache** check box. In the initial release, OTT caching applies only to YouTube.

Step 2 Click **Submit** or continue to Advanced Cache Settings. For more information on Advanced Cache Settings, see Advanced Mode.

Caching Types and Setting Caching Policies

This section contains the following topics:

- Transparent Caching
- Order of Preference for Caching Types
- Setting Caching Policies
- Server Address Validation

Note For Cisco WAAS with Akamai Connect, the terms Mode, Profile, and Policy are defined as follows:

Mode—The version of transparent caching (Basic, Standard, Advanced, or Bypass)

Profile—The set of host rules and caching types applied as a group, and which follows the Cache Engine order of precedence.

Policy—The set of rules and the conditions of caching, applied either individually or as a group, to a device or device group.

Transparent Caching

This section contains the following topics:

- Overview of Transparent Caching
- Four Modes of Transparent Caching

Overview of Transparent Caching

Transparent caching (which conforms to the RFC-2616 standard) delivers content from an origin server to the client without any modification. Transparent caching sends a request from a client to a server along with the associated authentication. No changes are made by proxy servers to either the headers or the returned packets along the way, although there are some headers that mark proxy actions that can be altered without the meaning of the cache control headers being altered.

There are four types of transparent caching modes: Basic, Standard, Advanced, and Bypass.

There are two modes in which transparent caching can operate: single-sided mode and dual-sided mode.

Note When accessing transparent caching via HTTPS, the default caching mode is Basic mode. This ensures that no sensitive content is accidentally cached (in Basic mode, only content that you explicitly mark is cached). If you want content cached in a different mode with HTTPS, create a host rule that matches the HTTPS server location. For more information on creating a host rule, see Setting Caching Policies.

Four Modes of Transparent Caching

This section describes the four modes for Transparent caching, as shown in Table 13-5.

Table 13-5 Transparent Caching: Overview of Caching Modes

Caching Mode	Description
Basic Mode	Caches only the objects marked explicitly as cacheable.
Standard Mode (Default)	Caches objects marked as cacheable, and objects with no explicit cache marker and with a last-modified date. Ignores “reload” headers from clients.
Advanced Mode	Caches media types more aggressively, and caches all object types for longer times, when there is no explicit expiration time.
Bypass Mode	Turns off caching for a configured site or sites.

Basic Mode

In Basic mode, the Cache Engine works in strict RFC-2616 behavior, and therefore, only caches responses that are marked explicitly as cacheable with Cache-Control Headers or that have an Expire header - to service and accelerate traffic from a datacenter to a branch office over any type of IP network. Caching is only in the branch or local router, and content can be cached from the Internet regardless of the location of the original source.

Standard Mode (Default)

In Standard mode (default), the Cache Engine also follows RFC-2616 behavior for cache control headers, but with the following differences from Basic mode:

- In Standard mode, the Cache Engine does not honor client cache override behavior, for example, must-revalidate and proxy-revalidate.
- If cache-control or expire headers are not present, and Last Modified Time appears, the Cache Engine performs a heuristic based on the Last Modified Time and stores objects for 10 percent of their apparent age, up to a maximum of one day.

Caution A properly configured website will work with Standard mode, but login pages, cookie setting pages, or dynamic content not properly marked as cacheable may break. We recommend that you test the website; this is especially important for a newly-created website or one that does not have many users.

Advanced Mode

In Advanced mode, the Cache Engine caches media types more aggressively, and caches all object types for longer times (when there is no explicit expiration time). Most of the benefits of Advanced mode over Standard mode occur if the website has not already marked cacheable media content properly. Advanced mode is best suited for media-rich Intranet sites.

If cache-control or expire headers are not present and Last Modified Time appears, the Cache Engine performs a heuristic based on the Last Modified Time and stores objects for 20 percent of their apparent age, up to a maximum of one day.

For certain media file types, listed in Table 13-6, Advanced Mode will cache these for a full day if the media type is not specified as uncacheable or the media type has no obvious age in the request. For all other media types, the system caches the object for a minimum of one hour to a maximum of seven days - regardless of whether the Last Modified Time is present.

Table 13-6 Advanced Mode: Media types that may be cached for a full day

Advanced Mode: Media types that may be cached for a full day (if not specified as uncacheable or has no obvious age in the request)										
3g2	3gp	aac	aif	aiff	asf	asx	au	avi	bin	bmp
cab	carb	cct	cdf	class	css	dcr	doc	docx	dtd	dv
dvd	dvr	dvr-ms	exe	flv	gcf	gff	gif	grv	hdml	hqx
ico	ini	jpeg	jpg	js	m1v	m4a	midi	mov	mp3	mp4
mpeg	mpg	mpv	nv	pct	pdf	png	ppc	ppt	pptx	pws
qt	swa	swf	tif	txt	vbs	w32	wav	wbmp	wma	wml
wmlc	wmls	wmlsc	wmv	xsd	xsl	xls	xlsx	zip		

Caution A properly configured website will work in Advanced mode, but Advanced mode may break the presentation of certain web pages if there are even minor caching misconfigurations. We recommend that you test the performance of this caching mode for your applications before you bring the Cache Engine into production. When testing, pay particular attention to dynamic URLs and to content that requires authentication to be presented to a client.

