Cisco VDS Internet Streamer 3.3 API Guide

June 8, 2016

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number:

Cisco VDS Internet Streamer 3.3 API Guide

Preface xi

Document Revision History xi

Audience xi

Document Organization xii

Document Conventions xii

Related Publications xiii

Obtaining Documentation and Submitting a Service Request xiii

CHAPTER 1 **Introduction to Cisco CDS Software APIs** 1-1

CDSM RESTful APIs 1-1

Authentication and Authorization 1-2

ID Format 1-2

Calling the RESTful APIs 1-2

Interactive Calls 1-3

Programmed Calls 1-3

API Status Codes 1-3

Error Status Codes 1-3

Success Status Codes 1-4

CDSM Legacy API 1-4

Calling the Legacy APIs 1-6

Interactive Calls 1-6

Programmed Calls 1-7

Sample Java Program 1-7

API Error Messages 1-9

API Tasks 1-11

Replication Status API 1-11

Provisioning APIs 1-11

Listing API 1-13

Statistics API 1-13

File Management API 1-14

Request Routing Engine APIs 1-15

Request Routing Engine API 1-15

Last-Resort URL Translator Web Services API 1-15

Proximity Engine SOAP API 1-15

Text Part Number:

CHAPTER 2

CDSM RESTful APIs 2-1

- Device APIs 2-1
  - Device Groups 2-1
  - Device Group Assignments 2-3
  - Locations 2-4
  - Replication 2-5
    - Default Bandwidth 2-5
    - Distribution 2-6
    - Multicast Distribution 2-6
  - Service Control 2-7
    - Enable Rules 2-7
    - Authorization Service 2-8
    - Transaction Logging 2-9
  - Application Control 2-13
    - Default and Maximum Bandwidth 2-13
    - Windows Media Streaming General Settings 2-14
    - Windows Media Streaming Bypass List 2-15
    - Movie Streamer General Settings 2-17
    - RTSP Advanced Settings 2-18
    - FMS Admin Allow Hosts 2-19
    - FMS General Settings 2-19
    - FMS Service Monitoring 2-20
    - HTTP Cache Freshness 2-21
  - General Settings 2-22
    - Content Management 2-22
    - Login Access Control 2-23
    - Authentication 2-29
    - Storage 2-34
    - Network 2-35
    - Notification and Tracking 2-44
    - Service Router Settings 2-50
    - Cache Router 2-52
    - DNS Based Redirection 2-54
  - Routing Settings 2-56
    - Request Routing General Settings 2-56
- Service APIs 2-60
  - Deliver Service 2-60
    - Service Engine Settings 2-60
    - Content Deletion 2-61

Other APIs 2-63
HTTPS General Settings 2-63
CRL File Schedule 2-64
Show Commands 2-66
File Management 2-67

CHAPTER 3 **CDSM Legacy APIs 3-1**
Replication Status APIs 3-1
Replication Status API Actions 3-1
getDeliveryServices 3-1
getSEsOfDeliveryService 3-2
getDelivheryServicesOfSE 3-2
getReplicatedContent 3-2
getNonReplicatedContent 3-3
getContent 3-4
getStatusOfContentItems 3-4
getStatusOfContentItemInDeliveryService 3-5
XML-Formatted Output for Replication Status 3-5
Provisioning APIs 3-6
Delivery Service Provisioning API Actions 3-6
createDeliveryService 3-8
createDeliveryServiceLoc 3-9
createDeliveryServiceGenSettings 3-9
addManifest 3-11
assignSEs 3-12
assignDeliveryServiceIp 3-13
fetchNow 3-13
modifyDeliveryService 3-14
modifyDeliveryServiceLoc 3-15
modifyDeliveryServiceGenSettings 3-15
modifyManifest 3-16
unassignSEs 3-17
unassignDeliveryServiceIp 3-18
deleteDeliveryServices 3-19
deleteDeliveryServiceGenSettings 3-19
addContentItem 3-19
modifyContentItem 3-21
deleteContentItem 3-22
processContentChanges 3-23

