



## CDSM Legacy APIs

---

### Replication Status APIs

This chapter describes the Replication Status API and the servlet actions it performs. The Replication Status API returns a list of delivery services, Service Engines, or contents, and for each delivery service, an indication whether replication of content for the specified delivery service is complete or not.

### Replication Status API Actions

The Replication Status API is the `ReplicationStatusApiServlet`.

#### Syntax

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet...`

This servlet performs one or more of the following actions:

- [getDeliveryServices](#)
- [getSEsOfDeliveryService](#)
- [getDelivheryServicesOfSE](#)
- [getReplicatedContent](#)
- [getNonReplicatedContent](#)
- [getContent](#)
- [getStatusOfContentItems](#)
- [getStatusOfContentItemInDeliveryService](#)

### getDeliveryServices

Obtains the status of content replication of specified delivery services.

#### Parameter

Either a list of delivery service IDs or the keyword **all** is required.

#### Return

A list of the delivery services, and for each delivery service, a flag indicating whether replication for the specified delivery service is complete or incomplete.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?
action=getDeliveryServices&deliveryService=[all | <deliveryService_ID>,<deliveryService_ID>,...]
```

**getSEsOfDeliveryService**

Obtains the status of content replication for all Service Engines assigned to the specified delivery service.

**Parameter**

- Delivery service ID (required)
- Refetch (optional)—The default is false.

If refetch is set to true, a background request to obtain a newly updated status is sent to all Service Engines assigned to this delivery service. To view the newly available information, the user must call the API again after several minutes without a refetch.

**Return**

A list of all Service Engines assigned to a specified delivery service and, for each specified Service Engine, a flag whether replication for the specified Service Engine is complete or incomplete.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?
action=getSEsOfDeliveryService&deliveryService=<deliveryService_ID>[&refetch=<true | false>]
```

**getDeliveryServicesOfSE**

Obtains the status of content replication for all delivery services assigned to the specified Service Engine.

**Parameter**

- Service Engine ID (required)
- Refetch (optional)—The default is false.

If refetch is set to true, a background request to obtain a newly updated status is sent to all Service Engines assigned to this delivery service. To view the newly available information, the user must call the API again after several minutes without a refetch.

**Return**

A list of all delivery services assigned to a specified Service Engine and, for each delivery service, a flag whether replication for the specified delivery service is complete or incomplete.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?
action=getDeliveryServicesOfSE&se=<SE_ID>[&refetch=<true | false>]
```

**getReplicatedContent**

Lists all replicated items of a specified Service Engine on a specified delivery service, with or without search criteria.

**Parameter**

- Service Engine ID (required)
- Delivery service ID (required)
- Search criteria (optional)

One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards \* or ?.

- Refetch (optional)—The default is false.

If refetch is set to true, a background request to retrieve the content is issued. The updated information is cached on the CDSM and can be retrieved in the next call.

**Return**

A list of all replicated content items on a specified Service Engine for a specified delivery service that matches the search criteria, if the search criteria have been specified.

**Syntax**

```
https://<cdsmlpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?
action=getReplicatedContent&se=<SE_ID>&deliveryService=<deliveryService_ID>[&criteria=
<criteria>][&refetch=<true | false>]
```

## getNonReplicatedContent

Lists all nonreplicated items of a specified Service Engine on a specified delivery service, with or without search criteria.

**Parameter**

- Service Engine ID (required)
- Delivery service ID (required)
- Search criteria (optional)

One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards \* or ?.

- Refetch (optional)—The default is false.

If refetch is set to true, a background request to retrieve the content is issued. The updated information is cached in the CDSM and can be retrieved in the next call.

**Return**

A list of all content items that are not replicated on a specified Service Engine of a specified delivery service that matches the search criteria, if search criteria have been specified. The list includes content items that are yet to be replicated, are in the process of being replicated, or have failed to be replicated.

**Syntax**

```
https://<cdsmlpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?
action=getNonReplicatedContent&se=<SE_ID>&deliveryService=<deliveryService_ID>[&criteria=
<criteria>][&refetch=<true | false>]
```

## getContent

Lists all content items of a Service Engine on a specified delivery service, with or without search criteria.

### Parameter

- Service Engine ID (required)
- Delivery service ID (required)
- Search criteria (optional)

One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards \* or ?.

- Refetch (optional)—The default is false.

If refetch is set to true, a background request to retrieve the content is issued. The updated information is cached on the CDSM and can be retrieved in the next call.

### Return

A list of all content items on the Service Engine of a specified delivery service that matches the specified criteria, if search criteria have been specified.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?action=getContent&se=<SE_ID>&deliveryService=<deliveryService_ID>[&criteria=<criteria>][&refetch=<true | false>]
```

## getStatusOfContentItems

Lists content items of a delivery service, with or without search criteria, in all the Service Engines assigned to that delivery service.

### Parameter

- Delivery service ID (required)
- Search criteria (optional)

One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards \* or ?.

- Refetch (optional)—The default is false.

If refetch is set to true, a background request to retrieve the content is issued. The updated information is cached in the CDSM and can be retrieved in the next call.



### Note

---

When refetch is set to true, the request is sent to the Service Engines assigned to the delivery service to obtain new information. This is a processor-intensive operation.

---

### Return

A list of all content items in the delivery service and their status across Service Engines, or a list of content items that matches the specified criteria and their status across Service Engines, if search criteria have been specified.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?action=
getStatusOfContentItems&deliveryService=<deliveryService_ID>[&criteria=<criteria>][&refetch=
<true | false>]
```

**getStatusOfContentItemInDeliveryService**

Lists the status of a specified content item in the delivery service on all the Service Engines assigned to the delivery service.

**Parameter**

- Delivery service ID (required)
- Complete URL of the content item (required)

**Return**

The status of the specified content item on all the Service Engines assigned to the delivery service.



**Note** This action must be called after the [getStatusOfContentItems](#) action; otherwise, unexpected output results. The URL must be one of the URLs listed in the output of the [getStatusOfContentItems](#) action.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.RepStatusApiServlet?action=
getStatusOfContentItemInDeliveryService&deliveryService=<deliveryService_ID>&criteria=
<complete URL of the delivery service content item>
```

**XML-Formatted Output for Replication Status**

The following is the Document Type Definition (DTD) of the XML-formatted output for the replication status:

```
<?xml version="1.0"?>
<!DOCTYPE replicationStatus[
  <!ELEMENT replicationStatus (message, CeStatus*, Delivery-Service-Status*)>
  <!ATTLIST replicationStatus
    action          CDATA #REQUIRED
    count           CDATA #REQUIRED
  >
  <!ELEMENT message EMPTY>
  <!ATTLIST message
    status          (success | failure) "success"
    message         CDATA #REQUIRED
  >
  <!ELEMENT CeStatus EMPTY>
  <!ATTLIST CeStatus
    ceId            CDATA #REQUIRED
    ceName         CDATA #IMPLIED
    channelId       CDATA #REQUIRED
    channelName    CDATA #IMPLIED
    state          CDATA #IMPLIED
    filesDone      CDATA #IMPLIED
    filesToDo      CDATA #IMPLIED
    filesFailed    CDATA #IMPLIED
```

```

        filesUpdateFailed CDATA #IMPLIED
        totalFiles       CDATA #IMPLIED
        updateTime        CDATA #IMPLIED
    >
<!ELEMENT Delivery servicestatus EMPTY>
<!ATTLIST Delivery servicestatus
    id          CDATA #REQUIRED
    totalNumCes CDATA #REQUIRED
    numCesComplete CDATA #REQUIRED
    numCesInProgress CDATA #REQUIRED
    numCesFailed CDATA #REQUIRED
    numCesUnknownState CDATA #REQUIRED
    rootCeState CDATA #REQUIRED
    manifestError CDATA #IMPLIED
    usedDiskQuota CDATA #IMPLIED
    validAsOf CDATA #IMPLIED
>

```

## Provisioning APIs

This chapter describes the following provisioning APIs and the servlet actions they perform:

- [Delivery Service Provisioning API Actions, page 3-6](#)
- [Location Provisioning API Actions, page 3-31](#)
- [Service Engine Provisioning API Actions, page 3-33](#)
- [Program API Actions, page 3-35](#)
- [Media API Actions for Programs, page 3-39](#)
- [URL Management API Actions, page 3-41](#)
- [Cache Storage Priority Class API Actions, page 3-45](#)
- [Multicast Cloud API Actions, page 3-47](#)
- [External System API Actions, page 3-52](#)

## Delivery Service Provisioning API Actions

The Delivery Service Provisioning API is the ChannelApiServlets.

Some of the output fields are not used for the following actions:

- createDeliveryService
- modifyDeliveryService
- createContentOrigin
- modifyContentOrigin
- configFailoverSettings
- createFailoverOS
- modifyFailoverOS
- deleteFailoverOS
- switchToOS

Table 3-1 lists the unused output fields.

**Table 3-1** Output Fields Not Used in the VDS-IS

Schema Object	Unused Field	Comment
CeConfig	TftpDirectoryListingId	“CeConfig” is mapped to the “Service Engine” schema object.
	WccpConfig	
	TftpProxyList: <list name="TftpProxyList" type="TftpProxy" size="0"/>	TFTP and WCCP are not used.
	WccpRouterListsPerCeForDg : <list name="WccpRouterListsPerCeForDg" type="WccpRouterListPerCeForDg" size="0" />	Although “TftpDirectoryListingId,” “TftpProxyList,” and “WccpRouterListsPerCeForDg” can be queried by API, they are not used in the VDS-IS.
Website	ContentProvidId	“Website” is mapped to the “content origin” schema object.
	CifsWebsites: <list name="CifsWebsites" type="CifsWebsites" size="0" />	Content Provider and CIFS configurations are not used.  Although “ContentProvidId” and “CifsWebsites” can be queried by API, they are not used in the VDS-IS.
Channel	ChannelMCasts: <list name="ChannelMCasts" type="ChannelMCast" size="0" />	“Channel” is mapped to the “delivery service” schema object.  Content Provider and multicast configurations are not used.  Although “ChannelMCasts” can be queried by API, it is not used in the VDS-IS.

### Syntax

<https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet...>

This servlet performs one or more of the following actions:

- [createDeliveryService](#)
- [createDeliveryServiceLoc](#)
- [createDeliveryServiceGenSettings](#)
- [addManifest](#)
- [assignSEs](#)
- [assignDeliveryServiceIp](#)
- [fetchNow](#)
- [modifyDeliveryService](#)
- [modifyDeliveryServiceLoc](#)
- [modifyDeliveryServiceGenSettings](#)
- [modifyManifest](#)
- [unassignSEs](#)

- [unassignDeliveryServiceIp](#)
- [deleteDeliveryServices](#)
- [deleteDeliveryServiceGenSettings](#)
- [addContentItem](#)
- [modifyContentItem](#)
- [deleteContentItem](#)
- [processContentChanges](#)
- [manageHostProxySettings](#)
- [createContentOrigin](#)
- [modifyContentOrigin](#)
- [deleteContentOrigin](#)
- [applyRuleFile](#)
- [applyGeoIpFile](#)
- [configFailoverSettings](#)
- [createFailoverOS](#)
- [modifyFailoverOS](#)
- [deleteFailoveOS](#)
- [switchToOS](#)
- [createChannelDeviceMcastConfig](#)
- [getChannelDeviceMcastConfig](#)
- [modifyChannelDeviceMcastConfig](#)
- [deleteChannelDeviceMcastConfig](#)

## createDeliveryService

Creates a delivery service.

### Parameter

- Delivery service name (required)
- Content origin ID associated with the specified delivery service (required)
- Weak certification (optional)—The default is false.
- Skip encryption (optional)—The default is false.
- Delivery service priority (optional)—The options are high, medium, or low. The default is medium.
- Multicast (optional)—The options are unicast\_only, multicast\_only, or multicast\_unicast. The default is unicast only.
- Live (optional)—The default is false.
- Delivery service quota (optional)—Only valid for non-live delivery services and only applies to prefetched content
- System Qos (optional)
- Delivery service description (optional)—The default is null.



- FailoverIntvl (optional)—The default is 120.
- Never (optional)—The default is false.
- Delivery service QoS (optional)
- sessionQuota (optional)—Session quota
- sessionQuotaAugBuf (optional)—Session quota augmentation buffer
- bandQuota (optional)—Bandwidth quota
- BandQuotaAugBuf (optional)—Bandwidth quota augmentation buffer
- StoragePriorityClass (optional)—Storage priority class ID

**Note**

The mcastEnable parameter is supported in Release 3.1.1.

**Return**

The newly created delivery service ID.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createDeliveryService&deliveryService=<deliveryService_name>&contentOrigin=<contentOrigin_ID>[&weakCert=<true|false>][&skipEncrypt=<true|false>][&priority=<high|medium|low>][&mcastEnable=<unicast_only|multicast_only|unicast_multicast>][&live=<true|false>][&quota=<quota>][&qos=<system|0-63>][&desc=<description>][&failoverIntvl=<failoverIntvl, <20|30|40|50|60|70|80|90|100|110|120>][&never=<true|false>][&deliveryQos=<0-63>][&sessionQuota=<quota>][&sessionQuotaAugBuf=<0-1000>][&bandQuota=<quota>][&bandQuotaAugBuf=<0-1000>][&storagePriorityClass=<storagePriorityClass_ID>]
```

## createDeliveryServiceLoc

Creates a delivery service location object.

**Parameter**

- hssStreamingFromNas(option)—The default is false

**Return**

The newly created delivery service location object value.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createDeliveryServiceLoc&deliveryService=<deliveryService_ID>&location=<location_ID>[&hssStreamingFromNas=<false | true>]
```

## createDeliveryServiceGenSettings

Creates new general settings for a delivery service. Each delivery service has one set of general settings, so this action can only be called once for a delivery service, unless the existing general settings are deleted.