Bypass Mode

In Bypass mode, the Cache Engine turns off caching for one or more configured sites. When Transparent Bypass mode is set for a particular hostname, the caching for the hostname specified in a rule is suppressed.

Bypass mode is useful when you want to turn off Akamai Connected Cache or OTT caching for a site or for a part of a site.

For example, if you have servers of the type images#.bar.com, you can configure a bypass rule so that only images2.bar.com is excluded from caching. All other images#.bar.com servers will continue to be cached under the existing rules.

Order of Preference for Caching Types

When there are multiple caching modes and policies in use, the Cache Engine applies an order of precedence in the execution of these. A rule that is higher in the order of precedence is executed first, and any other rules that are applied to that domain or digital property is ignored. The order of precedence is:

1. Transparent caching rules
2. OTT/Akamai Connected Cache
3. Default Transparent policy

For example, if test.com is an Akamai Connected Cache property, but an Adanced mode cache rule is set for this site, then Advanced mode will take precedence and Akamai Connected Cache will be skipped.

Note When cache prepositioning is turned on, it has the same priority as any other caching type.

Note Akamai Connect determines cache type based on most exact hostname match followed by cache priorities. www.host.com is more exact than *.host.com. In this scenario, if a lower-priority cache, such as Akamai Connected Cache (Order of Precedence #2), has a more exact match than a higher priority cache, such as transparent (Order of Precedence #1), the caching will occur with the more exact match and lower-priority cache.

Setting Caching Policies

For how to set one caching policy for all sites, or how to set individual caching policies for a specific site, follow these steps:

Step 1 From Devices or Device Groups, navigate to **Configure** > **Caching** > **Akamai Connect**.

Step 2 Choose the **Cache Settings** tab.

Step 3 In the **Advanced Cache Settings** section, at the **Default Transparent Caching Policy** drop-down list, choose a caching policy:

- Basic
- Standard (default)
- Advanced
- Bypass

Step 4 *To set a default caching policy for all sites,* choose a caching policy and click **Submit**. To enable transparent caching for a specific site, see Step 5.

Step 5 *To enable transparent caching for a specific site,* change the **Default Transparent Caching Policy** to **Bypass**.

Step 6 At the **Site Specific Transparent Caching Policy** section, click **Add Hostname/IP**. The **Site Caching Policy Task** dialog box opens.

a. In the **Hostname/IP** field, specify the hostname of the site to be configured. The hostname can be a specific server, or a domain name that contains a wildcard, such as *.cisco.com.

Note When you configure **Bypass mode** as the site-specific transparent caching policy, you *must* specify a complete server name or complete (FQDN) domain name. If you use a wildcard to specify sites for Bypass mode, the sites will still be optimized via Akamai Cache.

b. At the **Transparent Caching Policy** drop-down list, select the cache policy for this site: Basic, Standard, Advanced, or Bypass.

c. Click **OK**. The new hostname/IP is added as a line item to the Site Specific Transparent Caching Policy table.

Note The policy you set for a specific site takes precedence over the default caching policy set for all sites.

You can configure up to 512 hostnames for each site-specific transparent caching policy.

Step 7 Configure Cisco Cloud Web Security (CWS) user policy. For more information see Cisco Cloud Web Security and Force IMS Features.

Step 8 Configure HTTP Proxy:

- To configure WAAS Central Manager as HTTP proxy, see Using the WAAS Central Manager as HTTP Proxy
- To configure external HTTP proxy, see Configuring External HTTP Proxy.

Server Address Validation

This section contains the following topics:

- About Server Address Validation
- Configuring Server Address Validation
- Alarms Used with Server Address Validation
- Upgrade/Downgrade Considerations for Server Address Validation

About Server Address Validation

Server Address Validation prevents malicious content from infecting the Akamai Connect cache, in situations like the one described here:

Example situation:

- The server IP address to which a DNS name resolves might be an IP address of a server that contains malicious content, rather than that of an expected and trusted server.
- The resulting response would get cached, and the Akamai cache would then contain malicious content.
- After the cache is "poisoned" with this malicious content, other clients accessing the same content would also get served with this malicious data.

To prevent such situations, the server address validation feature:

- Performs DNS lookups on the name in the HTTP host header.
- Compares the lookup result with that connection's forward IP address.
- If there is a mismatch, the transaction is allowed to pass through the cache, however—*no* content is allowed to be cached.

Note For the server address validation feature to work properly, your system must have a valid DNS configuration. For how to configure DNS for your WAAS system, see Configuring the DNS Server in the chapter "Configuring Network Settings."

Note The Cisco Cloud Web Security (CWS) feature also performs traffic scanning and malware protection for the Akamai Connect cache. For more information on CWS, see Cisco Cloud Web Security and Force IMS Features.

Configuring Server Address Validation

This section describes how to use the WAAS Central Manager to enable or disable Server Address Validation, and to add, edit, or delete bypass server addresses into or from a whitelist.

Note To configure Server Address Validation with the CLI, use the **accelerator http object-cache validate-address enable** and **accelerator http object-cache validate-address bypass** global configuration commands. For more information on these commands, see the *Cisco Wide Area Application Services Command Reference*.

To configure Server Address Validation with the WAAS Central Manager, follow these steps:

Step 1 Before you begin, note these guidelines:

- A valid DNS configuration is required for Server Address Validation to work properly.