manageHostProxySettings 3-23
createContentOrigin 3-24
modifyContentOrigin 3-25
deleteContentOrigin 3-26
applyRuleFile 3-26
applyGeoIpFile 3-27
configFailoverSettings 3-27
createFailoverOS 3-28
modifyFailoverOS 3-28
deleteFailoveOS 3-29
switchToOS 3-29
createChannelDeviceMcastConfig 3-29
getChannelDeviceMcastConfig 3-30
modifyChannelDeviceMcastConfig 3-30
deleteChannelDeviceMcastConfig 3-31
Location Provisioning API Actions 3-31
createLocation 3-31
modifyLocation 3-32
deleteLocation 3-32
Service Engine Provisioning API Actions 3-33
activateSe 3-33
changeSeLocation 3-33
deleteSe 3-33
setSeMgmtIp 3-34
setMulticast 3-34
Program API Actions 3-35
createProgram 3-35
validateProgramFile 3-36
assignDeliveryService 3-36
assignSEs 3-36
fetchNow 3-37
modifyProgramFile 3-37
unassignDeliveryService 3-38
unassignSEs 3-38
deletePrograms 3-39
Media API Actions for Programs 3-39
addMedia 3-40
updateMedia 3-40
deleteMedia 3-41
URL Management API Actions 3-41

singleURLRemoval 3-41
batchURLRemoval 3-42
Cache Storage Priority Class API Actions 3-45
createStoragePrioClass 3-45
modifyStoragePrioClass 3-46
deleteStoragePrioClass 3-46
Multicast Cloud API Actions 3-47
createCloud 3-47
modifyCloud 3-48
deleteCloud 3-49
assignReceiverSe 3-50
unassignReceiverSe 3-50
assignDeliveryService 3-51
unassignDeliveryService 3-51
modifyChannelMCast 3-52
External System API Actions 3-52
create 3-53
modify 3-53
delete 3-54
Listing APIs 3-54
Listing API Actions 3-54
getContentOrigins 3-56
getDeliveryServices 3-56
getSEs 3-56
getClusters 3-57
getLocations 3-57
getDeviceGroups 3-57
getObjectById 3-59
getObjectByName 3-59
getPrograms 3-60
getPgmMcastAddrInUse 3-60
getMcastAddrInUse 3-61
getMCastClouds 3-61
getStoragePrioClasses 3-61
getExternalSystem 3-62
Device API Actions 3-62
getDeviceStatus 3-62
getDevices 3-62
Statistics APIs 3-63
Monitoring Statistics API Actions 3-63

getSeStats 3-63
getLocationStats 3-64
getCdnStats 3-65
Streaming Statistics API Actions 3-66
getHttp 3-67
getMovieStreamer 3-67
getWmt 3-67
XML-Formatted Output for Streaming Statistics 3-68
File Management APIs 3-69
File Management API Actions 3-69
listTypes 3-70
registerFile 3-71
validateFile 3-72
refetchFile 3-74
modifyFile 3-74
deleteFile 3-75
listFile 3-76
applyCZ 3-77
applyCdnSelector 3-77
Certificate and Key File Management API 3-78
registerFile 3-78
modifyFile 3-79
deleteFile 3-79
listFile 3-80

CHAPTER 4 **Request Routing Engine APIs** 4-1
Request Routing Engine APIs 4-1
Last-Resort URL Translator Web Services API 4-2

CHAPTER 5 **Proximity Engine SOAP APIs** 5-1
Routing Concepts and Overview 5-1
Terminology 5-2
Proximity SOAP API Actions 5-2
rate API 5-3
Fault Message 5-5
Proximity Engine WSDL File 5-7

APPENDIX A **Program Files in the VDS Software** A-1
Program File DTD A-2

Program File Examples A-6
WMT Multicast Live Event A-6
WMT Multicast Rebroadcast Event A-7
Movie Streamer Multicast Event A-7
Movie Streamer Live-Split Event A-8

Preface

This preface describes who should read the *Cisco VDS Internet Streamer 3.3 API Guide*, how it is organized, and its document conventions. It contains the following sections:

- Audience, page xi
- Document Organization, page xii
- Document Conventions, page xii
- Related Publications, page xiii
- Obtaining Documentation and Submitting a Service Request, page xiii

Document Revision History

Table 1This section records technical changes to this document. The table shows the document revision number for the change, the date of the change, and a brief summary of the change.

Table 1 Document Revision History

Revision	Date	Change Summary
—	June 8,2016	Introduced multicast rearm timer API in Multicast Cloud.
OL-31408-02	April 16, 2014	Introduced the Delivery Service Restful API and modified APIs for the VDS-IS Release.3.3.1.
OL-31408-01	December 2013	Initial release.

Audience

This application program interface (API) guide is written for the knowledgeable application programmer who understands the basic architecture of the VDS software product and Java servlets. The user should be fluent in the Java programming language and have prior practical experience developing content networking solutions. This guide is not intended to direct the user in how to program in the Java language and limits itself to describing how related VDS software servlets are used.