**Parameter**

- `deliveryService` (required)—Delivery service ID
- `Bitrate` (required)—Maximum bit rate limit per session for HTTP (0–2000000)
- `OsProtocol` (required)—Origin server streaming protocol support (0 means HTTP only support, 1 means HTTPS only support)
- `StreamProtocol` (required)—Delivery streaming protocol support (0 means HTTP only support, 1 means HTTPS only support)
- `HashLevel` (required)—URL Hash Level for Cache Routing (0–10)
- `TmpfsSize` (required)—Memory Cache Size (1–10)
- `OsHttpPort` (required)—Origin Server HTTP Port for Web Engine (1–65535, except well-known port numbers), default is 80
- `ReadTimeout` (required)—HTTP Read Timeout (1–60)
- `HttpAllow` (optional)—Disable HTTP Download (True = disable, False = enable)
- `ContentFlowTrace` (optional)—Enable Content Flow Trace (True = enable)
- `FilterTraceFlowToClient` (optional)—Enable Filter Trace Flow to Client (True = enable)
- `HttpExtAllow` (optional)—Enable streaming over HTTP (True = enable)
- `HttpExt` (optional)—HTTP Allowed Extensions (invalid if `HttpExtAllow` is false)
- `GreenCookie` (optional)—Outgoing Cookie
- `EnableCacheError` (optional)—Enable Error Response Caching (True = enable)
- `CacheError` (optional)—Cacheable Error Responses (invalid if `EnableCacheError` is false)
- `OSRedirectEnable` (optional)—Follow Origin Server redirects (True - enable)
- `NrOfRedir` (optional)—Number of redirects allowed (invalid if `OSRedirectEnable` is false)
- `EnableAbrLive` (optional)—Disable File Caching on Disk (True = enable)
- `SkipLL` (optional)—Skip Location Leader Selection for Edge SE (True = enabled)
- `WmtUserAgent` (optional)—WMT User Agent
- `QuotaUsageReporting` (optional)—Force quota usage reporting when bandwidth and session quotas are not configured for the delivery service
- `genericSessionTrack`(option)—The default is false
- `hssSessionTrack`(option)—The default is false
- `hlsSessionTrack`(option)—The default is false

**Note**


---

If the delivery service is a live delivery service, only `deliveryService` and `WmtUserAgent` are valid, all other parameters are not applicable for a live delivery service.

---

**Return**

The newly created delivery service general settings.

**Syntax**

```
https://<cdsmIpAddress>:
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createDeliveryServiceGenSettings&deli
veryService=<deliveryService_ID>&Bitrate=<Maximum bitrate limit per session for HTTP(Kbps)
```

```
(0-2000000)>&HashLevel=<URL Hash Level for Cache Routing(0-10)>&TmpfsSize=<Memory Cache
Size(MB)(1-10)>&OsHttpPort=<Origin Server HTTP Port(web-engine only,default 80)>
&ReadTimeout=<HTTP Response Read Timeout>&OsProtocol=<Delivery streaming protocol
support(0 - HTTP only,1 - HTTPS only)>&StreamingProtocol=<Origin Server streaming protocol
support(0 - HTTP only,1 - HTTPS only)>[&HttpAllow=<truelfalse> Disable HTTP Download]
[&ContentFlowTrace=<truelfalse> Enable Content Flow Trace][&FilterTraceFlowToClient=
<truelfalse> Enable Filter Trace Flow to Client][&HttpExtAllow=<truelfalse> Enable streaming over
HTTP][&HttpExt=<HTTP Allowed Extensions>][&GreenCookie=<Outgoing Cookie>]
[&EnableCacheError=<truelfalse> Enable Error Response Caching][&CacheError=<Cacheable Error
Responses>][&OSRedirectEnable=<truelfalse>Follow Origin Server redirects][&NrOfRedir=<Number
of redirects allowed(1-3)>][&EnableAbrLive=<truelfalse> Disable File Caching on Disk]
[&SkipLL=<truelfalse> Skip Location Leader Selection for Edge SE][&WmtUserAgent=<WMT User
Agent>][&QuotaUsageReport=<truelfalse> Force Quota Usage
Reporting][&genericSessionTrack=<false | true>][&hssSessionTrack=<false |
true>][&hlsSessionTrack=<false | true>]
```

## addManifest

Adds a Manifest file to a specified delivery service.

### Parameter

- Delivery service ID (required)
- Manifest URL (required)
- TTL (required)—In minutes
- User ID (optional)
- User password (optional)
- User domain name (optional)
- Not basic authentication (optional)—The default is false.
- No proxy (optional)—The default is false.
- Proxy IP address or host name (optional)
- Proxy port (optional)
- Proxy username (optional)
- Proxy password (optional)
- Proxy NTLM user domain name (optional)
- Proxy not basic authentication (optional)—The default is false.

### Return

The updated delivery service record.

### Syntax

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=addManifest&
deliveryService=<deliveryService_ID>&manifest=<manifest_URL>&ttl=<tll>
[&user=<user_name>][&password=<password>][&userDomainName=<user_domain_name>]
[&notBasicAuth=<truelfalse>][&noProxy=<true | false>][&proxyIpHostname=<proxy_ip_hostname>]
```

```
[&proxyPort=<proxy_port>][&proxyUser=<proxy_user>][&proxyPassword=<proxy_password>]
[&proxyNtlmUserDomainName=<proxy_ntlm_user_domain_name>][&proxyNotBasicAuth=
<true|false>]
```

## assignSEs

Assigns Service Engines to a specified delivery service.

This action need not be used if the [assignDeliveryService](#) action has already been used. If a delivery service has already been assigned to a program, the assignSEs action executes successfully but returns a warning message.

### Parameter

- Delivery service ID (required)
- Content Acquirer ID (required if no Content Acquirer is assigned; otherwise, this parameter is optional)
- Either a list of Service Engines or the keyword **all** is required (see the following rules).
- SE enable primed (optional)—Specifies the SEs (all or specific SE IDs) that are primed. Only valid when the delivery service is not a live delivery service.
- Either a list of clusters (cluster is the same thing as Service Engine) or the keyword **all** is required (see the following rules).

### Rules

- If a Service Engine list is set to **all**, a cluster list cannot be specified.
- If the cluster list is set to **all**, a Service Engine list cannot be specified.
- Both a Service Engine list and a cluster list cannot be set to **all** at the same time.

If these rules are violated, an error message is returned.



#### Note

---

A cluster is the same thing as a Service Engine.

---

### Return

None.



#### Note

---

The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine.

---

When assigning the Service Engine, specify one of the following options:

- List of Service Engines
- All Service Engines
- List of clusters
- All clusters
- List of Service Engines and clusters

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=assignSEs&
deliveryService=<deliveryService_ID>[&contentAcquirer=<contentAcquirer_ID>][&se=all |
<SE_ID>, <SE_ID>, ...][&se_enable_primed=all | <se_ID>, <se_ID> ...][&cluster=all | <Cluster_ID>,
<Cluster_ID>, ...]
```

**assignDeliveryServiceIp**

Assigns an IP address (IPv4 and IPv6) of a Service Engine to a single delivery service, a group of delivery services, or all delivery services to which the Service Engine belongs.

This action allows a delivery service to stream from an IP address configured on an interface of a Service Engine, while another delivery service streams from another IP address configured on the same interface of the Service Engine.

**Parameter**

- List of delivery service IDs or keyword "**all**" (required)
- IP address (required)
- Service Engine ID (required)

**Rules**

- IP address can be assigned to multiple delivery services, as long as the delivery services share the same content origin.
- IP address must be configured on an interface of the specified Service Engine.
- Service Engine must belong to the delivery services specified.

If these rules are violated, an error message is returned.

**Return**

None.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=assignDeliveryServiceIp&deliveryService=<all | deliveryService_ID,...>&ip=<IP Address>&se= <se_ID>
```

**fetchNow**

Immediately fetches the Manifest file.

Generally, the TTL (time-to-live) value of the Manifest is set to a reasonable value, such as 30 minutes. This servlet forces a freshness check of the Manifest file before the normal time-to-live interval expires on the delivery service specified. If the freshness check indicates that changes to the Manifest file have occurred, the Manifest file is parsed and the content processed. If you want the changes to the Manifest file to be processed immediately, use the fetchNow action.

**Parameter**

Delivery service ID (required)

**Return**

None.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=fetchNow&deliveryService=<deliveryService_ID>`

**modifyDeliveryService**

Modifies delivery service settings.

**Parameter**

- Delivery service ID (required)
- Name of the delivery service (optional)
- Content origin (optional)
- Weak certification (optional)
- Skip encryption (optional)
- Delivery service priority (optional)—The options are high, medium, or low. The default is medium.
- Multicast (optional)—The options are unicast\_only, multicast\_only, or multicast\_unicast. The default is unicast only.
- delivery service quota (optional)—Only valid for non-live delivery services
- Description (optional)
- FailoverIntvl (optional)
- Never (optional)
- Delivery service QoS (optional)
- sessionQuota (optional)—Session quota
- sessionQuotaAugBuf (optional)—Session quota augmentation buffer
- bandQuota (optional)—Bandwidth quota
- BandQuotaAugBuf (optional)—Bandwidth quota augmentation buffer
- StoragePriorityClass (optional)—Storage priority class ID

**Note**


---

The mcastEnable parameter is supported in Release 3.1.1.

---

**Note**


---

If a parameter is not specified, no change is made to the original delivery service settings.

---

**Return**

The updated delivery service record.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyDeliveryService&deliveryService=<deliveryService_ID>[&deliveryServiceName=<deliveryService_name>][&contentOrigin=<contentOrigin_ID>][&weakCert=<true|false>][&skipEncrypt=<true|false>][&priority=<high|medium|low>][&mcastEnable=<unicast_only|multicast_only|unicast_multicast>][&quota=<quota>][&qos=<system|0-63>][&desc=<description>][&failoverIntvl=<failoverIntvl,`

```
<20|30|40|50|60|70|80|90|100|110|120>][&never=<true|false>][&deliveryQos=<0-63>][&sessionQuota=<quota>][&sessionQuotaAugBuf=<0-1000>][&bandQuota=<quota>][&bandQuotaAugBuf=<0-1000>][&storagePriorityClass=<storagePriorityClass_ID>]
```

## modifyDeliveryServiceLoc

Modifies delivery service location.

### Parameter

- hssStreamingFromNas(option)—The default is false

### Return

The modified location object value.

### Syntax

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyDeliveryServiceLoc&deliveryServiceLoc=<deliveryServiceLoc_ID>[&hssStreamingFromNas=<false | true>]
```

## modifyDeliveryServiceGenSettings

Modifies delivery service general settings.

### Parameter

- deliveryService (required)—Delivery service ID
- Bitrate (optional)—Maximum bit rate limit per session for HTTP (0–2000000)
- OsProtocol ((optional)—Origin server streaming protocol support (0 means HTTP only support, 1 means HTTPS only support)
- StreamingProtocol ((optional)—Delivery streaming protocol support (0 means HTTP only support, 1 means HTTPS only support)
- HashLevel ((optional)—URL Hash Level for Cache Routing (0–10)
- TmpfsSize ((optional)—Memory Cache Size (1–10)
- OsHttpPort ((optional)—Origin Server HTTP Port for Web Engine (1–65535, except well-known port numbers), default is 80
- ReadTimeout ((optional)—HTTP Read Timeout (1–60)
- HttpAllow (optional)—Disable HTTP Download (True = disable, False = enable)
- ContentFlowTrace (optional)—Enable Content Flow Trace (True = enable)
- FilterTraceFlowToClient (optional)—Enable Filter Trace Flow to Client (True = enable)
- HttpExtAllow (optional)—Enable streaming over HTTP (True = enable)
- HttpExt (optional)—HTTP Allowed Extensions (invalid if HttpExtAllow is false)
- GreenCookie (optional)—Outgoing Cookie
- EnableCacheError (optional)—Enable Error Response Caching (True = enable)
- CacheError (optional)—Cacheable Error Responses (invalid if EnableCacheError is false)
- OSRedirectEnable (optional)—Follow Origin Server redirects (True - enable)
- NrOfReir (optional)—Number of redirects allowed (invalid if OSRedirectEnable is false)

- EnableAbrLive (optional)—Disable File Caching on Disk (True = enable)
- SkipLL (optional)—Skip Location Leader Selection for Edge SE (True = enabled)
- WmtUserAgent (optional)—WMT User Agent
- QuotaUsageReporting (optional)—Force quota usage reporting when bandwidth and session quotas are not configured for the delivery service
- genericSessionTrack(option)—The default is false
- hssSessionTrack(option)—The default is false
- hlsSessionTrack(option)—The default is false

**Note**

If the delivery service is a live delivery service, only WmtUserAgent is valid, all other parameters (except deliveryService) are not applicable for a live delivery service.

**Return**

The updated delivery service general settings record.

**Syntax**

```
https://<cdsmIpAddress>:
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyDeliveryServiceGenSettings&del
iveryService=<deliveryService_ID>[&Bitrate=<Maximum bitrate limit per session for HTTP(Kbps)
(0-2000000)>][&HashLevel=<URL Hash Level for Cache Routing(0-10)>][&TmpfsSize=<Memory
Cache Size(MB)(1-10)>][&OsHttpPort=<Origin Server HTTP Port(web-engine only,default 80)>]
[&ReadTimeout=<HTTP Response Read Timeout>][&OsProtocol=< Delivery streaming protocol
support(0 - HTTP only,1 - HTTPS only)>][&StreamingProtocol=<Origin Server streaming protocol
support(0 - HTTP only,1 - HTTPS only)>][&HttpAllow=<truelfalse> Disable HTTP Download]
[&ContentFlowTrace=<truelfalse> Enable Content Flow Trace][&FilterTraceFlowToClient=
<truelfalse> Enable Filter Trace Flow to Client][&HttpExtAllow=<truelfalse> Enable streaming over
HTTP][&HttpExt=<HTTP Allowed Extensions>][&GreenCookie=<Outgoing Cookie>]
[&EnableCacheError=<truelfalse> Enable Error Response Caching][&CacheError=<Cacheable Error
Responses>][&OSRedirectEnable=<truelfalse>Follow Origin Server redirects][&NrOfRedir=<Number
of redirects allowed(1-3)>][&EnableAbrLive=<truelfalse> Disable File Caching on Disk]
[&SkipLL=<truelfalse> Skip Location Leader Selection for Edge SE][&WmtUserAgent=<WMT User
Agent>][&QuotaUsageReport=<truelfalse> Force Quota Usage
Reporting][&genericSessionTrack=<false | true>][&hssSessionTrack=<false |
true>][&hlsSessionTrack=<false | true>]
```

**modifyManifest**

Modifies Manifest file settings.

**Parameter**

- Delivery service ID (required)
- Manifest URL (optional)
- TTL (optional)
- User ID (optional)
- User password (optional)
- NTLM user domain name (optional)



- Not basic authentication (optional)—The default is false.
- No proxy (optional)—The default is false.
- Proxy IP address or host name (optional)
- Proxy port (optional)
- Proxy username (optional)
- Proxy password (optional)
- Proxy NTLM user domain name (optional)
- Proxy not basic authentication (optional)—The default is false.

**Note**

If a parameter value is not specified, no change is made to the original Manifest file setting. If the parameter values need to be removed, use the “empty string” mechanism to delete an existing setting. For example, if a manifest was originally set for a delivery service and you now want to remove that manifest from the delivery service, set the manifest parameter to an empty string (manifest=“”) when using the modifyManifest action.