  For how to configure DNS for your WAAS system, see Configuring the DNS Server in the chapter "Configuring Network Settings."

- If Interposer-SSL is disabled, the following warning message is displayed:

  Interposer-SSL is in disabled state. Enable Interposer-SSL for HTTP Object Cache Server Validation feature to use SNI extension. Peformance for HTTPS connections, when this feature is enabled, might get affected in the absence of SNI.

  For more information on Interposer-SSL, see Enabling and Disabling the Global Optimization Features in the chapter "Configuring Application Acceleration."

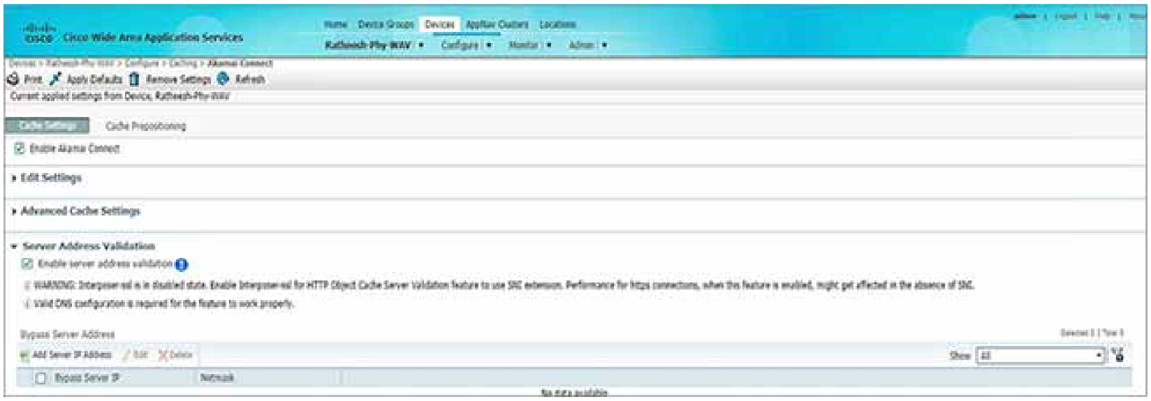
Step 2 From Devices or Device Groups, navigate to > **Configure** > **Caching** > **Akamai Connect**.

Step 3 Choose the **Cache Settings** tab.

Step 4 Enabling Server Address Validation:

In the Server Address Validation section (Figure 13-5) of the Cache Settings screen, check the **Enable server address validation** check box. The default setting for Server Address Validation is disabled.

Figure 13-5 Server Address Validation Panel



Current applied settings from Device, Ratheesh-Phy-WAV

☑ Enable Akamai Connect

☑ Enable server address validation

WARNING: Interposer-ssl is in disabled state. Enable Interposer-ssl for HTTP Object Cache Server Validation feature to use SNI extension. Performance for https connections, when this feature is enabled, might get affected in the absence of SNI.

Valid DNS configuration is required for the feature to work properly.

□ Bypass Server IP	Netmask

No data available

Note the following operating guidelines about enabling or disabling Server Address Validation:

- If server address validation is enabled or disabled at the device group level, the option is enabled or disabled to all devices in the device group.
- If you have enabled or disabled Server Address Validation from the CLI, it will be reflected in the WAAS Central Manager within two data feed cycles.

Step 5 Creating a server address whitelist:

In the Bypass Server Address table listing taskbar, click **Add Server IP Address**. The **Bypass Server** dialog box appears.

a. In the **Bypass Server IP** field, specify the server IP address.

b. In the **Netmask** field, specify the netmask.

c. Click **OK**. The new server IP address and netmask are added to the Bypass Server Address table listing.

d. You can configure up to 50 server IP addresses per whitelist.

Step 6 Editing or deleting a bypass server address:

At the Bypass Server Address table listing taskbard, use the **Edit** or **Delete** buttons to edit or delete existing server IP addresses for the whitelist.

Note A server address whitelist that you have created is stored on the data server until you delete it. The server IP address whitelist is not automatically deleted if you disable Server Address Validation.

Alarms Used with Server Address Validation

Table 13-7 shows the alarms used with Server Address Validation.

Table 13-7 Alarms Used with Akamai Connect Cache Server IP Address Validation

Alarm Name	Reason Alarm is Raised	User Action
DNS Lookup Failed	• Too frequently, address checks have been unable to look up hostnames.	• Verify that the DNS configured in WAAS is able to do host lookup.
Forward Proxy Detected Warning	• Address checks are enabled, but a forward proxy that is re-looking up hostnames has been detected.	• Do one of the following: – Disable address checks. – Disable the forward proxy from re-looking up hostnames. – Disable address checks for the forward proxies' IP addresses. – Add forward proxies' IP addresses to the whitelist.
Address Check Failures Warning	• Server address validation has found that too many address mismatches—the IP address returned by DNS lookup does not contain the address used by the client—are occurring.	• Use the **show hosts** command to list and verify the name servers and their corresponding IP addresses, and to list the hostnames, their corresponding IP addresses, and their corresponding aliases (if applicable).

Upgrade/Downgrade Considerations for Server Address Validation

- Downgrading to a WAAS version earlier than WAAS Version 6.4.1—Server Address Validation is not available for WAAS versions earlier than 6.4.1.
- Upgrading from WAAS Version 6.4.1 to a later version—Server Address Validation is available for all WAAS versions 6.4.1 and later.