Document Organization

This API guide includes the following chapters:

Chapter or Appendix	Title	Description
Chapter 1	Introduction to Cisco CDS Software APIs	Provides an introduction to the CDS software application program interfaces.
Chapter 2	CDSM RESTful APIs	Describes the CDSM RESTful APIs.
Chapter 3	CDSM Legacy APIs	Describes the CDSM Legacy APIs.
Chapter 4	Request Routing Engine APIs	Describes the Request Routing Engine APIs.
Chapter 5	Proximity Engine SOAP APIs	Describes the Proximity Engine APIs.
Appendix A	Program Files in the VDS Software	Describes the attributes of program files and examples for different program types.

Document Conventions

This API guide uses basic conventions to represent text and table information.

Convention	Description
boldface font	Commands, keywords, and button names are in **boldface**.
italic font	Variables for which you supply values are in *italics*. Directory names and filenames are also in italics.
`screen` font	Terminal sessions and information the system displays are printed in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
`italic screen` font	Variables you enter are printed in *`italic screen`* font.
string	Defined as a nonquoted set of characters. For example, when setting a community string for SNMP to “public,” do not use quotation marks around the string, or the string will include the quotation marks.
vertical bars (|)	Vertical bars separate alternative, mutually exclusive, elements.
< >	Variable for which you supply a value.
{ }	Elements in braces are required elements.
[ ]	Elements in square brackets are optional.
{**x** | **y** | **z**}	Required keywords are grouped in braces and separated by vertical bars.
[**x** | **y** | **z**]	Optional keywords are grouped in brackets and separated by vertical bars.
[{ }]	Braces within square brackets indicate a required choice within an optional element.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in the manual.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Related Publications

These documents provide complete information about the VDS and are available from the Cisco.com site:

- *Cisco VDS Internet Streamer 3.3 Software Configuration Guide*
- *Cisco VDS Internet Streamer 3.0–3.1 Quick Start Guide*
- *Cisco VDS Internet Streamer 3.3 Command Reference*
- *Cisco VDS Internet Streamer 3.3 Alarms and Error Messages Guide*
- *Release Notes for Cisco VDS Internet Streamer 3.3.1*
- *Cisco VDS Internet Streamer 3.0–3.3 Software Installation Guide for non-CDEs*
- *Cisco Content Delivery Engine 205/220/250/420 Hardware Installation Guide*
- *Regulatory Compliance and Safety Information for Cisco Content Delivery Engines*
- *Open Sources Used in VDS IS Release 3.3*

You can access the software documents at the following URL:

http://www.cisco.com/en/US/products/ps7127/tsd_products_support_series_home.html

You can access the hardware documents for the CDEs at the following URL:

http://www.cisco.com/en/US/products/ps7126/tsd_products_support_series_home.html

You can access the hardware documents for non-CDEs at the following URLs:

- *Cisco UCS C200 Installation and Service Guide*
  http://www.cisco.com/en/US/docs/unified_computing/ucs/c/hw/C200M1/install/c200M1.html
- *Cisco UCS C210 Installation and Service Guide*
  http://www.cisco.com/en/US/docs/unified_computing/ucs/c/hw/C210M1/install/C210M1.html

Obtaining Documentation and Submitting a Service Request

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

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

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

Introduction to Cisco CDS Software APIs

Starting from 3.3, CDSM provides HyperText Transport Protocol Secure (HTTPS) web services APIs that comply with the Representational State Transfer (REST) standard, Java API for XML RESTful Web Services (JAX-RS), version JSR-311. RESTful APIs provides function to manage Devices and Device Groups.

CDSM also provides HyperText Transport Protocol Secure (HTTPS) application program interfaces (APIs) for monitoring and managing the acquisition and distribution of content. To distinguish with RESTful APIs, it's referred as Legacy APIs in this document.

The Request Routing Engine on the Service Router implements an API that allows another platform's software client to make queries, in the form of an HTTP request, to the Request Routing Engine about which Service Engine the Request Routing Engine selects.

For last-resort URL translator, a Web Services API is used to communicate with the third-party URL translator.

The Proximity Engine on the Service Router implements a rating API on its Simple Object Access Protocol (SOAP) interface. The rating API calculates the proximity of a group of proximity target addresses (PTAs) to a proximity source address (PSA).