Setting a Manifest URL to null removes all the other settings.

**Return**

The updated delivery service record.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
modifyManifest&deliveryService=<deliveryService_ID>[&manifest=<manifest_URL>][&ttml=<ttml>][&
user=<user_name>][&password=<password>][&userDomainName=
<user_domain_name>][&notBasicAuth=<true | false>][&noProxy=<true | false>]
[&proxyIpHostname=<proxy_ip_hostname>][&proxyPort=<proxy_port>][&proxyUser=
<proxy_user>][&proxyPassword=<proxy_password>][&proxyNtlmUserDomainName=
<proxy_ntlm_user_domain_name>][&proxyNotBasicAuth=<true | false>]
```

## unassignSEs

Removes Service Engines from a specified delivery service.

This action need not be used if the [unassignDeliveryService](#) action has already been used. If a delivery service has already been assigned to a program, the unassignSEs action executes successfully but returns a warning message.

**Parameter**

- Delivery service ID (required)
- Either a list of Service Engines or the keyword **all** is required (see the following rules).
- Either a list of clusters (cluster is the same thing as Service Engine) or the keyword **all** is required (see the following rules).

**Rules**

- If a Service Engine list is set to **all**, a cluster list cannot be specified.
- If a cluster list is set to **all**, a Service Engine list cannot be specified.

- Both a Service Engine list and a cluster list cannot be set to **all** at the same time.

If these rules are violated, an error message is returned.

#### Return

None.



#### Note

The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine.

When removing the Service Engine from the delivery service, specify one of the following options:

- List of Service Engines
- All Service Engines
- List of clusters
- All clusters
- List of Service Engines and clusters

#### Syntax

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=unassignSEs&
deliveryService=<deliveryService_ID>[&se=all | <SE_ID>, <SE_ID>, ...][&cluster=all |
<Cluster_ID>, <Cluster_ID>, ...]
```

## unassignDeliveryServiceIp

Unassigns IP addresses of a Service Engine from a single delivery service or a group of delivery services. When an IP address of a Service Engine is unassigned from delivery services, any delivery service streaming on the IP address is interrupted.

#### Parameter

- List of delivery service IDs (required)
- Service Engine ID (required)

#### Rules

- All delivery services specified must share the same content origin.
- The Service Engine must belong to the delivery services specified.

If these rules are violated, an error message is returned.

#### Return

None.

#### Syntax

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=unassignDeliv
eryServiceIp&deliveryService=<deliveryService_ID,...>&se= <se_ID>
```

## deleteDeliveryServices

Deletes delivery services.

### Parameter

Either a list of delivery services or the keyword **all** is required.

### Return

None.

### Syntax

`https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteDeliveryServices&deliveryService=all | <deliveryService_ID>, <deliveryService_ID>, ...`

## deleteDeliveryServiceGenSettings

Deletes the general settings of a delivery service. After successful deletion of the general settings for the specified delivery service, the parameters are reset to the default values.

### Parameter

- `deliveryService` (required)

### Return

None (confirmation that the settings were deleted).

### Syntax

`https://<cdsMIpAddress>: 8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteDeliveryServiceGenSettings&deliveryService=<deliveryService_ID>`

## addContentItem

Creates and adds a content item to a delivery service.

### Parameter

- `deliveryService` (required)—Delivery service ID
- `ItemType` (required)—Content item type, the options are `singleItemType`, `crawlType`, `multipleItemType`.
- `SourceURL` (required)—URL of the content item, If "`itemType=multiItemType`" can provide upto 10 content urls as csv string.
- `Depth` (optional)—How many levels of a website to crawl or how many directory levels of an FTP server to crawl. The range is -1 to 2147483636.
- `HighPriority` (optional)—Acquisition of this content will take precedence, if (`highPriority = true`).
- `DisableBasicAuth` (optional)—Acquirer will not use basic authentication while fetching content, if (`disableBasicAuth = true`).
- `WeakCert` (optional)—Allow https protocol to accept expired or self-signed certificate, if (`weakCert = true`).

- StartServTime (optional)—Specifies the time for the SE to start delivering content. Use the format dd-mm-yyyy hh:mm:ss [TMZ] format, where TMZ (the time zone) is optional.
- StopServTime (optional)—Specifies the time for the SE to stop delivering content. Use the dd-mm-yyyy hh:mm:ss [TMZ] format, where TMZ (the time zone) is optional.
- Username (optional)—The username to log in to host servers that require authentication.
- Password (optional)—The password for the user account.
- NtlmUserDomain (optional)—NTLM user domain name for the NTLM authentication scheme.
- IgnoreQueryStr (optional)—If true, ignores any string after the question mark (?) character in the requested URL for playback.
- PlayDura (optional)—Play Duration.
- Ttl (optional)—Time period for revalidation of content. Select unit of measure from the drop-down list. If no TTL is entered, the content is fetched only once, and its freshness is never checked again (value in minutes).
- RetryInterval (optional)—Time period in which the Content Acquirer can attempt to acquire the content again if the acquisition fails(value in minutes).
- RequireAuth (optional)—Determines whether users need to be authenticated before the specified content is played. if (true=Requires, false=not required).
- rMimeType<1 to 5> (optional)—A content item is listed in the results only if its MIME type matches this MIME type (for example, video/mpeg).
- rExtension<1 to 5> (optional)—A content item is listed only if its extension matches this extension.
- rTimeBefore<1 to 5> (optional)—A content item is listed only if it was modified before this date. Click the Calendar icon to choose a date from the calendar, or enter the date in mm/dd/yyyy format.
- rTimeAfter<1 to 5> (optional)—A content item is listed only if it was modified after this date. Click the Calendar icon to choose a date from the calendar, or enter the date in mm/dd/yyyy format.
- rMinimumSize<1 to 5> (optional)—Content equal to or larger than this value is listed in the results. Choose MB, KB, or Bytes as the unit of measure. The range is 0 to 2147483636.
- rMaxSize<1 to 5> (optional)—Content equal to or less than this value is listed in the results. Choose MB, KB, or Bytes as the unit of measure. The range is 0 to 2147483636.

### Return

New content is created based on the parameters and will be added to the manifest file and cdn manifest xml will return.

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=addContentItem&deliveryService=<deliveryService_ID>&itemType=<item_type_of_content>[&sourceURL=<content_source_urls>][&depth=<link_depth_to_be_crawled>][&highPriority=<truefalse>][&disableBasicAuth=<truefalse>][&weakCert=<truefalse>][&startServTime=<MM/DD/YYYY
```

```

HH:MM:SS>][&stopServTime=<MM/DD/YYYYHH:MM:SS>][&username=<user_name_to_fetch_c
ontent>][&password=<password_to_fetch_content>][&ntlmUserDomain=<user_name_in_the_domain
>][&ignoreQueryStr=<truelfalse>][&playDura=<item_type_of_content>][&ttl=<frequency_of_rechec
king_in_minutes>][&retryInterval=<frequency_of_rechecking_if_acquisition_fails_in_minutes>][&re
quireAuth=<truelfalse>][&rMimeType<1_to_5>=<mime_type_of_acquisition_rule>][&rExtension<1_
to_5>=<file_extension_of_acquisition_rule>][&rTimeBefore<1_to_5>=<file_modified_before_MM/D
D/YYYY HH:MM:SS>][&rTimeAfter<1_to_5>=<file_modified_after_MM/DD/YYYY
HH:MM:SS>][&rMinimumSize<1_to_5>=<minimum_size_of_file_in_MB>][&rMaxSize<1_to_5>=<
maximum_size_of_file_in_MB>

```

## modifyContentItem

Modifies a content item to a delivery service.

### Parameter

- **DeliveryService** (required)—Delivery service ID.
- **SourceURL** (required)-URL of the content item, If "itemType=multiItemType" can provide upto 10 content urls as csv string.
- **Depth** (optional)—How many levels of a website to crawl or how many directory levels of an FTP server to crawl. The range is -1 to 2147483636.
- **HighPriority** (optional)—Acquisition of this content will take precedence, if (highPriority = true).
- **DisableBasicAuth** (optional)—Acquirer will not use basic authentication while fetching content, if (disableBasicAuth = true).
- **WeakCert** (optional)—Allow https protocol to accept expired or self-singed certificate, if (weakCert = true).
- **StartServTime** (optional)—Specifies the time for the SE to start delivering content. Use the format dd-mm-yyyy hh:mm:ss [TMZ] format, where TMZ (the time zone) is optional.
- **StopServTime** (optional)—Specifies the time for the SE to stop delivering content. Use the dd-mm-yyyy hh:mm:ss [TMZ] format, where TMZ (the time zone) is optional.
- **Username** (optional)—The username to log in to host servers that require authentication.
- **Password** (optional)—The password for the user account.
- **NtlmUserDomain** (optional)—NTLM user domain name for the NTLM authentication scheme.
- **IgnoreQueryStr** (optional)—If true, ignores any string after the question mark (?) character in the requested URL for playback.
- **PlayDura** (optional)—Play Duration.
- **ttl** (optional)—Time period for revalidation of content. Select unit of measure from the drop-down list. If no TTL is entered, the content is fetched only once, and its freshness is never checked again (value in minutes).
- **RetryInterval** (optional)—Time period in which the Content Acquirer can attempt to acquire the content again if the acquisition fails(value in minutes).
- **RequireAuth** (optional)—Determines whether users need to be authenticated before the specified content is played. if (true=Requires, false=not required).

- `rMimeType<1 to 5>` (optional)—A content item is listed in the results only if its MIME type matches this MIME type (for example, video/mpeg).
- `rExtension<1 to 5>` (optional)—A content item is listed only if its extension matches this extension.
- `rTimeBefore<1 to 5>` (optional)—A content item is listed only if it was modified before this date. Click the Calendar icon to choose a date from the calendar, or enter the date in mm/dd/yyyy format.
- `rTimeAfter<1 to 5>` (optional)—A content item is listed only if it was modified after this date. Click the Calendar icon to choose a date from the calendar, or enter the date in mm/dd/yyyy format.
- `rMinimumSize<1 to 5>` (optional)—Content equal to or larger than this value is listed in the results. Choose MB, KB, or Bytes as the unit of measure. The range is 0 to 2147483636.
- `rMaxSize<1 to 5>` (optional)—Content equal to or less than this value is listed in the results. Choose MB, KB, or Bytes as the unit of measure. The range is 0 to 2147483636.

### Return

The content is modified based on the parameters and will be added to the manifest file and cdn manifest xml will return.

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

### Syntax

`https://<cdsmIpAddress>`:

```
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyContentItem&deliveryService=<
deliveryService_ID>&sourceURL=<source_url_of_content>[&depth=<link_depth_to_be_crawled>][
&highPriority=<truelfalse>][&disableBasicAuth=<truelfalse>][&weakCert=<truelfalse>][&startServTi
me=<MM/DD/YYYY
HH:MM:SS>][&stopServTime=<MM/DD/YYYYHH:MM:SS>][&username=<user_name_to_fetch_c
ontent>][&password=<password_to_fetch_content>][&ntlmUserDomain=<user_name_in_the_domain
>][&ignoreQueryStr=<truelfalse>][&playDura=<item_type_of_content>][&ttl=<frequency_of_rechec
king_in_minutes>][&retryInterval=<frequency_of_rechecking_if_acquisition_fails_in_minutes>][&re
quireAuth=<truelfalse>][&rMimeType<1_to_5>=<mime_type_of_acquisition_rule>][&rExtension<1_
to_5>=<file_extension_of_acquisition_rule>][&rTimeBefore<1_to_5>=<file_modified_before_MM/D
D/YYYY HH:MM:SS>][&rTimeAfter<1_to_5>=<file_modified_after_MM/DD/YYYY
HH:MM:SS>][&rMinimumSize<1_to_5>=<minimum_size_of_file_in_MB>][&rMaxSize<1_to_5>=<
maximum_size_of_file_in_MB>]
```

## deleteContentItem

Deletes content items.

### Parameter

- Delivery service ID (required)
- URL of content items to be deleted (required)—

### Return

Error message is returned in each of below conditions:

- Mandatory parameter is missing.

- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

https://<cdsmIpAddress>:

8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteContentItem&deliveryService=<deliv  
eryService\_ID>&deleteItemUrls=<csv\_list\_of\_source\_url\_to\_delete>

## processContentChanges

Processes the content changes to the delivery service.

**Parameter**

- Delivery service ID (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

https://<cdsmIpAddress>:

8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=processContentChanges&deliveryService=<deliv  
eryService\_ID>

## manageHostProxySettings

**Parameter**

- Delivery service ID (required)
- Host name (required)
- Proxy server name (required)
- Proxy port number (required)
- Disable basic authentication (required)
- Username (required)
- Password (required)
- Remove proxy setting (optional)
- Assign proxy setting (optional)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.

- Invalid value of parameter is given.

**Syntax**

https://<cdsmIpAddress>:

```
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=manageHostProxySettings&deliveryService=<deliveryService_ID>&hosts=<host>&proxyServerName=<proxy_server_name>&proxyPort=<proxy_server_port>&disableBasicAuth=<truefalse>&username=<user_name_of_proxy_server>&password=<password_to_proxy_server>[&remove=<truefalse>][&assignProxy=<truefalse>]
```

**createContentOrigin**

Creates a content origin.

**Parameter**

- Content origin name (required)
- Origin server (required)
- Fully qualified domain name (FQDN) (required)




---

**Note** This is the FQDN used by the Service Router to route the requests to a Service Engine. For example, while processing a request for http://www.cnn.com (origin server FQDN), the Service Router may route the request to a Service Engine using the FQDN http://cdn.cnn.com.

---

- Enable content-based routing (optional)—The default is true.
- Network Attached Storage (NAS) file ID—The format is FileInfo\_xxx, where xxx is the file ID. The other option is to enter "none," for example, [&nasFile=none].