Using HTTP Proxy for Connections to the Akamai Network

This section contains the following topics:

- Overview of HTTP Proxy Connections to the Akamai Network
- Using the WAAS Central Manager as HTTP Proxy
- Configuring External HTTP Proxy

Overview of HTTP Proxy Connections to the Akamai Network

When using Akamai Connect, the WAAS Central Manager and WAAS device(s) must be able to communicate with the Akamai Network: with the Akamai Luna API servers to provision entries for WAAS devices, and with the Akamai AMG devices for Akamai Connected Cache and OTT features.

However, some WAAS deployments may disallow outgoing connections to the Internet for the WAAS Central Manager or WAAS device(s). For these deployments, the WAAS device(s) may use an HTTP proxy to contact the Akamai Network.

You can set up the following proxy configurations:

- No HTTP proxy use
- Using the WAAS Central Manager as HTTP Proxy
- Configuring External HTTP Proxy

For these three proxy configurations, WAAS supports five deployment scenarios:

Deployment Scenario	Deployment Connections	WAAS Central Manager to Luna API Servers	WAAS HTTP Cache Engine to Akamai AMG
No HTTP proxy use	Direct/ Direct	Direct	Direct
WAAS Central Manager as HTTP proxy	Direct/ WAAS Central Manager as proxy	Direct	WAAS Central Manager as HTTP proxy
External HTTP proxy	Direct/ External HTTP proxy	Direct	External HTTP proxy
External HTTP proxy	External HTTP proxy/ Direct	External HTTP proxy	Direct
External HTTP proxy	External HTTP proxy/ External HTTP proxy	External HTTP proxy	External HTTP proxy

The following considerations apply to all HTTP proxy deployments:

- You configure HTTP proxy from the WAAS Central Manager; there are no CLI commands for HTTP proxy. Configuring HTTP proxy settings does not require restart of the WAAS Central Manager.
- HTTP Proxy must support HTTP Connect method for tunneling HTTPS connections.
- Configuring the HTTP proxy setting does not require restart of the WAAS Central Manager.

Note WAAS v5.5.1 does not support HTTP proxy user authentication. It is recommended that you restrict access to proxy using IP address ACLs.

Using the WAAS Central Manager as HTTP Proxy

Note the following considerations when using the WAAS Central Manager as a proxy to the Akamai network:

- When using Akamai Connected Cache, each WAAS Cache Engine device is communicating with the Akamai network. Some WAAS deployments may disallow WAE devices to establish outgoing connections to the Internet (i.e., private networks). In this case, the WAE device may use the WAAS Central Manager device(s) as proxy for all connections to the Akamai network.
- You may still have to allow a hole for the WAAS Central Manager to make communications on TCP port 443 outbound.
- There is no option for the WAAS Central Manager to use a proxy device to get to the Internet.
- All connections are made from the WAAS Cache Engine device or WAAS Central Manager out to the Akamai network; never from the Akamai network to the WAAS Cache Engine device or WAAS Central Manager.
- You configure this feature from the WAAS Central Manager only, not the CLI.

To use the WAAS Central Manager as HTTP proxy, follow these steps:

Step 1 From Devices or Device Groups, navigate to **Configure** > **Caching** > **Akamai Connect**.

Step 2 Choose the **Cache Settings** tab.

Step 3 Check the **Use HTTP proxy for connections to Akamai network** check box.

Step 4 At the **HTTP Proxy:** drop-down list, select **Central Manager as HTTP Proxy**.

Step 5 Click **Submit**.

Configuring External HTTP Proxy

When using the Akamai Connected Cache, WAAS devices are communicating with the Akamai Network. Some deployments may disallow outgoing connections to the Internet for WAAS devices. For these deployments, WAAS devices can use an HTTP proxy to contact the Akamai Network. For more information on HTTP proxy, see Using HTTP Proxy for Connections to the Akamai Network.

Note HTTP proxy must support HTTP CONNECT for tunneling HTTPS connections.

This section contains two procedures:

- Configuring External HTTP Proxy for a Device or Device Group
- Configuring External HTTP Proxy for All Devices

Configuring External HTTP Proxy for a Device or Device Group

To configure external HTTP proxy for a device or device group, follow these steps:

Step 1 From Devices or Device Groups, navigate to **Configure** > **Caching** > **Akamai Connect**.

Step 2 Check the **Use HTTP proxy for connections to Akamai network** check box.

Step 3 At the **HTTP Proxy:** dropdown list, select **External HTTP Proxy**.

Step 4 Specify a Proxy Host and a Proxy Port:

- **Proxy Host** field - Enter a hostname or address.
- **Proxy Port** field- Enter a value between 1-65535.

Note If the WAAS Central Manager is already using an external HTTP proxy, there is no option displayed to use the WAAS Central Manager as proxy; these fields will display the currently configured HTTP proxy.

Step 5 Click **Submit**.

Configuring External HTTP Proxy for All Devices

To configure external HTTP proxy for all devices, follow these steps:

Step 1 From Home, navigate to **Configure > Global > External HTTP Proxy**.