This chapter contains the following sections:

- CDSM RESTful APIs, page 1-1
- CDSM Legacy API, page 1-4
- Request Routing Engine APIs, page 1-15
- Proximity Engine SOAP API, page 1-15

CDSM RESTful APIs

In RESTful web services APIs, the URL is used to uniquely identify a resource that can be mapped to one domain object or a collection of domain objects. Each URL (called a resource URL) exposes uniform interfaces to the API clients. The API clients call the URLs by way of the standard HTTP methods of POST, GET, PUT, and DELETE. The HTTP methods are used to describe the create, read, update, and delete (CRUD) actions to be performed.

CDSM exposes RESTful web service APIs through HTTPS by way of port 8443. Extensible Markup Language (XML) is used as the data format for the request body and response body.

The CDSM provides the following RESTful APIs:

- Device APIs

- Service APIs
- Other APIs

Note All URLs starting with "/api/" are RESTful APIs that are intercepted by the CDSM REST service.

Authentication and Authorization

CDSM uses a basic authentication method to authenticate the API user. As an example, the following GET request returns all existing origin services in the response body:

```
https://admin:default@<CDSM_ip>:8443/api/SEs
```

CDSM uses Role Based Access Control (RBAC) for both GUI and API access. Users are created and configured by way of the CDSM GUI pages (System > AAA). The CDSM does not differentiate between GUI users and API users. For more information, see the Cisco VDS IS 3.3 Software Configuration Guide or the CDSM online help

ID Format

All ID variables ({id}, {id2}, and so on.) in the URL templates (URLs with variables are called URL templates) should be substituted with the long integer value of the corresponding objects.For example, the origin service URL is https://admin:default@<CDSM_ip>:8443/api/SEs/{id}; so for a CDSM with an IP address of 192.168.1.25 and an SE ID of 193, the URL is

```
https://admin:default@192.168.1.25:8443/api/SEs/193.
```

Note The admin password, default, is the built-in admin password, and is used as an example. We strongly recommend that the built-in admin password be changed as soon as possible. To do so, log in to the CLI of the CDSM device, and use the **username** *admin* **password** <*password*> global configuration command.

The "id," and "uri" are identity attributes returned along with the XML response body to the API client.

As a general rule, the POST method is used to create an object that is allocated a new ID, and the PUT method is used to modify an object that is identified by a resource URL. The ID is part of the URL if a resource type has multiple objects; otherwise, the ID is not needed if the resource object represented by the URL only has one instance

Calling the RESTful APIs

The CDSM RESTful APIs can be executed interactively or through a caller program. API calls must follow the correct syntax. If the user credential is invalid or the syntax is incorrect, the API is not executed. If a user error occurs, a specific standard HTTP error code is returned (for example, 400, 403, 404, 500, and so on).

Note All API parameters are case sensitive.

Two headers must be set for the RESTful API requests:

- Accept -Accept header must be specified to indicate what content type the API client expects to receive from CDSM RESTful API services, supported values are: application/xml, or application/json
- Content-Type-Content-Type header must be specified to indicate the content type of the request body the API client sends to the CDSM RESTful API services, supported values are: application/xml, or application/json

Interactive Calls

Use a REST client plug-in for a browser or curl command to execute the API interactively. The browser address bar only supports the GET method; however, the CDSM APIs use all HTTP methods. The user is prompted to enter a username and password for authentication and authorization. Once the user is validated, the API is executed. If the execution is successful and an output is to be returned as a result, the output is displayed in the browser if a browser was used to make the API call, or the output can be redirected to a file if a curl command was used to make the API call. If the execution is unsuccessful, an error message is returned.

Programmed Calls

To make an API call, write a caller program using an HTTPS request. The username and password are set in the HTTPS request for AAA validation. If validation and execution are successful and an output is to be returned as a result, the output or a success code is returned. If the execution is unsuccessful, a failure code is returned.

API Status Codes

The CDSM returns HTTP response codes.

Error Status Codes

If a server error occurs while the APIs are invoked, the error message is reflected in an HTTP response code; for example, 404 not found. All error codes are standard HTTP response codes. If an error occurs, an XML-formatted or json- formatted message is returned.