---

**Note** NAS is only supported in lab integrations as proof of concept.

---

- WMT authentication (optional)—The default is none.
  - None
  - Basic
  - NTLM
  - Digest
  - Negotiate
- httpAuthType (optional)—HTTP Authentication Type (**none**, **basic**, or **challenged**)
- httpAuthHeader (optional)—Authentication header
- httpAuthSharedKey (optional)—Authentication shared key (16–128 TEXT characters as defined in RFC 2616))
- httpAuthHeaderPrefix (optional)—Authentication header prefix
- httpAuthSharedSecKey (optional)—Authentication shared secret key (16–128 TEXT characters as defined in RFC 2616))
- httpAuthHashFunc (optional)—Hashing function (only **MD5** is supported)
- Description (optional)



**Return**

A confirmation that the new content origin has been created and the newly created content origin object has been saved.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
createContentOrigin&name=<contentorigin_name>&origin=<origin_server_IP_or_domain>
&fqdn=<fqdn>[&contentBasedRouting=<true | false>][&nasFile=<FileInfo_id | none>]
[&wmtAuth=<basic | ntlm | digest | negotiate>][&description=<description>]
[&httpAuthType=<nonelbasiclchallenged>][&httpAuthHeader=<auth_header>][&httpAuthSharedKey
=<auth_shared_key>][&httpAuthHeaderPrefix=<auth_header_prefix>][&httpAuthSharedSecKey=
<auth_shared_secret_key>][&httpAuthHashFunc=<MD5>] [&description=<description>]
```

**modifyContentOrigin**

Modifies content origin settings.

**Parameter**

- Content origin ID (required)
- Content origin name (optional)
- Origin server (optional)
- FQDN (optional)
- Enable content-based routing (optional)—The default is true.
- NAS file ID. The format is FileInfo\_xxx, where xxx is the file ID. The other option is to enter "none," for example, [`&nasFile=none`]




---

**Note** NAS is only supported in lab integrations as proof of concept.

---

- WMT authentication (optional)
  - None
  - Basic
  - NTLM
  - Digest
  - Negotiate
- httpAuthType (optional)—HTTP Authentication Type (**none**, **basic**, or **challenged**)
- httpAuthHeader (optional)—Authentication header
- httpAuthSharedKey (optional)—Authentication shared key (16–128 TEXT characters as defined in RFC 2616))
- httpAuthHeaderPrefix (optional)—Authentication header prefix
- httpAuthSharedSecKey (optional)—Authentication shared secret key (16–128 TEXT characters as defined in RFC 2616))
- httpAuthHashFunc (optional)—Hashing function (only **MD5** is supported)
- Description (optional)

**Return**

A confirmation that content origin attributes have been modified and an updated record for the content origin object.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
modifyContentOrigin&contentOrigin=<contentorigin_ID>[&name=<contentorigin_name>]
[&origin=<origin=<origin_server_IP_or_domain>][&fqdn=<fqdn>]
[&contentBasedRouting=<true|false>] [&nasFile=<<FileInfo_id | none>][&wmtAuth=<none | basic |
ntlm | digest | negotiate>][&description=<description>]
[&httpAuthType=<none|basic|challenged>][&httpAuthHeader=<auth_header>][&httpAuthSharedKey=
<auth_shared_key>][&httpAuthHeaderPrefix=<auth_header_prefix>][&httpAuthSharedSecKey=
<auth_shared_secret_key>][&httpAuthHashFunc=<MD5>] [&description=<description>]
```

**deleteContentOrigin**

Deletes content origins.

**Parameter**

Either a list of content origin IDs or the keyword **all** is required.

**Return**

A confirmation that the content origins have been deleted.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
deleteContentOrigins&contentOrigin=all | <contentorigin_ID>, <contentorigin_ID>, ...
```

**applyRuleFile**

Assigns a Service Rule file to a delivery service or unassigns a Service Rule file from a delivery service.

**Parameter**

- Delivery service ID (required)—The format is Channel\_xxx, where xxx is the ID of the delivery service.
- Rule file ID (required). Valid values are:
  - None—Unassigns the Rule file from the delivery service.
  - File ID—The format is FileInfo\_xxx, where xxx is the file ID.

**Return**

Confirmation that the Service Rule file has been assigned to the delivery service or unassigned from the delivery service.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=applyRuleFile
&deliveryService=<Channel_<xxx>&ruleFile=<none | FileInfo_ID>
```

**Note**

The `applyRuleFile` action expects the `deliveryService` parameter to be in the form `Channel_<xxx>`, where `xxx` is the ID of the delivery service.

## applyGeoIpFile

Assigns a Geo/IP file to a delivery service or unassigns a Geo/IP file from a delivery service.

**Parameter**

- Delivery service ID (required)—The format is `Channel_<xxx>`, where `xxx` is the ID of the delivery service.
- Geo/IP file ID (required). Valid values are:
  - None—Unassigns the Geo/IP file from the delivery service.
  - File ID—The format is `FileInfo_<xxx>`, where `xxx` is the file ID.

**Return**

Confirmation that the Geo/IP file has been assigned to the delivery service or unassigned from the delivery service.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
applyGeoIpFile&deliveryService=<Channel_<xxx>&geoIpFile=<none | geoIpFile_ID>
```

**Note**

The `applyGeoIpFile` action expects the `deliveryService` parameter to be in the form `Channel_<xxx>`, where `xxx` is the ID of the delivery service.

## configFailoverSettings

**parameter**

- `contentOrigin` (required)
- `failoverEnabled` (optional)
- `failureAlarmDuration` (optional)—The default is 5 minutes.
- `recoveryAlarmDuration` (optional)—The default is 5 minutes.

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=configFailoverSettings&contentOrigin=
<contentOrigin_ID>[&failoverEnabled=<false |
true>][&failureAlarmDuration=<0-525600>][&recoveryAlarmDuration=<0-525600>]
```

**createFailoverOS****Parameter**

- contentOrigin (required)
- fqdn (optional)—It should be unique and cant be same as OFQDN of Content Origin.
- failureDetectTimeout (optional)—The default is 5 seconds.
- failureDetectRetry (optional)—The default is 0 second.
- priority (optional)—The default is 500, 1 is the highest and 1000 is the lowest.

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createFailoverOS&contentOrigin=<cont
entOrigin_ID>&fqdn=<FQDN of Origin
Server>&failureDetectTimeout=<1-255>&failureDetectRetry=<0-255>&priority=<1-1000>
```

**modifyFailoverOS****Parameter**

- originServer (required)
- fqdn (optional)—It cannot be modified for Primary Origin Server.
- failureDetectTimeout (optional)—The default is 5 seconds.
- failureDetectRetry (optional)—The default is 0 second.
- priority (optional)—The default is 500, 1 is the highest and 1000 is the lowest.

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:  
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyFailoverOS&originServer=<originServer_ID>[&fqdn=<FQDN of Origin Server>][&failureDetectTimeout=<1-255>][&failureDetectRetry=<0-255>][&priority=<1-1000>]
```

## deleteFailoverOS

**Parameter**

- originServer (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:  
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteFailoverOS&originServer=<originServer_ID>
```

## switchToOS

**Parameter**

- contentOrigin (required)
- originServer (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:  
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=switchToOS&contentOrigin=<contentOrigin_ID>&originServer=<originServer_ID>
```

## createChannelDeviceMcastConfig

**Parameter**

- Delivery service ID (required)
- SE device ID (required)
- Multicast sender configuration (required)

- Multicast receiver configuration (required)
- Unicast sender configuration (optional)
- Unicast receiver configuration (optional)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=
createChannelDeviceMcastConfig&deliveryService=<Channel_id>&seDevice=<CeConfig_id>&sende
rEnabled=<truelfalse>&receiverEnabled
=<truelfalse>[&unicastSenderEnabled=<truelfalse>][&unicastReceiverEnabled =<truelfalse>]
```

**getChannelDeviceMcastConfig****Parameter**

- Delivery service ID (required)
- SE device ID (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=getChannelDe
viceMcastConfig&deliveryService=<Channel_id>&seDevice=<CeConfig_id>
```

**modifyChannelDeviceMcastConfig****Parameter**

- Configuration ID (required)—Example of format for file ID=220 is &channelDeviceMcastConfig=220
- Delivery service ID (required)—The format is Channel\_xxx, where xxx is the ID of the delivery service
- SE device ID (required)—The format is CeConfig\_xxx, where xxx is the ID of the device
- Multicast sender configuration (required)
- Multicast receiver configuration (required)
- Unicast sender configuration (required)

- Unicast receiver configuration (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyChannelDeviceMcastConfig&channelDeviceMcastConfig=<channelDeviceMcastConfig_id>&deliveryService=<deliveryService_id>&seDevice=<se_id>&senderEnable=<true|false>&receiverEnable=<true|false>&unicastSenderEnabled=<true|false>&unicastReceiverEnabled=<true|false>
```

## deleteChannelDeviceMcastConfig

**Parameter**

- Configuration ID (required)

**Return**

Error message is returned in each of below conditions:

- Mandatory parameter is missing.
- Invalid parameter is given.
- Invalid value of parameter is given.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteChannelDeviceMcastConfig&channelDeviceMcastConfig=<channelDeviceMcastConfig_id>
```

## Location Provisioning API Actions

The Location Provisioning API is the LocationApiServlet.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.LocationApiServlet...
```

This servlet performs one or more of the following actions:

- [createLocation](#)
- [modifyLocation](#)
- [deleteLocation](#)

### createLocation

Creates a specified location.

**Parameter**

- Location name (required)
- Parent location ID (optional)
- Description (optional)

**Return**

The newly created location object.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.LocationApiServlet?action=
createLocation&location=<location_name>[&parent=<parent_ID>][&desc=<description>]
```

## modifyLocation

Modifies a specified location.

**Parameter**

- Location ID (required)
- Location name (optional)
- Parent location ID (optional)
- Description (optional)

**Return**

The modified location object.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.LocationApiServlet?action=
modifyLocation&location=<location_ID>[&name=<location_name>][&parent=<parent_ID>]
[&desc=<description>]
```

## deleteLocation

Deletes a specified location.

**Parameter**

Location ID (required)

**Return**

A message that the specified location has been deleted.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.LocationApiServlet?action=
deleteLocation&location=<location_ID>
```



## Service Engine Provisioning API Actions

The Service Engine Provisioning API is the CeApiServlet.

### Syntax

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet...`

This servlet performs one or more of the following actions:

- [activateSe](#)
- [changeSeLocation](#)
- [deleteSe](#)
- [setSeMgmtIp](#)
- [setMulticast](#)

### activateSe

Activates a specified Service Engine.

#### Parameter

- Service Engine ID (required)
- Location ID (required)

#### Return

The modified Service Engine object.

#### Syntax

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet?action=activateSe&se=<SE_ID>&location=<location_ID>`

### changeSeLocation

Changes the location of a specified Service Engine.

#### Parameter

- Service Engine ID (required)
- Location ID (required)

#### Return

The modified Service Engine object.

#### Syntax

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet?action=changeSeLocation&se=<SE_ID>&location=<location_ID>`

### deleteSe

Deletes a specified Service Engine.

**Parameter**

Service Engine ID (required)

**Return**

A message that the specified Service Engine has been deleted.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet?action=deleteSe&se=
<SE_ID>
```

**setSeMgmtIp**

Sets the IP address of the management communication on a specified Service Engine.

**Parameter**

- Service Engine ID (required)
- Management type (required)—Value of 1 sets the management IP address as the primary interface IP address. Value of 2 means to manually configure the management IP address
- IP address (required for manually configured)

**Return**

The modified Service Engine object.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet?action=setSeMgmtIp&se=
<se_ID>&mgmtIpType=<1 - Use Primary Interface | 2 - Manually
Config>&mgmtIp=<management_IP(required if mgmtIpType == 2)>
```

**setMulticast**

Enables an SE as a multicast sender and multicast receiver.

**Parameter**

- Service Engine (required)—CeConfig\_Id is the same as the se\_ID
- Enable Sender (optional)
- Enable Receiver (optional)

**Return**

The modified Service Engine object.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CeApiServlet?action=setMulticast&se=
<CeConfig_Id>[&enableSender=truelfalse][&enableReceiver=truelfalse]
```

## Program API Actions

The Program API is the ProgramApiServlet.

### Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet...



### Note

You must have administrator-level access privileges to execute Program API actions.

This servlet performs one or more of the following actions:

- [createProgram](#)
- [validateProgramFile](#)
- [assignDeliveryService](#)
- [assignSEs](#)
- [fetchNow](#)
- [modifyProgramFile](#)
- [unassignDeliveryService](#)
- [unassignSEs](#)
- [deletePrograms](#)

## createProgram

Fetches a program file using HTTP, validates it, and creates a program based on the input. This action also reserves a multicast address, if the program requires one. The multicast address reserved for the program is not released until the program is deleted.

### Parameter

- Program file URL (required)
- Update interval (required)—Interval (in minutes) at which to access the program file to check for updates
- User ID (optional)
- User password (optional)

### Return

The newly created program record with the program ID. If the program file fails validation, an error message is returned.

[Appendix A, “Program Files in the Videoscape Distribution Suite, Internet Streamer Software,”](#) provides a DTD for the information that is returned.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=
createProgram&file=<program_file_URL>&updateInterval=<update_interval_minutes>[&user=
<user_name>][&password=<password>]
```

## validateProgramFile

Fetches a program file using HTTP and validates it.

### Parameter

Program file URL (required)

### Return

None, if there are no errors. If there are errors, returns a list of errors.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=
validateProgramFile&file=<program_file_URL>
```

## assignDeliveryService

Assigns a delivery service to a program. When you assign a delivery service to a program, all Service Engines associated with the delivery service are associated with the program. Any modification to the Service Engine delivery service assignment also updates the program.

This action should not be used if the [assignSEs](#) action has already been used. If a Service Engine has already been assigned to a program, the `assignDeliveryService` action fails and returns the following error message:

```
<?xml version="1.0" ?>
- <programApi action="assignDeliveryService">
  <message status="fail" message="Constraint Error: Can not associate a delivery service
with the playlist. Service Engines are already assigned to the playlist." />
  <error code="3" message="Constraint Error: Can not associate a delivery service with the
playlist. Service Engines are already assigned to the playlist." />
</programApi>
```

### Parameter

- Program ID (required)
- Delivery service ID (required)

### Return

The updated program record.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=
assignDeliveryService&program=<program_ID>&deliveryService=<deliveryService_ID>
```

## assignSEs

Assigns Service Engines to a program.

This action should not be used if the [assignDeliveryService](#) action has already been used. If a delivery service has already been assigned to a program, the `assignSEs` action fails and returns the following error message:

```
<?xml version="1.0" ?>
- <programApi action="assignSEs">
```

```
<message status="fail" message="Constraint Error: Can not assign Service Engines to the
playlist. The playlist is already associated with a delivery service." />
<error code="3" message="Constraint Error: Can not assign Service Engines to the
playlist. The playlist is already associated with a delivery service." />
</programApi>
```

**Note**

This action fails if the program represents a live event, because live programs must be assigned to a live delivery service.

**Parameter**

- Program ID (required)
- Either a list of Service Engines or the keyword **all** is required.

**Return**

The updated program record.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=assignSEs&program=<program_ID>&se=all | <SE_ID>, <SE_ID>, ...`

## fetchNow

Fetches a program file immediately using HTTP and updates the program.