The screen displays the following message:

```
Some deployments may disallow direct connections from Central Manager to Internet hosts.
This would affect WAAS features such as Akamai Connect, where Central Manager needs to communicate with
Akamai servers. For such deployments WAAS Central Manager may use an external HTTP proxy to contact Internet.
HTTP proxy must support HTTP CONNECT method for tunneling HTTPS connections.
```

Step 2 Specify a Proxy Host and a Proxy Port:

- **Proxy Host** field - Enter a hostname or address.
- **Proxy Port** field- Enter a value between 1-65535.

Note If the WAAS Central Manager is already using an external HTTP proxy, there is no option displayed to use the WAAS Central Manager as proxy; these fields will display the currently configured HTTP proxy.

Step 3 Click **Submit**.

Cisco Cloud Web Security and Force IMS Features

The Cisco Cloud Web Security (CWS) feature provides content scanning of HTTP and secure HTTP/S traffic and malware protection service to web traffic. CWS servers scan web traffic content and either allow or block the traffic based on configured policies. Servers use credentials such as private IP addresses, user names, and user groups to identify and authenticate users and redirect the traffic for content scanning.

Traffic is transparently proxyed by an ASA or ISR to cloud-based CWS servers (called towers), where the web traffic is scanned and if deemed acceptable is provided to the origin server. All traffic coming back is through the CWS tower.

Note For WAAS Version 6.2.1 and later, the CWS feature enforces content filtering by enabling force IMS (If Modified Since) for every cached object, for both single-sided and dual-sided deployment.

For WAAS Versions earlier than 6.2.1, content filtering is enforced on single-sided deployments.

Note the following considerations when using the Cisco CWS and option:

- CWS can be used only when one WAAS device is present in the path.
- When you enable CWS, the Akamai Cache Engine always adds an "if modified since" header to the request so that the response needs to go remote to the origin server (in this case, the Scansafe tower) - so all requests get scanned and no security is bypassed. If a 304 Not Modified is returned, then the Akamai Cache Engine provides the object from the cache. If a 200 Okay is returned, then the object is fetched from the origin server.
- If preposition is enabled and is possible that the flow may be redirected to a CWS tower, follow these recommendations:
  - (Preferred choice) configure a white-list on the ISR or CWS tower to bypass the WAE IP address.
  - On the CWS tower, configure a user or group that the WAE will fall into for authentication and allow it access to all sites on which the preposition is occurring.

To enforce the Cisco CWS user policy, follow these steps:

Step 1 Navigate to **Configure** > **Caching** > **Akamai Connect** > **Cache Settings** tab.

Step 2 At the Advanced Settings section, check the **Force IMS DIA** check box to enable CWS user policy enforcement for content access in case of DIA.

Step 3 At the Advanced Settings section, check the **Force IMS Always** check box to apply CWS user policy enforcement for content access in case of all flows.

Step 4 Click **Submit**.

Configuring Cache Prepositioning for Akamai Connect

This section contains the following topics:

- Overview of HTTP Proxy Connections to the Akamai Network
- Configuring a Cache Preposition Task
- Viewing Cache Prepositioning Task Status
- Copying Cache Prepositioning Tasks

Overview of Cache Prepositioning for Akamai Connect

Cache prepositioning, also known as cache warming, allows you to specify a policy to prefetch and cache content at a specified time. For example, prepositioning content with a URL inside the branch office during non-peak hours can help to improve performance during peak hours, by significantly offloading WAN links.

Cache prepositioning runs at the same priority as other caching types, for example, Akamai Connected Cache or OTT.

For WAAS Version 6.2.1 and later with Akamai Connect, cache prepositioning for Akamai Connect also provides the following cache prepositioning features:

- Processing of manifest files for the video streaming protocols HLS (HTTP Live Streaming) and HDS (HTTP Dynamic Streaming).
- Prepositioning of JNLP (Java Network Launch Protocol) files, which contain URL reference for Java Web Start.

Note In order for HTTPS content to be prepositioned, you must define an SSL accelerated service; otherwise, any HTTP requests encountered in the job will fail, although the preposition task will continue and any objects available via HTTP will be retrieved. For more information on how to define an SSL accelerated service, see Configuring SSL Acceleration in Chapter 12, “Configuring Application Acceleration.

When a scheduled fetch operation begins or is complete, it is added to the Cache Preposition Status table.

Configuring a Cache Preposition Task

A cache preposition task contains elements that are configured in the following sections of the Cache Prepositioning tab:

- Cache Prepositioning Task dialog box—Specify the preposition task name, base URLs for prepositioning, include/exclude types, download rate, recursion depth, and task duration.
- Cache Prepositioning Task dialog box Advanced Settings—Specify the recursion delay time and recursion domains.
- Cache Prepositioning Schedule dialog box—Specify the schedule name for the preposition task, frequency of the task (such as daily or monthly), and start time.

To configure a cache preposition task, follow these steps:

Step 1 From Devices or Device Groups, navigate to **Configure** > **Caching** > **Akamai Connect**.

The Akamai Connect window appears with two tabs: Cache Settings and Cache Prepositioning.

Step 2 Choose the **Cache Prepositioning** tab. At this tab, you can add, edit, or delete cache prepositioning tasks, as well as monitor cache preposition task status.

Step 3 (Optional) Check the **Preposition with DRE** check box to enable DRE for preposition connections. The default is disabled, to prevent negative impact to the DRE byte cache for data that will be stored at the object level.

Step 4 Click **Add Cache Preposition Task**.

