



# Provisioning APIs

The following sections contain information to help you provision your APIs:

- [Delivery Service Provisioning API Actions, page 3-1](#)
- [Location Provisioning API Actions, page 3-11](#)
- [Service Engine Provisioning API Actions, page 3-12](#)
- [Program API Actions, page 3-13](#)
- [URL Management API Actions, page 3-20](#)
- [Multicast Cloud API Actions, page 3-24](#)

## 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 ECDS**

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 ECDS.

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

Schema Object	Unused Field	Comment
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 ECDS.
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 ECDS.

**Syntax**

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

This servlet performs one or more of the following actions:

- [createDeliveryService](#)
- [addManifest](#)
- [assignSEs](#)
- [assignDeliveryServiceIP](#)
- [fetchNow](#)
- [modifyDeliveryService](#)
- [modifyManifest](#)
- [unassignSEs](#)
- [unassignDeliveryServiceIp](#)
- [deleteDeliveryServices](#)
- [createContentOrigin](#)
- [modifyContentOrigin](#)
- [deleteContentOrigin](#)

## 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 description (optional)—The default is null.
- FailoverIntvl (optional)—The default is 120.
- Never (optional)—The default is false.
- Live (optional)—The default is false.

#### Return

The newly created delivery service ID.

#### Syntax

```
https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createDelivery
Service&deliveryService=<deliveryService_name>&contentOrigin=<contentorigin_ID>[&weakCert=
<true | false>][&skipEncrypt= <true | false>][&priority=<high | medium | low>]
[&failoverIntvl=<20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120>][&never=<true | false>]
[&qos=<system|0-63>][&desc=<description>][&live=<true | false>]
```

## addManifest

Adds a Manifest file to a specified delivery service.

#### Parameter

- Delivery service ID (required)
- Manifest URL (required)
- Quota (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>&quota=<quota>&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\_ID> entry in the form **CeConfig\_####**.
- Either a list of clusters 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.

**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>, ...][&cluster=all | <Cluster_ID>, <Cluster_ID>, ...]
```

## assignDeliveryServiceIP

Assigns an IP address 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)—<SE\_ID> entry in the form **CeConfig\_####**

**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=assignDeliver
yServiceIp&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)
- Weak certification (optional)
- Skip encryption (optional)
- Delivery service priority (optional)—The default is medium.

The settings are:

- High
- Medium
- Low
- FailoverIntvl (optional)
- Never (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>][&weakCert=<true | false>][&skipEncrypt=<true | false>][&priority=<high | medium | low>][&failoverIntvl=<20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120>][&never=<true | false>][&desc=<description>]`

## modifyManifest

Modifies Manifest file settings.

**Parameter**

- Delivery service ID (required)
- Manifest URL (optional)
- Quota (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>][&quota=
<quota>][&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). <SE\_ID> entry in the form **CeConfig\_####**.
- Either a list of clusters 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)—<SE\_ID> entry in the form **CeConfig\_####**

**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>, ...`

## createContentOrigin

Creates a content origin.

**Parameter**

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



---

**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`.

---

- 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>[&fqdn=<fqdn>]
[&wmtAuth=<basic | ntlm | digest | negotiate>][&description=<description>]
```

## modifyContentOrigin

Modifies content origin settings.

**Parameter**

- Content origin ID (required)
- Content origin name (required)
- Origin server (optional)
- FQDN (optional)
- 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_server>][&fqdn=<fqdn>][&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>, ...`

## 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](#)

## activateSe

Activates a specified Service Engine.

**Parameter**

- Service Engine ID (required)—<SE\_ID> entry in the form **CeConfig\_####**
- 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)—<SE\_ID> entry in the form **CeConfig\_####**

**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>
```

## 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](#)
- [startSession](#)
- [pauseSession](#)
- [stopSession](#)

## 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 ECDS 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>]
```



#### Note

To enable alternate URL in the Live Program with createProgram API, the ucastInfo reference URL in program XML file must be given with different LiveProgram events. For more information on Program file example refer [WMT Alternate URL Live Event](#).



#### Note

Refer [Appendix A, “Program Files in the ECDS Software”](#) for information on Program file example [WMT Unicast-Multicast Live Event](#).