**Parameter**

Program ID (required)

**Return**

None, if there are no errors. Displays an error message if the program file fails validation.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=fetchNow&program=<program_ID>`

## modifyProgramFile

Modifies program file settings.

**Parameter**

- Program ID (required)
- Program file URL (optional)
- Update interval (optional)
- User ID (optional)
- User password (optional)

**Note**

If a parameter value is not specified, no change is made to the original program file setting. If the parameter values need to be removed, use the “empty string” mechanism to delete an existing setting. For example, if you now want to remove the user ID from the program file, set the user ID parameter to an empty string (user=“”) when using the modifyProgramFile action.

**Note**

You cannot set the program file URL to an empty string. Setting the program file URL to null removes all the other settings.

**Return**

The updated program record.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=
modifyProgramFile&program=<program_ID>[&file=<program_file_URL>][&updateInterval=
<update_interval>][&user=<user_name>][&password=<password>]
```

## unassignDeliveryService

Removes a delivery service from the specified program.

This action should not be used if the [unassignSEs](#) action has already been used. The unassignDeliveryService action executes successfully even if a Service Engine has already been unassigned from a program, but displays a warning that the delivery service is not assigned to the program.

**Parameter**

- Program ID (required)
- Delivery service ID (required)

**Return**

The updated program record.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=
unassignDeliveryService&program=<program_ID>&deliveryService=<deliveryService_ID>
```

## unassignSEs

Removes Service Engines from the specified program.

This action need not be used if the [unassignDeliveryService](#) action has already been used. The unassignSEs action executes successfully even if a delivery service has been already unassigned from a program, but displays a warning that the Service Engines are not assigned to the program.

**Parameter**

- Program ID (required)
- Either a list of Service Engines or the keyword **all** is required.

**Return**

The updated program record.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=unassignSEs&program=<program_ID>&se=all | <SE_ID>, <SE_ID>, ...`

## deletePrograms

Deletes programs.

**Parameter**

A list of programs by service type (such as WMT or Movie Streamer) or program ID, or the keyword **all** is required.

**Return**

None.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=deletePrograms&program=all | id=<program_ID>, <program_ID>, ... | type=<wmt | MovieStreamer>`

## Media API Actions for Programs

The Media API is the SelectMediaApiServlet, which is used to update the media lists for Movie Streamer rebroadcasts.

**Note**

If the Media API is used to change the media list and there is an associated program file (XML file) that is used with the Program API, the program file must be updated with the changed media list. Use the getPrograms List API action to get the media list for the program, and insert this list in the XML program file. For more information, see the [“getPrograms” section on page 3-59](#).

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SelectMediaApiServlet...`

**Note**

You must have administrator-level access privileges to execute Media API actions.

This servlet performs one or more of the following actions:

- [addMedia](#)
- [updateMedia](#)
- [deleteMedia](#)

## addMedia

Adds a media file to the end of the media list of a Movie Streamer rebroadcast program. Use the `getContent` Replication API action ([“getContent” section on page 3-4](#)) to get the list of prefetched content, then choose the media file to add.

### Parameters

- Program ID (required)—In the format "Playlist\_XXX," where *xxx* is an integer.
- Delivery Service ID (required)—In the format "Channel\_XXX," where *xxx* is an integer.
- File URL (required)
  - Example 1: [protocol]://myhost/myfile.mp4, where the *protocol* is http, https, or ftp. Protocol is optional.
  - Example 2: //myserver/folder/myfile.mp4

### Return

The newly created program record with the added media file. If the action parameter is missing, or cannot be recognized, the API usage is returned.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SelectMediaApiServlet?action=addMedia
&program=<Program_ID>&deliveryService=<deliveryService_id>&url=<file_url>
```

## updateMedia

Updates the order of the media files in a Movie Streamer rebroadcast program.



### Note

The list of media files in the update action must have the same media IDs and number of objects as the media list returned by the `getPrograms List` API action. The list of media files must not contain media files that have not been assigned to the program, and must not omit any media files that have been assigned to the program. For more information on getting the list of media files, see the [“getPrograms” section on page 3-59](#).

### Parameter

- Program ID (required)—In the format "Playlist\_XXX," where *xxx* is an integer.
- Media file list (required)

The media file list must be uploaded by posting as a multipart/form-data request. The Document Type Definition (DTD) for the media file list follows:

```
<?xml version="1.0"?>
<!DOCTYPE media_list[
<!ELEMENT media_list (media+)>
<!ELEMENT media EMPTY>
<!ATTLIST media
index CDATA #IMPLIED // List Order
id CDATA #IMPLIED // Playlist Media ID
>
]>
```



**Return**

The updated program record with the new media file list. If the action parameter is missing, or cannot be recognized, the API usage is returned. If any parameter value is not valid; for example, the media file list does not have the same media IDs and number of media files assigned to the program, an error message is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SelectMediaApiServlet?action=updateMedia&program=<Program_ID>
```

## deleteMedia

Deletes a media file from the Movie Streamer rebroadcast program. The deleteMedia action can only delete a media file that is not being streamed.

**Parameters**

- Playlist media file ID (required)—In the format "PlayListMedia\_XXX," where *xxx* is an integer.

**Return**

None (confirmation that the settings were deleted). If the action parameter is missing, or cannot be recognized, the API usage is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SelectMediaApiServlet?action=deleteMedia&id=<PlaylistMedia_id>
```

## URL Management API Actions

The URL Management API is the `UrlManagementApiServlet`.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.UrlManagementApiServlet...
```

This servlet performs one or more of the following actions:

- [singleURLRemoval](#)
- [batchURLRemoval](#)

### singleURLRemoval

Removes content items from delivery service based on a specified URL. The details for each content removal request are displayed.

**Parameter**

Single URL (required)

**Return**

200 Ok—Content URL removal is successful on all Service Engines.

500 Failed to communicate with SE at IP: <SE IP addr>—Please ensure the SE is online and the Centralized Management System (CMS) processes are running. The CLI **show cms processes** command can be used for viewing the status of the CMS processes and **cms enable** for enabling the CMS.

500 Failed to remove the content from the SE at IP: <SE IP addr> | 200 Ok—Content URL(s) removal is successful on the Service Engines with the following IPs: <SE IP addr1, SE IP addr2, ...>

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.UrlManagementApiServlet?
action=singleURLRemoval&singleUrl=<url>
```

## batchURLRemoval

Removes content items from the delivery service based on a specified set of URLs. The details for each content removal request are displayed.

### Parameter

Batch URL (required)

### Return

None.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.UrlManagementApiServlet?
action=batchURLRemoval<Only programmed API call allowed>
```



### Note

The batchURLRemoval requires a programmed API call; it does not work as an interactive API call.

Following is an example of Java code that can be used to call the batchURLRemoval API. Java Development Kit (JDK) 1.6 or higher is required to compile and use this Java code example.

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

public class BatchURLDemo {

    public static class newHostNameVerifier implements HostnameVerifier {
        /**
         * ignore hostname checking
         */
        public boolean verify(String hostname, SSLSession session) {
            return true;
        }
    }
}
```

```

    }

    public static void main(String args[]) {
        try {

            String userName_ = "admin"; /* CDSM user name*/
            String password_ = "default"; /* CDSM password name*/
            String cdsmAddress_ = "10.77.153.98"; /* CDSM IP address OR hostname */
            String apiServlet = "UrlManagementApiServlet"; /* API servlet name to call */
            String action = "batchURLRemoval"; /* API action name to call */
            String urlsFile = "C:\\batchremoval.xml"; /* The path for URLs XML file */
            int cdsmPort_ = 8443; /* CDSM https port number */

            /**
             * Create a trust manager that does not validate certificate chains
             */
            TrustManager[] trustAllCerts = new TrustManager[] { new X509TrustManager() {
                public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                    return null;
                }

                public void checkClientTrusted(
                    java.security.cert.X509Certificate[] certs,
                    String authType) {
                    /**
                     * do any special handling here, or re-throw exception.
                     */
                }

                public void checkServerTrusted(
                    java.security.cert.X509Certificate[] certs,
                    String authType) {
                    /**
                     * Possibly pop up a dialog box asking whether to trust the cert chain
                     */
                }
            } };

            /**
             * Install the all-trusting trust manager
             */
            SSLContext sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new java.security.SecureRandom());
            HttpURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());

            String sAuth = userName_+"."+password_;
            String sEncodedAuth = new sun.misc.BASE64Encoder().encode(sAuth.getBytes());

            URL url = new URL(null,
                "https://" + cdsmAddress_ + ":" + cdsmPort_ + "/servlet/com.cisco.unicorn.ui." +
                    apiServlet + "?action="+action);

            HttpURLConnection conn = null;
            DataOutputStream dos = null;

            String lineEnd = "\r\n";
            String hyphenLiteral = "--";
            String mPartBoundary = "*****";

            int maxBufferSize = 1024 * 1024;
            int bytesRead, bytesAvailable, bufferSize;
            byte[] buffer;

```

```

try {
    /**
     * initialize the HTTPS connection with post method
     */
    FileInputStream fileInputStream = new FileInputStream(new File(
        urlsFile));
    conn = (HttpsURLConnection) url.openConnection();
    conn.setRequestProperty("Authorization", "Basic "
        + sEncodedAuth);
    conn.setHostnameVerifier(new newHostNameVerifier());
    conn.setDoInput(true);
    conn.setDoOutput(true);
    conn.setUseCaches(false);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Connection", "Keep-Alive");
    conn.setRequestProperty("Content-Type",
        "multipart/form-data;boundary=" + mPartBoundary);
    dos = new DataOutputStream(conn.getOutputStream());
    dos.writeBytes(hyphenLiteral + mPartBoundary + lineEnd);
    dos
        .writeBytes("Content-Disposition: form-data; name=\"upload\";"
            + " filename=\""
            + urlsFile
            + "\"
            + lineEnd);
    dos.writeBytes(lineEnd);

    /**
     * load the URL xml file and upload it to server
     */
    bytesAvailable = fileInputStream.available();
    bufferSize = Math.min(bytesAvailable, maxBufferSize);
    buffer = new byte[bufferSize];

    bytesRead = fileInputStream.read(buffer, 0, bufferSize); // write

    while (bytesRead > 0) {
        dos.write(buffer, 0, bufferSize);
        bytesAvailable = fileInputStream.available();
        bufferSize = bytesAvailable;
        bytesRead = fileInputStream.read(buffer, 0, bufferSize);
    }

    dos.writeBytes(lineEnd);
    dos.writeBytes(hyphenLiteral + mPartBoundary + hyphenLiteral
        + lineEnd);

    fileInputStream.close();
    dos.flush();
    dos.close();
    catch (MalformedURLException ex) {
    System.out.println("Printing Exception Message " + ex);
    catch (IOException ioexception) {
    System.out.println("Printing Exception Message " + ioexception);
    }

}

/**
 * Handling the response from CDSM
 */
try {
    BufferedReader inStreamReader = new BufferedReader(
        new InputStreamReader(conn.getInputStream()));
    String str;
    while ((str = inStreamReader.readLine()) != null) {

```

```
        System.out.println("Response from CDSM : ");
        System.out.println(str);
    }
    inStreamReader.close();
    catch (IOException ioexception) {
        System.out.println("Printing Exception Message " + ioexception);
    }

    catch (Exception e) {
        System.out.println("Printing Exception Message " + e);
        e.printStackTrace();
    }
}
}
```

If the above Java code was saved in a file called “BatchRULDemo.java,” then to compile the code you would enter the **javac BatchURLDemo.java** command, and to run the script you would enter the **java BatchURLDemo** command.

```
javac BatchURLDemo.java
java BatchURLDemo
```

The following is an example of the XML file that is used in the Java code:

```
<?xml version="1.0" encoding="UTF-8"?>
<URLRemovalList xmlns='http://cisco.com/unicorn/cds/urlmgmt'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
<url-entry>http://2.2.23.32/Thursday.html</url-entry>
<url-entry>http://2.2.23.32/Hello.html</url-entry>
</URLRemovalList>
```

The following is an example of the output returned for the above Java code:

```
<?xml version="1.0"?><URLManagement action="batchURLRemoval"><message status="success"
message="200 OK - Content URL(s) removal is successful on all streaming
engines."/></URLManagement>
```

The details for each content removal request is displayed.

## Cache Storage Priority Class API Actions

The Cache Storage Priority Class API is the StoragePrioClassApiServlet.

### Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.StoragePrioClassApiServlet...

This servlet performs one or more of the following actions:

- [createStoragePrioClass](#)
- [modifyStoragePrioClass](#)
- [deleteStoragePrioClass](#)

### createStoragePrioClass

Creates a storage priority class.

**Parameter**

- Name—Class name (required)
- Factor—Storage multiplication factor (required)
- Comments—Comments (optional)

**Return**

The newly created storage priority class with a StoragePriorityClass ID.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.StoragePrioClassApiServlet?action=
createStoragePrioClass&name=<Class_Name>&factor=<storage_popularity_factor>[&comments=<co
mments>]
```

**modifyStoragePrioClass**

Modifies a storage priority class.

**Parameter**

- Storage priority class—Record ID of the storage priority class (required)
- Name—Class name (optional)
- Factor—Storage multiplication factor (optional)
- Comments—Comments (optional)

**Return**

The modified storage priority class.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.StoragePrioClassApiServlet?action=
modifyStoragePrioClass&storagePriorityClass=<storagePriorityClass_ID>[&name=<Class_Name>&f
actor=<storage_popularity_factor>&comments=<comments>]
```

**deleteStoragePrioClass**

Deletes a storage priority class.

**Parameter**

- Storage priority class—Record ID of the storage priority class (required)

**Return**

None.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.StoragePrioClassApiServlet?action=
deleteStoragePrioClass&storagePriorityClass=<storagePriorityClass_ID>
```

## Multicast Cloud API Actions

The Multicast Cloud API is the MCastApiServlet.

### Syntax

`https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet...`

This servlet performs one or more of the following actions:

- [createCloud](#)
- [modifyCloud](#)
- [deleteCloud](#)
- [assignReceiverSe](#)
- [unassignReceiverSe](#)
- [assignDeliveryService](#)
- [unassignDeliveryService](#)
- [modifyChannelMCast](#)

### createCloud

Creates a multicast cloud.

