



CHAPTER 3

Provisioning APIs

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

- [Delivery Service Provisioning API Actions, page 3-1](#)
- [Location Provisioning API Actions, page 3-15](#)
- [Service Engine Provisioning API Actions, page 3-17](#)
- [Program API Actions, page 3-18](#)
- [Media API Actions for Programs, page 3-23](#)
- [URL Management API Actions, page 3-25](#)

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

[Table 3-1](#) lists the unused output fields.

Table 3-1 Output Fields Not Used in the CDS

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 CDS.

Table 3-1 Output Fields Not Used in the CDS (continued)

Schema Object	Unused Field	Comment
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 CDS.
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 CDS.

Syntax

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

This servlet performs one or more of the following actions:

- [createDeliveryService](#)
- [createDeliveryServiceGenSettings](#)
- [addManifest](#)
- [assignSEs](#)
- [assignDeliveryServiceIp](#)
- [fetchNow](#)
- [modifyDeliveryService](#)
- [modifyDeliveryServiceGenSettings](#)
- [modifyManifest](#)
- [unassignSEs](#)
- [unassignDeliveryServiceIp](#)
- [deleteDeliveryServices](#)
- [deleteDeliveryServiceGenSettings](#)
- [createContentOrigin](#)
- [modifyContentOrigin](#)
- [deleteContentOrigin](#)
- [applyRuleFile](#)
- [applyGeoIpFile](#)

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 default is medium.

The settings are:

- High
- Medium
- Low
- Delivery service quota (optional)—Only valid for non-live delivery services
- Delivery service description (optional)—The default is null.
- FailoverIntvl (optional)—The default is 120.
- Never (optional)—The default is false.
- Delivery service QoS (optional)
- Live (optional)—The default is false.

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>][&quota=<quota>][&failoverIntvl=<20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120>][&never=<true | false>][&qos=<system|0-63>][&deliveryQos=<0-63>][&desc=<description>][&live=<true | false>]
```

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

**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=create
DeliveryServiceGenSettings&deliveryService=<deliveryService_ID>&Bitrate=<Maximum bitrate
limit per session for HTTP(Kbps)(0-2000000)>&OsProtocol=<0|1 0-HTTP 1-HTTPS>&Streaming
Protocol=<0|1 0-HTTP 1-HTTPS>&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 Read Timeout>[&HttpAllow=
<truelfalse> Enable 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][&Cache
Error=<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>]
```

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=<ttl>
[&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>]
```

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://<cismIpAddress>: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://<cismIpAddress>: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 default is medium.

The settings are:

- High
- Medium
- Low
- delivery service quota (optional)—Only valid for non-live delivery services
- FailoverIntvl (optional)
- Never (optional)
- Delivery service QoS (optional)
- Description (optional)


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>][&quota=<quota>][&failoverIntvl=<20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120>][&never=<true | false>][&deliveryQos=<0-63>][&desc=<description>]
```

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

**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=modify
DeliveryServiceGenSettings&deliveryService=<deliveryService_ID>[&Bitrate=<Maximum bitrate
limit per session for HTTP(Kbps)(0-2000000)>][&OsProtocol=<0|1 0-HTTP 1-HTTPS >]
[&StreamingProtocol=<0|1| 0-HTTP 1-HTTPS >][&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 Read
Timeout>][&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>]
```

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>][&ttl=<ttl>][&
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=unassignDeliveryServiceIp&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>`

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
- 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>]
```

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
- 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>]
```

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 API 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.

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

This servlet performs one or more of the following actions:

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

activateSe

Activates a specified Service Engine.

Parameter

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

Return

The modified Service Engine object.

Syntax

```
https://<cismIpAddress>: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)>`

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 CDS 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 4-6.

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 2-4) 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 medial 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 4-6](#).

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());
HttpsURLConnection.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);

HttpsURLConnection 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.