The Cache Prepositioning Task dialog box opens.

Step 5 Enter values in the Cache Prepositioning Task dialog box fields, shown in Table 13-8.

Table 13-8 Cache Prepositioning Task Dialog Box Fields

Field	Description
Name	The name of the preposition task. Preposition task name is an alphanumeric identifier up to 47 characters. Special characters like ‘,/,\,{,},(,),?,”,<,>,[,],&,*,” are not allowed. Note the following when specifying a task: • You can configure up to 10 URLs per task. • You can configure up to 10 schedules per task. • You can configure up to 50 tasks per device or device group.
URLs	The base URLs for prepositioning. The maximum length for the URL is 900 characters. Characters that are not allowed in the URL are space, double quotes (“). ASCII characters are allowed in the range of ASCII 33 through ASCII 125. • Use a space to separate multiple URLs. • You can configure up to 10 URLs per task.
Include Types	The object types to include in caching, such as .jsp or .asp, each separated by a comma. The list of object name patterns to be included has a total pattern field limit of 47 characters.
Exclude Types	The object types to exclude from caching, such as .jsp or .asp, each separated by a comma. The list of object name patterns to be excluded has a total pattern field limit of 47 characters.
Download Rate	The maximum download rate, in KBps. Select any value between 0 to 10,000,000 KBps. The default is 20 KBps. A selection of 0 indicates unlimited, or no enforced rate limiting.
Recursion Depth	The depth of the link level at which the content is retrieved. Recursion depth is active only if you check the **Recursive Task** check box. Select 1, 2, 3, 5, 8, 13, or 21 from the drop-down list, or enter any custom value between 1 to 1000. The default is 1. **Note** A greater number of specified levels of links means a greater amount of data stored in the cache, sometimes exponentially more. If the amount of requested prefetched data becomes larger than the cache, the newly requested data will flush all previously stored data, and may slow down other operations that attempt to use the cache.
Duration	The maximum amount of time, in minutes, a preposition task can run before it is halted. The default is no set duration. To set a duration time, select from a range of 1 to 2,147,483,647 minutes. Setting the duration of a task is especially useful to: • Ensure that preposition tasks do not overlap with each other. • Ensure that preposition tasks do not overlap with times of higher user traffic.

Field	Description
User Agent	Provides information on browser and operating system type that servers use to identify and respond to. The server populates the cache with content that is dependent in part on the type of user agent used for this cache prepositioning task. Enter browser and operating system information, in an alphanumeric string, up to a maximum of 256 characters. You can also user the **user-agent** *useragent-name* global configuration command to configure the user agent string
Enable Task	Check the **Enable Task** check box to enable the specified preposition task to run. For the task to run, you must specify at least one URL and one schedule (described in Step 5).
Enable Proxy	For WAAS Version 6.2.1 and later, you can configure a HTTP/S proxy support for preposition tasks. Check the **Enable Proxy** check box to configure external proxy for this preposition task. For details on configuring external proxy for a preposition task, see Configuring HTTP/S Preposition Proxy for Akamai Connect.

Step 6 At the **Advanced Settings** section of the **Cache Prepositioning Task** dialog box, you can specify recursion delay time and recursion hostnames:

Field	Description
Recursion Delay Time	The delay time, in seconds, between requests during recursive download. This simulates user wait time. Recursive delay time is necessary because some servers use the lack of time between requests to detect and restrict web spiders. • Enter a value between 0 to 600 seconds. The default is 2 seconds. • A value of zero provides the best performance when there are no web spider restrictions.
Recursion Domains	The list of server domain suffixes for which recursive spidering is permitted. If the list is empty, then spidering is only permitted within the same domain as the specified URL. You can configure up to ten servers: • The server name is up to 255 characters. • Server names are separated by comma or space.

Step 7 In the **Cache Prepositioning Schedule** section, click **Add Schedule**.

The Cache Prepositioning Schedule dialog box opens

Step 8 Specify the following:

Field	Description
Schedule Name	The name of the schedule for this preposition task. Schedule name is an alphanumeric identifier up to 256 characters. The Schedule Name allows you to provide your own representation of a schedule. For example, for a schedule that occurs each Monday, Wednesday, and Friday at 10:30 a.m. can be named as **`Weekly MWF 10:30AM`** or **`Every Week - MON-WED-FRI at 10:30AM`**.
Frequency	The specified time for prepositioning: yearly, daily, weekly, or monthly days. If you choose monthly days, a calendar with check boxes opens for you to check one, some, or all the days in a month for this schedule.
Start Time (HH:MM)	From the two drop-down lists, choose the hour and minute at which the task schedule should start.

Step 9 In the Cache Prepositioning Schedule dialog box, click **OK**.

Step 10 In the Cache Prepositioning Task dialog box, click **OK**.

Step 11 Click **Submit**.

The new cache prepositioning task is added as a line item to the Cache Prepositioning listing table.

Viewing Cache Prepositioning Task Status

Two tables are provided in the Cache Prepositioning section to show the status of a cache prepositioning task. To view the status of a cache preposition task you have configured, select the task from the first table, the Cache Preposition Listing table. The second table, the Cache Prepositioning Status table, displays information on the selected task.

- *For an individual device*, the cache prepositioning status table shows the selected task status for the current device.
- *For a device group*, the cache prepositioning status table shows the status of the selected cache preposition task, for all devices under that device group.