#### Parameter

- Name—Multicast cloud name (required)
- Advertisement IP—Unique advertisement address (required)
- Port—Port used for file addresses (required)
- Start IP address—Start of the IP address range, which must be within the range 224.0.0.0 to 239.255.255.255 (required)
- End IP address—End of the IP address range (required)
- Primary sender SE—Primary sender SE (required)
- Default multicast out bandwidth—Maximum multicast rate in kilobits per second (required)
- Multicast medium—Means of transmitting the multicast (Satellite or Terrestrial). Satellite is default (optional)
- FEC transmission group—Size of the FEC (forward error correction) block in packets. Allowable inputs are 2, 4, 8, 16, 32, 64, and 128. Default is 16. (optional)
- Carousel passes—Maximum number of times a multicast sender sends missing content (optional)
- Carousel delay—Delay, in minutes, between file transmissions (optional)
- Backup sender SE—Backup sender SE (optional)
- Failover grace period—Period of time backup sender goes without getting heartbeat from primary sender before taking over (optional)
- Fallback grace period—Period of time primary sender goes without getting heartbeat from backup sender before taking over (optional)
- PGM router assist—True means IP routers are used to assist in distribution of content. (optional)

- Description—(optional)
- defaultDSDDataRate—Default multicast data rate for delivery services in kilo bits per second (optional)
- maxConcurrentSessions—Maximum concurrent session (optional)

**Note**

The primarySenderSe and backupSenderSe needs a clusterId; for example, ClusterConfig\_2221. The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine. The Listing API can be used to get the cluster ID. For more information, see the “getSEs” section on page 3-56 and “getClusters” section on page 3-56.

**Return**

The newly created multicast cloud.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=createCloud&name=<cloud_name>&advertisementIp=<advertisement_ip>&port=<port>&startIp=<start_ip>&endIp=<end_ip>&primarySenderSe=<primary_sender_SE_cluster_id>&defaultMOutBandwidth=<default_multicast_bandwidth>[&medium=<satellite/terrestrial>][&fecTransmissionGroup=<fec_transmission_group>][&carouselPass=<carousel_pass>][&carouselDelay=<carousel_delay>][&tTl=<tTl>][&backupSenderSe=<backup_sender_SE_cluster_id>][&failoverGrace=<failover_grace_period>][&fallbackGrace=<fallback_grace_period>][&pgmRouterAssist=<true/false>][&description=<description>][&defaultDSDDataRate=<default_DS_data_rate>][&maxConcurrentSessions=<Maximum_concurrent_sessions>]
```

## modifyCloud

Modifies a multicast cloud.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- Name—Multicast cloud name (optional)
- Advertisement IP—Unique advertisement address (optional)
- Port—Port used for file addresses (optional)
- Start IP address— Start of the IP address range, which must be within the range 224.0.0.0 to 239.255.255.255 (optional)
- End IP address—End of the IP address range (optional)
- Primary sender SE—Primary sender SE (optional)
- Default multicast out bandwidth—Maximum multicast rate in kilobits per second (optional)
- Multicast medium—Means of transmitting the multicast (Satellite or Terrestrial). Satellite is default (optional)
- FEC transmission group—Size of the FEC (forward error correction) block in packets. Allowable inputs are 2, 4, 8, 16, 32, 64, and 128. Default is 16. (optional)
- Carousel passes—Maximum number of times a multicast sender sends missing content (optional)
- Carousel delay—Delay, in minutes, between file transmissions (optional)
- Backup sender SE—Backup sender SE (optional)



- Failover grace period—Period of time backup sender goes without getting heartbeat from primary sender before taking over (optional)
- Fallback grace period—Period of time primary sender goes without getting heartbeat from backup sender before taking over (optional)
- PGM router assist—True means IP routers are used to assist in distribution of content. (optional)
- Description—(optional)
- defaultDSDataRate—Default multicast data rate for delivery services in kilo bits per second (optional)
- maxConcurrentSessions—Maximum concurrent session (optional)

**Note**

The primarySenderSe and backupSenderSe needs a clusterId; for example, ClusterConfig\_2221. The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine. The Listing API can be used to get the cluster ID. For more information, see the “getSEs” section on page 3-56 and “getClusters” section on page 3-56.

**Return**

The modified multicast cloud.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=modifyCloud&cloud=<cloud_Id>[&name=<cloud_name>][&advertisementIp=<advertisement_Ip>][&port=<port>][&startIp=<start_Ip>][&endIp=<end_Ip>][&primarySenderSe=<primary_sender_SE_cluster_Id>][&defaultMOutBandwidth=<default_mcast_bandwidth>][&medium=<satellite/terrestrial>][&fecTransmissionGroup=<fec_transmission_group>][&carouselPass=<carousel_pass>][&carouselDelay=<carousel_delay>][&ttl=<ttl>][&backupSenderSe=<backup_sender_SE_cluster_Id>][&failoverGrace=<failover_grace_period>][&fallbackGrace=<fallback_grace_period>][&pgmRouterAssist=<true/false>][&description=<description>][&defaultDSDataRate=<default_DS_data_rate>][&maxConcurrentSessions=<Maximum_concurrent_sessions>]
```

## deleteCloud

Deletes a multicast cloud.

**Parameter**

- Cloud ID—Multicast cloud ID (required)

**Return**

None.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=deleteCloud&cloud=<cloud_ID>
```

## assignReceiverSe

Assigns a receiver SE to multicast cloud.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- SE cluster ID—Cluster ID of the SE (required)

**Note**

The SE cluster ID is the needed to identify the SE. The getSEs action of the Listing API can be used to get the cluster ID. For more information, see the [“getSEs” section on page 3-56](#).

**Return**

None.

**Syntax**

`https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=assignReceiverSe&cloud=<cloud_Id>&SE=<se_cluster_Id>,<se_cluster_Id> ...`

## unassignReceiverSe

Removes a receiver SE from a multicast cloud.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- SE cluster ID—Cluster ID of the SE (required)

**Note**

The SE cluster ID is the needed to identify the SE. The getSEs action of the Listing API can be used to get the cluster ID. For more information, see the [“getSEs” section on page 3-56](#).

**Return**

None.

**Syntax**

`https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=unassignReceiverSe&cloud=<cloud_Id>&SE=<se_cluster_Id>,<se_cluster_Id> ...`

## assignDeliveryService

Assigns a multicast cloud to a delivery service.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- Delivery service ID —Format is Channel\_xxx, where xxx is the ID of the delivery service (required)
- Multicast IP address—Multicast IP address assigned to this delivery service from multicast cloud address range (required)
- carouselPass—Carousel pass value (optional)
- maxDataRate—Maximum multicast data rate for the delivery services in kilo bits per second (required)
- maxConcurrentSessions—Maximum concurrent session (optional)

- fecTransmissionGroup—FEC transmission group value (optional)

**Note**

The assignDeliveryService action expects the deliveryService parameter to be in the form Channel\_xxx, where xxx is the ID of the delivery service.

**Return**

None.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=assignDeliveryService&cloud=<MCastCloud_Id>&Delivery%20Service=<channel_Id>[&mcastIp=<multicast_Ip>][&carouselPass=<carousel_Pass>]
[&maxDataRateMax_data_rate=<Maximum_data_rate_control_for_DS>][&maxConcurrentSessions=<Maximum_concurrent_sessions>][&fecTransmissionGroup=<fec_trans_group>]
```

## unassignDeliveryService

Removes a multicast cloud from a delivery service.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- Delivery service ID —Format is Channel\_xxx, where xxx is the ID of the delivery service (required)

**Note**

The unassignDeliveryService action expects the deliveryService parameter to be in the form Channel\_xxx, where xxx is the ID of the delivery service.

**Return**

None.

**Syntax**

```
https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=unassignDeliveryService&cloud=<cloud_Id>&Delivery%20Service=<channel_Id>
```

## modifyChannelMcast

Modifies a ChannelMcast configuration.

**Parameter**

- Cloud ID—Multicast cloud ID (required)
- Name—Multicast cloud name (required)
- maxDataRate—Maximum multicast data rate for the delivery services in kilo bits per second (required)
- maxConcurrentSessions—Maximum concurrent session (optional)

**Return**

The modified ChannelMcast configuration.

**Syntax**

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=modifyChannelMCast&Delivery%20Service=<Channel\_id>&cloud=<MCastCloud\_id>&maxDataRate=<Maximum\_date\_rate\_control>[&maxConcurrentSessions=<Maximum\_concurrent\_sessions>]

## External System API Actions

The External System API is the ExternalSysApiServlet.

**Syntax**

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ExternalSysApiServlet...

This servlet performs one or more of the following actions:

- [create](#)
- [modify](#)
- [delete](#)

### create

Creates an external system.

**Parameter**

- Name (required)—Name of the External System.
- Description—Description of the External System.
- Register with Prime Central—Indication to register with Prime Central.
- Prime Central IP address (required)—IP address of the Prime Central.
- Prime Central database schema ID (required)—Database schema ID of the Prime Central.
- Prime Central database port (required)—Database Port number of the Prime Central.
- Prime Central database user (required)—Database User name for the Prime Central.
- Prime Central database password (required)—Database Password for the Prime Central.
- Prime Central fault Manager IP address (required)—IP address of the Prime Central Fault Manager.
- Prime Central fault Manager port (required)—Port number used by CDSM to send SNMP traps to the Prime Central.

**Return**

Information of the external system, if the operation is successful.

**Syntax**

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ExternalSysApiServlet?action=create&name=<ExternalSysConfig\_name>[&description=<Description>]&pcRegister=<true|false>&pcIp=<pcIp>&pcDbSid=<pcDbSid>&pcDbPort=<1521>&pcDbUser=<pcDbUser>&pcDbPassword=<pcDbPassword>&pcFmIp=<pcFmIp>&pcFmPort=<1162>

## modify

Modifies external system.

### Parameter

- External system configuration ID (required)
- Name
- Description
- Register with Prime Central
- Prime Central IP address
- Prime Central database schema ID
- Prime Central database port
- Prime Central database user
- Prime Central database password
- Prime Central Fault Manager IP address
- Prime Central Fault Manager port



### Note

The external system configuration ID is returned when the external system creation is successful. Alternatively, you can use the `getExternalSystems` action of the Listing API servlet. For more information, see the [“getExternalSystem” section on page 3-61](#).

### Return

Information of the external system, if the operation is successful.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ExternalSysApiServlet?action=modify&externalSys=<ExternalSysConfig_Id>[&name=<ExternalSysConfig_name>][&pcRegister=<true|false>][&pcIp=<pcIp>][&pcDbSid=<pcDbSid>][&pcDbPort=<1521>][&pcDbUser=<pcDbUser>][&pcDbPassword=<pcDbPassword>][&pcFmIp=<pcFmIp>][&pcFmPort=<1162>]
```

## delete

Deletes an external system.

### Parameter

- External System Configuration ID

### Return

ID of the external system, if the operation is successful.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ExternalSysApiServlet?action=delete&externalSys=<ExternalSysConfig_Id>
```

## Listing APIs

This chapter describes the following listing APIs and the servlet actions they perform:

- [Listing API Actions, page 3-54](#)
- [Device API Actions, page 3-61](#)

## Listing API Actions

The Listing API is the ListingApiServlet. If there is a list inside the object, the listed items are printed as elements of the object.

Some of the output fields are not used for the following actions:

- getSEs
- getDeliveryServices
- getContentOrigins

[Table 3-2](#) lists the unused output fields.

**Table 3-2** Output Fields Not Used in the VDS-IS

Schema Object	Unused Field	Comment
CeConfig	TftpDirectoryListingId	“CeConfig” is mapped to the “Service Engine” schema object.
	WccpConfig	TFTP and WCCP are not used.
	TftpProxyList: <list name="TftpProxyList" type="TftpProxy" size="0"/>	Although “TftpDirectoryListingId,” “TftpProxyList,” and
	WccpRouterListsPerCeForDg : <list name="WccpRouterListsPerCeForDg" type="WccpRouterListPerCeForDg" size="0" />	“WccpRouterListsPerCeForDg” can be queried by API, they are not used in the VDS-IS.
Website	ContentProvidId	“Website” is mapped to the “content origin” schema object.
	CifsWebsites: <list name="ChannelMCasts" type="ChannelMCast" size="0" />	Content Provider and CIFS configurations are not used.  Although “ContentProvidId” and “CifsWebsites” can be queried by API, they are not used in the VDS-IS.
Channel	MCastEnabled	“Channel” is mapped to the “delivery service” schema object.
	ChannelMCasts: <list name="ChannelMCasts" type="ChannelMCast" size="0" />	Content Provider and multicast configurations are not used.  Although “MCastEnabled,” and “ChannelMCasts” can be queried by API, they are not used in the VDS-IS.

**Syntax**

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet...

This servlet performs one or more of the following actions:

- [getContentOrigins](#)
- [getDeliveryServices](#)
- [getSEs](#)
- [getClusters](#)
- [getLocations](#)
- [getDeviceGroups](#)
- [getObjectById](#)
- [getObjectByName](#)
- [getPrograms](#)
- [getPgmMcastAddrInUse](#)
- [getMcastAddrInUse](#)
- [getMCastClouds](#)
- [getStoragePrioClasses](#)
- [getExternalSystem](#)

## getContentOrigins

Lists selected content origin names or lists every content origin.

**Parameter**

Either a list of content origin names or the keyword **all** is required.

**Return**

A list of all content origins specified and their details.

**Syntax**

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getContentOrigins&param=all | <contentOrigin\_name>, <contentOrigin\_name>, ...

## getDeliveryServices

Lists selected delivery service names and related content origin ID or lists all delivery services.

**Parameter**

A list of delivery service names with related content origin IDs, a Service Engine ID, a program ID, or the keyword **all** is required.

**Return**

A list of all delivery services specified and their details.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=
getDeliveryServices&param=all | [name=]<contentOrigin_ID>:(all | <deliveryService_name>) ... |
se=<seConfig_ID> | program=<playlist_ID>
```

**getSEs**

Lists selected Service Engines by Service Engine name, delivery service, or location, or lists all Service Engines. When Service Engines are listed by location, all Service Engines in the given location and all Service Engines (child, grandchild, and so forth) in the subordinate locations are listed.

**Parameter**

A list of Service Engine names, a delivery service ID, a location ID, or the keyword **all** is required.

**Return**

A list of all Service Engines specified and their details.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getSEs&param=
all | [name=]<se_name>, <se_name>, ... | deliveryService=<deliveryService_ID> |
location=<location_ID>
```

**getClusters**

Lists selected cluster names or lists every cluster.

**Parameter**

Either a list of cluster IDs or the keyword **all** is required.

**Note**


---

A cluster is the same thing as a Service Engine.

---

**Return**

A list of all clusters specified and their details.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action= getClusters&
param=all | <Cluster_ID>, <Cluster_ID>, ...
```

**getLocations**

Lists the location of the specified Service Engines or the locations of all Service Engines.

**Parameter**

Either the Service Engine ID or the keyword **all** is required.

**Return**

The requested location record.



**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getLocations&param=all | bySeId=<SE_ID>
```

**getDeviceGroups**

Lists selected device group names or lists all device groups.