The following HTTP error status codes may be returned:

- 400 Bad Request. (When the request body is not valid XML message required by specific APIs)
- 403 Authorization failure
- 404 Not found: the requested URL or the requested resource object doesn't exist.
- 500 Internal Error

Following is an example of error case:

```
curl -u admin:default -k -i -H "Content-Type: application/xml" -X GET
"https://10.74.23.40:8443/api/SEs/111"
HTTP/1.1 404 Not Found
Date: Wed, 24 Oct 2012 08:13:20 GMT
Server: Apache
Content-Length: 175
Content-Type: application/xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorResponse uri="/api/LiveChannels/111">
    <error>The requested object is not found: 111</error>
</ErrorResponse>

Following is an example of an XML-formatted message:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ErrorResponse uri="/api/LiveChannels/111">
    <error>The requested object is not found: 111</error>
</ErrorResponse>

Following is an example of a json-formatted message:
{
   "@uri": "/api/SEs/22",
   "error": "The requested object is not found: 22"
}
```

Success Status Codes

Success status codes consist of 201, 200 OK and 204 No Content (OK with empty response body). The 204 codes may be returned for the following:

- POST-For create operation, the Location header of the response returns the URI of the new created resource object. The response body is empty.
- PUT-Used when modifying a resource object. The modify request is handled, 204 is returned with empty response in the response body.
- DELETE-Used when deleting a resource object. The delete request is handled, 204 is returned with empty response in the response body.

The 201 code is returned for

- POST-content deletion, The Location header of the response is the URL of the deletion task. The response body is empty.

CDSM Legacy API

The CDSM provides the following Legacy APIs:

- Replication Status APIs
- Delivery Service Provisioning
- Location Provisioning
- Service Engine Provisioning
- Program
- Media for Programs
- URL Management
- Listing
- Monitoring Statistics
- Streaming Statistics
- File Management APIs
- Certificate and Key File Management

These Legacy APIs are Java servlets whose return outputs are generated in XML format. CDS software uses these servlets to monitor and modify specified content acquisition and distribution parameters. Table 1-1 describes these APIs. For most API actions, a unique website and delivery service name must be provided to the API so that the delivery service can be located.

Table 1-1 Cisco CDS Software Legacy APIs

API	Description
Replication Status	Returns a list of delivery services, Service Engines, or contents, and for each delivery service, an indication whether replication of content for the specified delivery service is complete or not.
Provisioning APIs	Provides the CDSM[1] with CDS delivery service, location, and Service Engine information. • Delivery Service Provisioning API—Monitors and modifies CDS network delivery services. • Location Provisioning API—Creates, modifies, or deletes a CDS network location object. • Service Engine Provisioning API—Activates, locates, or deletes a specified Service Engine. • Program API—Creates, modifies, validates, or deletes programs, and assigns or unassigns Service Engines and delivery services to programs. • Media API—Adds or deletes a media file from a Movie Streamer rebroadcast program, and updates the media file list for a Movie Streamer rebroadcast program. • URL Management API—Deletes single or multiple content objects. • Storage Priority Class API—Creates, modifies, and deletes a cache storage priority class used for delivery services. • Multicast API—Creates, modifies, and deletes multicast clouds, assigns and unassigns receiver SEs, and assigns and unassigns multicast cloud to delivery services. • External System API—Creates, modifies, and deletes external system configurations for forwarding SNMP traps from the CDSM.
Listing API	Obtains object information from the local embedded database.
Monitoring Statistics API	Obtains monitoring statistics data about a single Service Engine or all the Service Engines in the CDS network.

Table 1-1 Cisco CDS Software Legacy APIs (continued)

API	Description
Streaming Statistics API	Reports WMT[2], HTTP, Movie Streamer, and Flash Media data collected from the Service Engines or device groups and sends this data to the CDSM. Data obtained with the Streaming Statistics API can be saved and a customized report generated.
File Management APIs	Performs file management functions on the external XML files used to configure the CDS, and applies Coverage Zone and CDN Selector files to SRs[3]. • File Management API—Manges XML files registered to the CDSM • Certificate and Key File Management API—Manages the HTTP Streaming Certificate and Key file

1. CDSM = Content Delivery System Manager.
2. WMT = Windows Media Technology.
3. SRs = Service Routers.

CDS software also provides authentication, authorization, and accounting (AAA) functions to support users who access external servers and local databases. Authentication verifies the identity and IP address of a user, authorization permits or denies access privileges for authenticated users in the CDS network, and accounting logs authorized usage of network services. These AAA functions are enforced by the APIs so that user credentials must be validated before an API can be executed.

Calling the Legacy APIs

You can execute the Legacy APIs interactively or through a caller program. API calls must follow the correct syntax. If the user credential is invalid or the syntax is incorrect, the API is not executed. If a user error occurs, a warning is returned that explains the nature of the error along with the syntax of the particular API.

Note All API parameters are case sensitive.