The following types of information are displayed for the selected task:

Field	Description
Device Name	The name of the selected device.
Start Time	The date, hour, and minute for the task schedule to start.
End Time	The date, hour, and minute for the task schedule to end.
Byte Count	The total number of bytes in cache during the most recent preposition task run.
Object Count	The total count of objects in cache during the most recent preposition task run.
Refresh Bytes	The number of bytes refreshed in cache during the most recent preposition task run.
Refresh Count	The count of objects refreshed in cache during the most recent preposition task run.
Store Bytes	The number of unmodified bytes for objects found in cache during the most recent task run.
Store Count	The count of unmodified objects found in cache during the most recent task run.

Field	Description
Uncacheable Bytes	The number of bytes of uncacheable objects encountered during the most recent task run.
Uncacheable Count	The count of uncacheable objects encountered during the most recent task run.
Status	The status of the task, such as Scheduled, Complete, or Error.
Error	If the task status is "Error," an error message describing the task status is displayed.

Copying Cache Prepositioning Tasks

You can copy cache prepositioning tasks that have a device or device group enabled with Akamai Connect, with WAAS running v5.5.1 or 5.4.1. Use the following methods to copy cache prepositioning tasks:

- Device to device
- Device to device group
- Device group to device
- Device group to device group

Note **Cache Preposition Tasks and WAAS versions:** You can also use the **Copy Tasks** feature to copy a cache preposition task between WAAS Version 5.5.1 devices and device groups and WAAS versions earlier than Version 5.5.1 devices and device groups.

To copy a cache preposition task, follow these steps:

Step 1 Navigate to **Configure** > **Caching** > **Akamai Connect** > **Cache Prepositioning** tab > **Cache Prepositioning** section.

Step 2 Click the **Copy Tasks** button.

The **Cache Prepositioning Task** dialog box opens.

Step 3 At the **From** drop-down list, select a device or device group as the source.

Step 4 At the next drop-down list, select a device or device group as the destination.

Note If you try to copy a task with the same name between device and device groups, the following error message is displayed: **`One or more preposition tasks with the same name already exists in the destination device/DG.`**

Step 5 At the **Existing Cache Prepositioning Tasks** table, select one, some or all of the preposition tasks to be copied.

Step 6 Click **OK**.

The selected cache prepositioning tasks are copied from the source to the destination.

Configuring HTTP/S Preposition Proxy for Akamai Connect

This section contains the following topics:

- Overview of HTTP/S Preposition Proxy for Akamai Connect
- Configuring Global Proxy Host and Port for Preposition Tasks
- Modifying Proxy Settings for an Individual Preposition Task
- Removing Proxy Settings for an Individual Preposition Task

Overview of HTTP/S Preposition Proxy for Akamai Connect

For WAAS Version 6.2.1 and later, you can preposition external content in the case of a deployment with proxy. Consider the following when configuring HTTP/S preposition proxy for Akamai Connect:

- IPv4 proxy is supported for HTTP/S prepositioning.
- The HTTP preposition proxy feature is a feature independent of the WAAS Central Manager and external HTTP proxy features described in the sections Using the WAAS Central Manager as HTTP Proxy and Configuring External HTTP Proxy.
- Specific IP address-based proxy configuration is supported for HTTP/S preposition proxy. File-based and auto-detected configurations are not supported for HTTP/S preposition proxy.

Configuring Global Proxy Host and Port for Preposition Tasks

To configure global proxy host and port for preposition tasks, follow these steps.

Step 1 From the WAAS Central Manager menu, from either the Device Groups or Devices tab, choose **Configure** > **Caching** > **Akamai Connect**.

The Akamai Connect window appears, with two tabs: Cache Settings and Cache Prepositioning.

Step 2 Choose the **Cache Prepositioning** tab.

Step 3 In the **Proxy Host** field, enter the hostname or IP address for the proxy host.

Step 4 In the **Proxy Port** field, enter the port number. Valid port numbers are 0 to 65535.

Step 5 Click **Submit**.

Step 6 Create a preposition task, as described in Configuring a Cache Preposition Task.

Step 7 In the Cache Prepositioning Task dialog box, check the **Enable Proxy** check box.

Step 8 Schedule the task, as described in Steps 7 through 9 of Configuring a Cache Preposition Task.

Step 9 Click **Submit**.

Modifying Proxy Settings for an Individual Preposition Task

To modify proxy settings for an individual preposition task, follow these steps.

Step 1 From the WAAS Central Manager menu, from either the Device Groups or Devices tab, choose **Configure** > **Caching** > **Akamai Connect**.

The Akamai Connect window appears, with two tabs: Cache Settings and Cache Prepositioning.

Step 2 Choose the **Cache Prepositioning** tab.

Step 3 Select a cache prepositioning task that you have configured as proxy.

Step 4 Modify the particular setting or settings.

Step 5 Check the **Enable Task** check box.

Step 6 Check the **Enable Proxy** check box.

Step 7 In the **Cache Prepositioning Schedule** dialog box, select parameters to reschedule the task.

Step 8 Click **OK**.

Step 9 In the **Cache Prepositioning Task** dialog box, click **OK**.

Step 10 Click **Submit**.

Removing Proxy Settings for an Individual Preposition Task

To remove proxy settings for an individual preposition task, follow these steps.