**Parameter**

Either a list of device group names or the keyword **all** is required.

**Return**

A list of all device groups specified and their details.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getDeviceGroups&param=all | <device_group_name>, <device_group_name>, ...
```

## getObjectById

Lists an object, based on its string ID.

### Parameter

Object string ID

The following are the object types:

- Service Engine
- Delivery service
- Cluster (cluster is the same thing as Service Engine)
- Device group
- Content origin
- Program
- Channel Device Multicast Config

### Return

The requested object.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action= getObjectById&
param=<SE_ID | DeliveryService_ID | Cluster_ID | DeviceGroup_ID | ContentOrigin_ID | Playlist_ID
|ChannelDeviceMcastConfig_ID>
```



### Note

---

This API is restricted based on permissions granted to the specified user requesting the API. The CDSM allows assignment of API access rights to any user. A user with administrator's privileges bypasses the authentication. For other users, this API is accessed by granting particular rights in the CDSM AAA system.

---

## getObjectByName

Lists an object, based on its name.

### Parameter

Object type:object name

The following are the object types:

- Service Engine
- Delivery service
- Device group
- Content origin
- Program

The delivery service name format is content origin name:delivery service name.

**Return**

The requested object.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=
getObjectByName&param=SE:<seName> | DeliveryService:<deliveryServiceName> | DG:<dgName> |
ContentOrigin:<contentOriginName> | Program:<programName>
```

**Note**

If the type of object is a program, you must have administrator-level access privileges to execute this action, or have user-specific access rights granted for this API.

## getPrograms

Lists all programs specified or all programs and their details.

**Parameter**

A list of program types, program names, delivery service ID, or program ID, or the keyword **all** is required.

**Return**

A list of all programs specified and their details.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getPrograms&para
m=all | type=<wmt | movieStreamer> | name=<program_name>,<program_name>, ... |
deliveryService=<deliveryServiceID> | id=<playlist_ID>,<playlist_ID>...
```

**Note**

You must have administrator-level access privileges to execute this action, or have user-specific access rights granted for this API.

## getPgmMcastAddrInUse

Lists all multicast addresses currently in use by programs.

**Parameter**

None.

**Return**

A list of all the multicast addresses currently in use by programs.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?
action=getPgmMcastAddrInUse
```

**Note**

You must have administrator-level access privileges to execute this action, or have user-specific access rights granted for this API.

## getMcastAddrInUse

Lists all multicast addresses currently in use.

### Parameter

None.

### Return

A list of all the multicast addresses currently in use.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?
action=getMcastAddrInUse
```



### Note

You must have administrator-level access privileges to execute this action, or have user-specific access rights granted for this API.

## getMCastClouds

Lists all the multicast clouds.

### Parameter

mcastCloud\_name

### Return

A list of all the multicast clouds.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getMCastClouds&
param=all | <mcastCloud_name>,<mcastCloud_name>, ....
```

## getStoragePrioClasses

Lists the storage priority classes and their StoragePriorityClass IDs.

### Parameter

Either a list of StoragePriorityClass IDs or the keyword **all** is required.

### Return

A list of all storage priority classes specified and their details.

### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=
getStoragePrioClasses&param=all | <storagePriorityClass_name>, <storagePriorityClass_name>, ...
```

## getExternalSystem

Lists the external system and their ExternalSysConfig IDs.

### Parameter

Either a list of external system IDs or the keyword **all** is required.

### Return

The information of the external systems is returned.

### Syntax

```
https://<cdsmlpAddress>:8443/servlet/com.cisco.unicorn.ui.ListApiServlet?action=getExternalSystem
s&param=all | <external_system_ID>,<external_system_ID>...
```

## Device API Actions

The Device API is the DeviceApiServlet.

### Syntax

```
https://<cdsmlpAddress>:8443/servlet/com.cisco.unicorn.ui.DeviceApiServlet...
```

This servlet performs one or more of the following actions:

- [getDeviceStatus](#)
- [getDevices](#)

## getDeviceStatus

Lists the status of a device by name.

### Parameter

Name of the device that contains the ID of the device or device group.



### Note

---

The name of the device is case sensitive.

---

### Return

A list of devices and their status.

### Syntax

```
https://<cdsmlpAddress>:8443/servlet/com.cisco.unicorn.ui.DeviceApiServlet?action=
getDeviceStatus[&name=<device_ID> | <deviceGroup_ID>]
```

## getDevices

Provides information about the devices in the VDS-IS.

### Parameters

- Type—Type of device (required) is one of the following: **DG** (device group), **SE**, **SR**, **CDSM**, or **all**

- Name—Device name
- ID—Device ID

**Return**

Returns information about all the devices that are the specified device type.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.DeviceApiServlet?action=
getDevices&type=<all|DG|SE|SR|CDSM>[&name=<device_name>][&id=<device_ID>]
```

## Statistics APIs

This chapter describes the Monitoring Statistics API and Streaming Statistics API, and the servlet actions they perform. This chapter contains the following sections:

- [Monitoring Statistics API Actions, page 3-62](#)
- [Streaming Statistics API Actions, page 3-65](#)

## Monitoring Statistics API Actions

This section describes the Monitoring Statistics API and the servlet actions it performs. The Monitoring Statistics API gets monitoring statistics data about a single Service Engine or all the Service Engines in a VDS-IS network. The Monitoring Statistics API is the MonitoringApiServlet.

**Syntax**

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.MonitoringApiServlet...
```

This servlet performs one or more of the following actions:

- [getSeStats](#)
- [getLocationStats](#)
- [getCdnStats](#)

### getSeStats

Obtains monitoring statistics information for the specified Service Engine.

**Parameter**

- Service Engine ID (required)
- Monitoring statistics type (required)

The monitoring statistics types are:

- bytes\_served
- bandwidth\_efficiency\_gain
- streaming\_sessions
- cpu\_utilization

- Time frame (optional)—The time period over which monitoring statistics are obtained. The default is last\_hour.

The options are:

- last\_hour
- last\_day
- last\_week
- last\_month
- custom




---

**Note** If you choose custom, you must specify the time frame using the End time from and End time to options.

---

- End time from (optional)—The date and time that collection of monitoring statistics data should start. You can specify only the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- End time to (optional)—The date and time that collection of monitoring statistics data should end. You can specify only the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- Time zone (optional)—The time zone used to generate the monitoring statistics data. The default is utc.

The time zones are:

- utc—Time in UTC (Coordinated Universal Time)
- se\_local\_time—Time zone specified on the Service Engine
- cdsm\_local\_time—Time zone specified on the CDSM

#### Return

Requested monitoring statistics data for the selected time period for the Service Engine.

#### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MonitoringApiServlet?action=getSEstats
&id=<SE_id>&type=<bytes_served | bandwidth_efficiency_gain | streaming_sessions |
cpu_utilization>[&time_frame=<last_hour | last_day | last_week | last_month | custom>]
[&end_time_from=<mm/dd/yyyy [hh:mm[:ss]]>][&end_time_to=<mm/dd/yyyy [hh:mm[:ss]]>]
[&time_zone=<utc | se_local_time | cdsm_local_time>]
```

## getLocationStats

Obtains monitoring statistics information for all the Service Engines in the specified location.

#### Parameter

- Location ID (required)
- Monitoring statistics type (required)

The monitoring statistics types are:

- bytes\_served

- bandwidth\_efficiency\_gain
- streaming\_sessions
- Time frame (optional)—The time period over which monitoring statistics are obtained. The default is last\_hour.

The options are:

- last\_hour
- last\_day
- last\_week
- last\_month
- custom




---

**Note** If you choose custom, you must specify the time frame using the End time from and End time to options.

---

- End time from (optional)—The date and time that collection of monitoring statistics data should start. You can specify only the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- End time to (optional)—The date and time that collection of monitoring statistics data should end. You can specify only the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- Time zone (optional)—The time zone used to generate the monitoring statistics data. The default is utc.

The time zones are:

- utc—Time in UTC (Coordinated Universal Time)
- cdsm\_local\_time—Time zone specified on the CDSM

#### Return

Requested monitoring statistics data for the selected time period for the Service Engines in the specified location.

#### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MonitoringApiServlet?action=
getLocationStats&id=<Location_ID>&type=<bytes_served | bandwidth_efficiency_gain |
streaming_sessions>[&time_frame=<last_hour | last_day | last_week | last_month | custom>]
[&end_time_from=<mm/dd/yyyy [hh:mm[:ss]]>][&end_time_to=<mm/dd/yyyy [hh:mm[:ss]]>]
[&time_zone=<utc | cdsm_local_time>]
```

## getCdnStats

Obtains monitoring statistics information for the entire VDS-IS network.

#### Parameter

- Monitoring statistics type (required)

The monitoring statistics types are:



- bytes\_served
- bandwidth\_efficiency\_gain
- streaming\_sessions
- Time frame (optional)—The time period over which monitoring statistics are obtained. The default is last\_hour.

The options are:

- last\_hour
- last\_day
- last\_week
- last\_month
- custom




---

**Note** If you choose custom, you must specify the time frame using the End time from and End time to options.

---

- End time from (optional)—The date and time that collection of monitoring statistics data should start. You can specify the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- End time to (optional)—The date and time that collection of monitoring statistics data should end. You can specify the date or the date and time. The date format is mm/dd/yyyy, and the time format is hh:mm. Optionally, you can specify the time in hh:mm:ss.
- Time zone (optional)—The time zone used to generate the monitoring statistics data. The default is utc.

The time zones are:

- utc—Time in UTC (Coordinated Universal Time)
- cdsm\_local\_time—Time zone specified on the CDSM

#### Return

Requested monitoring statistics data for the selected time period for all the Service Engines in the VDS-IS network.

#### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MonitoringApiServlet?action=getCdnStats&type=<bytes_served | bandwidth_efficiency_gain | streaming_sessions>[&time_frame=<last_hour | last_day | last_week | last_month | custom>][&end_time_from=<mm/dd/yyyy [hh:mm[:ss]]>][&end_time_to=<mm/dd/yyyy [hh:mm[:ss]]>][&time_zone=<utc | cdsm_local_time>]
```

## Streaming Statistics API Actions

This section describes the Streaming Statistics API and the servlet actions it performs. The streaming statistics are collected from the VDS-IS network Service Engines and device groups and sent to the CDSM. The HTTP, Movie Streamer, and WMT streaming statistical data is monitored and displayed in the CDSM for all Service Engines, all device groups, or all the Service Engines within a selected device group.

The Streaming Statistics API is the `SprayerApiServlet`. The CDSM must be running for the servlet to function.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SprayerApiServlet...`

This servlet performs one or more of the following actions to collect statistics for each Service Engine (SE), device group (DG), or device group name for all the Service Engines in the specified device group:

- [getHttp](#)
- [getMovieStreamer](#)
- [getWmt](#)

## getHttp

Collects HTTP streaming statistics data from the Service Engines and device groups and sends it to the CDSM.

**Parameter**

The SE keyword, DG keyword, or the name of the device group is required.

**Return**

Requested HTTP streaming statistics data for the Service Engines or device groups.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SprayerApiServlet?action=getHttp&param=SE | DG | <DG_name>`

## getMovieStreamer

Collects Movie Streamer streaming statistics data from the Service Engines and device groups and sends it to the CDSM.

**Parameter**

The SE keyword, DG keyword, or the name of the device group is required.

**Return**

Requested Movie Streamer streaming statistics data for the Service Engines or device groups.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SprayerApiServlet?action=getMovieStreamer&param=SE | DG | <DG_name>`

## getWmt

Collects WMT streaming statistics data from the Service Engines and device groups and sends it to the CDSM.

**Parameter**

The SE keyword, DG keyword, or the name of the device group is required.

**Return**

Requested WMT streaming statistics data for the Service Engines or device groups.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.SprayerApiServlet?action=getWmt&
param=SE | DG | <DG_name>
```

**XML-Formatted Output for Streaming Statistics**

The following is the Document Type Definition (DTD) of the XML-formatted output for streaming statistics:

```
<?xml version="1.0" ?>
DOCTYPE <!DOCTYPE sprayerStats [

<!ELEMENT sprayerStats (message, (HttpStats* | MovieStreamerStats* | WmtStats* |
FmsStats*) )>
<!ATTLIST sprayerStats
    action (getHttp | getMovieStreamer | getWmt | getFms )#REQUIRED
    count CDATA #REQUIRED>

<!ELEMENT message EMPTY>
<!ATTLIST message
    status (success | fail) success
    message CDATA #IMPLIED>

<!ELEMENT HttpStats EMPTY>
<!ATTLIST HttpStats
    name CDATA #REQUIRED
    requestsPerSec CDATA #REQUIRED
    bytesPerSec CDATA #REQUIRED
    hitRate CDATA #REQUIRED>

<!ELEMENT MovieStreamerStats EMPTY>
<!ATTLIST MovieStreamerStats
    name CDATA #REQUIRED
    totalBytes CDATA #REQUIRED
    totalPackets CDATA #REQUIRED
    rtspConnections CDATA #REQUIRED
    allConnections CDATA #REQUIRED>

<!ELEMENT WmtStats EMPTY>
<!ATTLIST WmtStats
    name CDATA #REQUIRED
    requestsPerSec CDATA #REQUIRED
    bytesPerSec CDATA #REQUIRED
    hitRate CDATA #REQUIRED>

<!ELEMENT FmsStats EMPTY>
<!ATTLIST FmtStats
    name CDATA #REQUIRED
    allConnections CDATA #REQUIRED
    bytesPerSec CDATA #REQUIRED
    hitRate CDATA #REQUIRED>

]>
```

# File Management APIs

This chapter describes the following file management APIs and the servlet actions they perform:

- [File Management API Actions, page 3-68](#)
- [Certificate and Key File Management API, page 3-77](#)

## Using Multipart/Form-Data Request to Upload a File

There are two import methods for the FileMgmtApiServlet actions and the CertKeyFileMgmtApiServlet actions:

- Import-imports a file from an external HTTP, HTTPS, or FTP server
- Upload-uploads a file from any location that is accessible from your PC

For the “upload” import method, a multipart/form-data request is used. Following is an example of the upload import method for the registerFile action of the FileMgmtApiServlet that uses the curl utility to upload a file for the HTTPS root CA:

```
curl -k -u admin:default -F "rootfile=@rootc.crt"
https://10.74.61.199:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=registerFile&importMethod=upload&fileType=26&destName=rootc.pem
```

In this example, the curl utility uploads the file and the URL sets the parameters; specifically the destination filename is rootc.pem.