## 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>
```



### Note

Unicast URL reference and multicast URL reference will appear in Live Program Event only after assigning the delivery service.

**Note**

Although there is a convert option in the GUI that converts API interface to CDSM interface, programs that are managed by CDSM interface cannot be modified by an API interface. Once a program is converted into CDSM interface, there is no option to convert it back to an API interface.

## 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. <SE\_ID> entry in the form **CeConfig\_####**.

**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. Modify program file also supports Alternate URL of WMT unicast failover.

**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>]
```

**Note**

For configuring AlternateUnicastUrl using this ModifyProgramFile API, refer sample program file [WMT Alternate URL Live Event](#) in Appendix A “Program Files in the ECDS Software”.

## 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.

For sample program file to configure AlternateUnicastUrl, Refer Appendix A “Program Files in the ECDS Software” for information on program file examples "[Movie Streamer Multicast Event](#)"

**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. <SE\_ID> entry in the form **CeConfig\_####**.

**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>
```

## startSession

Makes an HTTP GET request when the client selects Play or when the content starts to stream.

### Parameters

- Subscriber or client IP address
- Subscriber or client destination port for content stream
- Service Engine IP address serving stream
- Service Engine port
- Request identifier returned from Camiant CDN Application Manager Authorization module during authorization

### Return

200 Ok—Request was successful.

400 Bad request—Request parameters were incomplete or invalid. QoS request was not successful.

500 Internal server error—Internal error occurred. QoS request was not successful.

### Syntax

```
/cdnam/startSession.do?subip=x.x.x.x&subport=n&ceip=y.y.y.y&ceport=m&rid=r
```

## pauseSession

Makes an HTTP GET request when the client selects Pause.

### Parameter

Request identifier returned from Camiant CDN Application Manager Authorization module during authorization.

### Return

200 Ok—Request was successful.

400 Bad request—Request parameter was unknown. No QoS action occurred.

500 Internal server error—Internal error occurred. QoS request was not successful.

### Syntax

```
/cdnam/pauseSession.do?rid=r
```

## stopSession

Makes an HTTP GET request when the client selects Stop, or when the content stops streaming.

### Parameter

Request identifier returned from Camiant CDN Application Manager Authorization module during authorization

**Return**

200 Ok—Request was successful.

400 Bad request—Request parameter was unknown. No QoS action occurred.

500 Internal server error—Internal error occurred. QoS request was not successful.

**Syntax**

/cdnam/stopSession.do?rid=r

## 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.

**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 remote execution agent (REA) agent is running. The CLI **show rea info** command can be used for viewing the status of the REA and **rea start** for starting REA.

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.

**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

```

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/ecds/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>
```

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>
```

## 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:

- [create Cloud](#)
- [modify Cloud](#)
- [assign Receiver Se](#)
- [create Multicast Delivery Service](#)
- [Assign Multicast Cloud to Delivery Service](#)
- [Unassign Multicast Cloud from Delivery Service](#)
- [Unassign Receiver SE](#)
- [delete Cloud](#)

## create Cloud

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)

**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, page 4-4](#) and [getClusters, page 4-6](#).

**Return**

Success message with Multicast ID of the newly created 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_mcast_bandwidth>[&medium=<satellitelerrestrial>][&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>]
```

## modify Cloud

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)

**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, page 4-4](#) and [getClusters, page 4-6](#).

**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=<satelliteterrestrial>][&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>]
```

## assign Receiver Se

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 4-4](#).

**Return**

None.

**Syntax**

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

## create Multicast Delivery Service

Creates a multicast 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.
- System Qos (optional)
- Delivery service description (optional)—The default is null.
- FailoverIntvl (optional)—The default is 120.
- Never (optional)—The default is false.
- 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 [from release 2.6](#).

---

### 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=<truefalse>][&skipEncrypt=<truefalse>][&priority=<high|medium|low>][&mcastEnable=<unicast_only|multicast_only|unicast_multicast>][&live=<truefalse>][&qos=<system|0-63>][&desc=<description>][&failoverIntvl=<failoverIntvl, <20|30|40|50|60|70|80|90|100|110|120>][&never=<truefalse>]
```

## Assign Multicast Cloud to Delivery Service

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)



### 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=<cloud_Id>&Delivery Service=<channel_Id>&mcastIp=<multicast_Ip>
```

## Unassign Multicast Cloud from Delivery Service

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 Service=<channel_Id>
```

## Unassign Receiver SE

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, page 4-4](#).

### Return

None.

### Syntax

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

## delete Cloud

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>
```