Step 1 From the WAAS Central Manager menu, from either the Device Groups or Devices tab, choose **Configure** > **Caching** > **Akamai Connect**.

The Akamai Connect window appears, with two tabs: Cache Settings and Cache Prepositioning.

Step 2 Choose the **Cache Prepositioning** tab.

Step 3 Select a cache prepositioning task that you have configured as proxy.

Step 4 Check the **Enable Task** check box.

Step 5 Uncheck the **Enable Proxy** check box.

Step 6 In the **Cache Prepositioning Schedule** dialog box, select parameters to reschedule the task.

Step 7 Click **OK**.

Step 8 In the **Cache Prepositioning Task** dialog box, click **OK**.

Step 9 Click **Submit**.

Cisco Support for Microsoft Windows Update

Cisco support for Microsoft Windows Update enables caching of objects used in Windows OS and application updates. Cisco support for Microsoft Windows Update is enabled by default, and enabled only for specific sites.

This section contains the following topics:

- Benefits of Cisco Support for Microsoft Windows Update

- Viewing Statistics for Cisco Support for Microsoft Windows Update
- Cisco Support for Microsoft Windows Update and Akamai Cache Engine

Benefits of Cisco Support for Microsoft Windows Update

The Microsoft OS and application updates are managed by update clients such as Microsoft Update. Microsoft Update downloads the updates via HTTP, often in combination with BITS (Background Intelligent Transfer Service) to help facilitate the downloads. Clients use HTTP range request to fetch updates.

The objects that comprise the updates, such as .cab files, are typically cacheable, so that HTTP object cache is a significant benefit for this process.

For example, for Windows 7 and 8 OS updates—via direct Internet or WSUS (Windows Server Update Services), versions 2012 and 2012R2— more than 98% of the update files, such as .cab, .exe, and .psf files, are served from cache on subsequent updates. Cisco support for Microsoft Windows Update reduces the volume of WAN offload bytes and reduces response time for subsequent Windows updates.

Viewing Statistics for Cisco Support for Microsoft Windows Update

There are two ways to view data generated by Cisco support for Microsoft Windows Update:

- Akamai Connected Cache Charts in Chapter 15, “Monitoring Your WAAS Network,” provides information including WAN response time and WAN offload bytes.
- For WAAS Version 6.1.1 and later, the cache engine access log file has two new fields for Microsoft Windows Update statistics:
  - rm-w (range miss, wait)—The main transaction, a cache miss, which waited for the sub-transaction to fetch the needed bytes.
  - rm-f (range miss, full)—The sub-transaction, a cache write of the entire document.

Example 1:

Example 1 contains two log lines, the main transaction and sub-transaction, when a range is requested on an object that is not in cache:

```
ws8-rt-kb2863725-x64_dd8522e527483cd69bf61d98ee849a2406b97172.psf - -
08/28/2015 12:22:29.663 (fl=27520) 300 13.164 0.000 446 - - 34912 172.25.30.4

191.234.4.50 2905 h - - rm-w 206 GET
http://fg.v4.download.windowsupdate.com/d/msdownload/update/software/secu/2013/07/wind
ows8-rt-kb2863725- x64_dd8522e527483cd69bf61d98ee849a2406b97172.psf - -

08/28/2015 12:24:31.448 (fl=27520) 300 134.949 0.000 355 344 3591542 568 172.25.30.4
191.234.4.50 2f25 m-s - - rm-f 200 GET
http://fg.v4.download.windowsupdate.com/d/msdownload/update/software/secu/2013/07/wind
ows8-rt-kb2863725-x64_dd8522e527483cd69bf61d98ee849a2406b97172.psf - -
```

Example 2:

Example 2 shows a cache hit when a range is requested on an object that is either completely in cache, or in the process of being downloaded. If it is in the process of being downloaded, then the main transaction has latched onto a sub-transaction like the one shown in Example 1.

```
08/28/2015 03:34:36.906 (fl=26032) 300 0.000 50.373 346 - - 13169 172.25.30.4
8.254.217.62 2905 h - - - 206 GET
http://fg.v4.download.windowsupdate.com/d/msdownload/update/software/secu/2013/07/wind
ows8-\ rt-kb2863725-x64_dd8522e527483cd69bf61d98ee849a2406b97172.psf - -
```

Cisco Support for Microsoft Windows Update and Akamai Cache Engine

Cisco support for Microsoft Windows Update enables Akamai Cache Engine to support Windows Update caching in two ways:

- Download and cache full objects even when ranges within objects that not in cache are requested.
- Future range requests on the objects can be served out of cache.

There is a limit, set by OTT metadata during the Akamai Connect registration process, from the start of the object—the number of bytes or the percent of file length—where the download functionality is triggered. A request of a size above the set limit does not initiate a full object download, and the request is forwarded to the origin as is.

Caution Cisco Support for Microsoft Windows update is enabled by default, and enabled only for specific sites. The enabled sites are updated via OTT metadata.

If you want to disable Cisco Support for Microsoft Windows Update, you must disable OTT caching. To do this, uncheck the **Over the Top Cache** check box. However, note that unchecking the **Over the Top Cache** check box disables *all* OTT functionality, both global and custom OTT configurations.

For more information on the Akamai Connect registration process, see Activating the Akamai Connect License.