If the curl utility is used, another way to upload the file is to use the option -F "file=!sourceFile.xml" can upload the original file sourceFile.xml as a multipart/form-data request.

The following actions of the FileMgmtApiServlet use a multipart/form-data request for the importMethod=upload:

- registerFile
- validateFile
- modifyFile

The refetchFile only uses importMethod=import.

The following actions of the CertKeyFileMgmtApiServlet use multipart/form-data request:

- registerFile
- modifyFile

## File Management API Actions

File Management API actions are used to manage external XML files registered with the CDSM. These external files include Coverage Zone files, Network Attached Storage (NAS) files, Service Rule files, and CDN Selector files.



### Note

---

NAS is only supported in lab integrations as proof of concept.

---

Coverage Zone files are registered with the CDSM and associated with a specific Service Router (SR) or applied globally to the CDN network using the File Management API.

NAS files are registered with the CDSM using the File Management API. The following Delivery Service Provisioning API actions are used to associate a NAS file with a Content Origin, which, through delivery services, makes the NAS file settings available to all root devices located in the same tier as the Content Acquirer:

- [createContentOrigin](#), page 3-24
- [modifyContentOrigin](#), page 3-25

Service Rules files are registered with the CDSM using the File Management API. The Delivery Service Provisioning API action, [applyRuleFile](#), page 3-16, is used to apply a Service Rule file to devices associated with a delivery service.

CDN Selector files are registered with the CDSM and applied to an SR using the File Management API.

The File Management API uses the FileMgmtApiServlet. Some of the output fields are not used for the following actions:

- registerFile
- modifyFile
- listFile

[Table 6-1](#) lists the unused output fields.

**Table 3-3** Output Fields Not Used in the VDS-IS

Schema Object	Unused Field	Comment
Record	PacInfos: <list name="PacInfos" type="PacInfo" size="0"/>	Although PacInfos and DsvcLocations can be queried by API, they are not used in the VDS-IS.
	DsvcLocations: <list name="DsvcLocations" type="DsvcLocation" size="0"/>	

### Syntax

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet...`

The servlet performs one or more of the following actions:

- [listTypes](#)
- [registerFile](#)
- [validateFile](#)
- [refetchFile](#)
- [modifyFile](#)
- [deleteFile](#)
- [listFile](#)
- [applyCZ](#)
- [applyCdnSelector](#)

## listTypes

List all of the file types supported by the File Management API.

**Parameter**

None.

**Return**

The list of file types supported by the File Management API.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=listTypes`

**registerFile**

Registers a file with the CDSM using either the import or upload method. The import method allows you to import a supported file from an external HTTP, HTTPS, FTP, or CIFS server. The upload method allows you to upload a supported file from any location that is accessible from your PC.

**Parameter**

- File type (required)

The settings are:

- 1—Coverage Zone file
- 17—Geo/IP file
- 19—CDN Selector file
- 20—Rule file
- 22—NAS file
- 26—HTTPS Root Certificate file (Used by SEs to validate the Origin server certificates, one or more root certificates can be uploaded to the CDSM.)

- Destination file name (required)

- Import method (required)

The settings are:

- Import
- Upload

If the import method is set to import, the following parameters also apply:

- URL of the origin file (required)—For example, `//myserver/folder/myfile.txt` or `<protocol>://myhost/myfile.txt`. The protocols supported are:
  - `http://`
  - `https://`
  - `ftp://`
- Time-to-live (TTL) (optional)—Frequency, in minutes, with which the CDSM looks for changes in the source file. The default is 10. The range is from 1 to 1440.
- NT LAN Manager (NTLM) user domain name (optional)
- Username (optional)
- Password (optional)
- Disable Basic Authentication (optional)—The default is false.

When set to true, NTLM headers cannot be stripped off to allow fallback to the basic authentication method.

#### Rule

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the [“Using Multipart/Form-Data Request to Upload a File” section on page 3-68](#).

#### Return

A confirmation that the file has been registered.



#### Note

In the XML file returned, an internal reference is assigned to the file in the format of FileInfo\_XXX, where XXX is the ID of the file.

#### Syntax

To register a file using the import method, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=registerFile
&fileType=<1 | 17 | 19 | 20 | 22 | 26>&destName=<destination_filename>&importMethod=import&
originUrl=<file_url> [&ttl=<update_interval>][&username=<username>][&password=<password>]
[&domain=<ntlm_domain>][&disableBasicAuth=<>false | true>]
```

To register a file using the upload method, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=registerFile
&fileType=<1 | 17 | 19 | 20 | 22 | 26>&destName=<destination_filename>&importMethod=upload
```

## validateFile

Validates a registered file, or uploads or imports a file into the CDSM and validates the file.

#### Parameter

- File type (required)

The settings are:

- 1—Coverage Zone file
- 17—Geo/IP file
- 19—CDN Selector file
- 20—Rule file
- 22—NAS file

- File ID (required if the destination filename is not specified)—The format is FileInfo\_XXX, where XXX is the ID of the file.

- Import method (required if ID is not specified)

The settings are:

- Import
- Upload

- Destination filename (required if ID is not specified)

If the import method is set to import, the following parameters also apply:

- URL of the origin file (required)—For example, //myserver/folder/myfile.txt or *<protocol>://myhost/myfile.txt*. The protocols supported are:
  - http://
  - https://
  - ftp://
- TTL (optional)—Frequency, in minutes, with which the CDSM looks for changes in the source file. The default is 10. The range is from 1 to 1440.
- NTLM user domain name (optional)
- Username (optional)
- Password (optional)
- Disable Basic Authentication (optional)—The default is false.
 

When set to true, NTLM headers cannot be stripped off to allow fallback to the basic authentication method.

#### Rules

- If the file ID is specified, all optional parameters are ignored.
- If the file ID is not specified, the destination filename and import method must be specified.

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the [“Using Multipart/Form-Data Request to Upload a File” section on page 3-68](#).

#### Return

A confirmation that the file is valid.

#### Syntax

To validate a registered file, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=validateFile
&fileType=<1 | 17 | 19 | 20 | 22>&id=<FileInfo_id>
```

To import a file from an external HTTP, HTTPS, FTP, or CIFS server into the CDSM and validate it, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=validateFile
&fileType=<1 | 17 | 19 | 20 | 22>&destName=<destination_filename>&importMethod=import
&originUrl=<file_url> [&tTl=<update_interval>][&username=<username>][&password=<password>]
[&domain=<ntlm_domain>][&disableBasicAuth=<false | true>]
```

To upload a file from a location accessible from your PC and validate it, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=validateFile
&fileType=<1 | 17 | 19 | 20 | 22>&destName=<destination_filename>&importMethod=upload
```



#### Note

Regardless of whether you use the upload or import method to validate the file, the validate action does not register the file with the CDSM.



## refetchFile

Notifies the CDSM to refetch a registered file immediately.

**Note**

The refetchFile action applies to files registered using the import method only.

**Parameter**

- File type (required)  
The settings are:
  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (required)—The format is FileInfo\_xxx, where xxx is the ID of the file.

**Return**

A confirmation that the file will be refetched shortly.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=refetchFile&fileType=<1 | 17 | 19 | 20 | 22 | 26>&id=<FileInfo_id>
```

## modifyFile

Modifies the metadata of a file or modifies either the credentials or TTL settings that control the updates to the file.

**Parameter**

- File type (required)  
The settings are:
  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (required)—The format is FileInfo\_xxx, where xxx is the ID of the file.
- Import method (required if ID is not specified)  
The settings are:
  - Import

- Upload
- Destination filename

If the import method is set to import, the following parameters also apply:

- URL of the origin file (required)—For example, //myserver/folder/myfile.txt or <protocol>://myhost/myfile.txt. The protocols supported are:
  - http://
  - https://
  - ftp://
- TTL (optional)—Frequency, in minutes, with which the CDSM looks for changes in the source file. The default is 10. The range is from 1 to 1440.
- NT LAN Manager (NTLM) user domain name (optional)
- Username (optional)
- Password (optional)
- Disable Basic Authentication (optional)—The default is false.

When set to true, NTLM headers cannot be stripped off to allow fallback to the basic authentication method.

#### Rules

- If the file ID is specified, all optional parameters are ignored.
- If the file ID is not specified, the destination filename and import method must be specified.

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the [“Using Multipart/Form-Data Request to Upload a File” section on page 3-68](#).

#### Return

A confirmation that the file was successfully modified.

#### Syntax

To modify the settings of a file that was imported into the CDSM from an external server, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?
action=modifyFile&fileType=<1 | 17 | 19 | 20 | 22 | 26>&id=<FileInfo_id>&destName=
<destination_filename>&importMethod= import&originUrl=<file_url>[&ttl=<update_interval>]
[&username=<username>][&password=<password>][&domain=<ntlm_domain>][&disableBasicAuth
=<false | true>]
```

To modify the settings of a file that was uploaded from a location accessible from your PC into the CDSM, use the following syntax:

```
https://<cdsMIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?
action=modifyFile&fileType=<1 | 17 | 19 | 20 | 22 | 26>&id=<FileInfo_id>&destName=
<destination_filename>&importMethod=upload
```

## deleteFile

Removes a registered file from the CDSM.

**Parameter**

- File type (required)  
The settings are:
  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (required)—The format is FileInfo\_xxx, where xxx is the ID of the file.

**Rule**

You can only delete files that are not currently in use.

**Return**

A confirmation that the file has been deleted.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=deleteFile&fileType=<1 | 17 | 19 | 20 | 22 | 26>&id=<FileInfo_id>
```

## listFile

Displays the details of a specified registered file or displays the details of all registered files of a specified file type.

**Parameter**

- File type (required)  
The settings are:
  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (optional)—The format is FileInfo\_xxx, where xxx is the ID of the file.

**Return**

If a file ID was specified, the details of the requested file are listed. If no file ID was provided, the message lists all files of the specified type and their details. If no files of the specified type exist in the CDSM, the message returned indicates that the request was successful but warns that no files of the specified type exist.

**Syntax**

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=listFile&fileType=<1 | 17 | 19 | 20 | 22 | 26>[&id=<FileInfo_id>]`

## applyCZ

Applies a Coverage Zone file to an SR, removes a Coverage Zone file from an SR, or configures global routing.

### Parameter

- Target—Deploy a Coverage Zone file globally to the entire VDS-IS network or deploy a Coverage Zone file on the specified SR only. Valid values are:
  - Global
  - SR ID —The format is CrConfig\_XXX, where XXX is the ID of the active SR.
- Coverage Zone file ID. Valid values are:
  - None—Removes the association of the Coverage Zone file with the target.
  - File ID (required)—The format is FileInfo\_XXX, where XXX is the ID of the file.

If the target is set to global, the following parameter applies:

- DNS TTL (optional)—Time period (in seconds) for caching DNS replies. The default is 60 seconds. The range is from 0 to 60.

### Return

Confirmation that the Coverage Zone file has been applied to the SR or that global routing has been set.

### Syntax

To set global routing for a Coverage Zone file, use the following syntax:

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=applyCZ&target=global&czId=<FileInfo_id>[&dnsTtl=<dns_ttl>]`

To reset global routing for a Coverage Zone file, use the following syntax:

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=applyCZ&target=global&czId=none`

To apply a Coverage Zone file to an SR, use the following syntax:

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=applyCZ&target=<CrConfig_id>&czId=<FileInfo_id>`

To remove a Coverage Zone file from an SR, use the following syntax:

`https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=applyCZ&target=<CrConfig_id>&czId=none`

## applyCdnSelector

Assigns a CDN Selector file to an SR or unassigns a CDN Selector file from an SR.

### Parameter

- SR ID— The format is CrConfig\_XXX, where XXX is the ID of the SR.
- CDN Selector file. Valid values are:

- None—Unassigns the CDN Selector file from the SR.
- File ID (required)—The format is FileInfo\_xxx, where xxx is the ID of the file.

**Return**

Confirmation that the CDN Selector file has been assigned to the SR or unassigned from the SR.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.FileMgmtApiServlet?action=applyCdnSelector&SR=<CrConfig_id>&cdnSelector=<none | FileInfo_id>
```

## Certificate and Key File Management API

The Certificate and Key File Management API is used to upload the certificate and key files for HTTPS Streaming to the CDSM, where they are distributed to all SEs. Uploading new certificate and key files overwrites the existing files.

The Certificate and Key File Management API uses the CertKeyFileMgmtApiServlet.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CertKeyFileMgmtApiServlet...
```

This servlet performs one or more of the following actions:

- [registerFile](#)
- [modifyFile](#)
- [deleteFile](#)
- [listFile](#)

### registerFile

Uploads the certificate and key files.

**Parameters**

- certDestName—Destination filename of the certificate file
- keyDestName—Destination filename of the key file

Both the certDestName and keyDestName are actually not parameters, but instead refer to files that are uploaded by posting a multipart/form data request.



**Note** The path and filename of the certificate file and the path and filename of the key file must point to the actual files; otherwise, an error stating inconsistent information is reported.

**Rule**

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the [“Using Multipart/Form-Data Request to Upload a File” section on page 3-68](#).

**Return**

If the registerFile action is successful, a confirmation that the files have been uploaded is returned. If the registerFile action fails, a warning or error code is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CertKeyFileMgmtApiServlet?
action=registerFile&certDestName=<certificate_file_name>&keyDestName=<keyfile_name>
```

## modifyFile

Uploads and overwrites the certificate and key files.

**Parameters**

- certDestName—Destination filename of the certificate file
- keyDestName—Destination filename of the key file

Both the certDestName and keyDestName are actually not parameters, but instead refer to files that are uploaded by posting a multipart/form data request.




---

**Note** The path and filename of the certificate file and the path and filename of the key file must point to the actual files; otherwise, an error stating inconsistent information is reported.

---

**Rule**

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the [“Using Multipart/Form-Data Request to Upload a File” section on page 3-68](#).

**Return**

If the modifyFile action is successful, a confirmation that the files have been uploaded is returned. If the modifyFile action fails, a warning or error code is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CertKeyFileMgmtApiServlet?
action=modifyFile&certDestName=<certificate_file_name>&keyDestName=<keyfile_name>
```

## deleteFile

Deletes the certificate and key files from the VDS-IS. Only use this action if you want to disable the HTTPS feature on all delivery services.

**Parameters**

None.

**Return**

If the deleteFile action is successful, a confirmation that the files have been deleted is returned. If the deleteFile action fails, a warning or error code is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CertKeyFileMgmtApiServlet?  
action=deleteFile
```

**listFile**

Provides information about uploaded certificate file and the key file.

**Parameters**

None.

**Return**

If the listFile action is successful, information about the files is returned.

**Syntax**

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.CertKeyFileMgmtApiServlet?  
action=listFile
```