Interactive Calls

Use a browser or Lynx command to execute the API interactively. The user is prompted to enter a username and password for authentication and authorization. Once the user is validated, the API is executed. If the execution is successful and an output is to be returned as a result, the output is displayed in the browser if a browser was used to make the API call, or the output can be redirected to a file if a Lynx command was used to make the API call. If the execution is unsuccessful, an error message is returned.

Programmed Calls

To make an API call, write a caller program using an HTTPS request. The username and password are set in the HTTPS request for AAA validation. If validation and execution are successful and an output is to be returned as a result, the output or a success code is returned. If the execution is unsuccessful, a failure code is returned.

Sample Java Program

The following is a sample Java client program that requires two Simple API for XML (SAX) parsing APIs. This sample code requires the "org.xml.sax.*" API and "org.xml.sax.helpers.*" API for the parser and the HTTPS URL package for the connection.

```
package testing.download.client;

import java.io.BufferedReader;
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.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;

import javax.net.ssl.X509TrustManager;


public class Client
{

    public static void main (String[] args)
    {
        /**
         * Setting parameters for the API call
         */
        String cdmAddress_ = "cds-demo-cdsm.cds.cisco.com";
        String cdmPort_ = "8443";
        String taskAPI_ = "com.cisco.unicorn.ui.ListApiServlet";
        String action_ = "getDeliveryServices";
        String channelId_ = "all";
        String urlString_ = "https://" + cdmAddress_ + ":" + cdmPort_ + "/servlet/" +
                    taskAPI_+"?action=" + action_ + "&param=" + channelId_;
        String userName_ = "admin";
        String password_ = "default";

        /**
          * Install the all-trusting trust manager
          */
        try {
            SSLContext sc = SSLContext.getInstance("SSL");
            sc.init(null, trustAllCerts, new java.security.SecureRandom());
            HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
        }
        catch (Exception e) {
            System.out.println("Printing Exception Message "+e);
        }

        /**
```

```
         * Insert the credentials
         */
        String sAuth = userName_+":"+password_;
        String sEncodedAuth = new sun.misc.BASE64Encoder().encode(sAuth.getBytes());
        /**
          * Create the HTTPS Connection
         */
        HttpsURLConnection conn = null;
        try {
            URL url = new URL(null, urlString_ );

            System.out.println(url.toString());

            conn = (HttpsURLConnection)url.openConnection();
            conn.setRequestProperty("Authorization", "Basic " + sEncodedAuth);
            conn.setHostnameVerifier(new newHostNameVerifier());
            conn.setDoInput(true);
            conn.setDoOutput(true);
            conn.setUseCaches(false);
            conn.setRequestProperty("Connection", "Keep-Alive");
            conn.setRequestMethod("GET");

        }
        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);
        }
    }

/**
 * Create a trust manager that does not validate certificate chains
 */
    private static TrustManager[] trustAllCerts = new TrustManager[]{
            new X509TrustManager() {
            public java.security.cert.X509Certificate[] getAcceptedIssuers() {
                    return null;
                }
                public void checkClientTrusted(
                   java.security.cert.X509Certificate[] certs, String authType) {
```

```
                }
                public void checkServerTrusted(
                    java.security.cert.X509Certificate[] certs, String authType) {

                }
            }
    };

    private static class newHostNameVerifier implements HostnameVerifier {

        /**
         * ignore hostname checking
         */
        public boolean verify(String hostname, SSLSession session) {
          return true;
        }
    }

}
```

API Error Messages

When a server error occurs while the APIs are invoked, an XML-formatted message is returned. For example, when Internal Server Error—500 occurs, the client sees the following output:

```
<?xml version="1.0"?>
<Error>
<message status="fail" message="Internal Server Error –5 00"/>
</Error>
```

The following common errors are supported in the message syntax:

- Bad Request—400
- Authorization Required—401
- Forbidden—403
- File Not Found—404
- Request Timeout—408
- Internal Server Error—500

Typically, APIs return error messages when API execution fails. If the execution is successful, APIs do not return any error messages. However, APIs may return warning messages even when the execution is successful.

APIs use numeric error and warning codes. Table 1-2 describes the generic numeric codes used for errors and warnings. Table 1-3 describes some of the numeric codes used by the Program and File Management in API errors and warnings.

Table 1-2 Numeric Codes for Errors and Warnings in APIs

Error or Warning Code	Description
0	None
1	Syntax error
2	Input error

Table 1-2 Numeric Codes for Errors and Warnings in APIs

Error or Warning Code	Description
3	Constraint error
4	Input warnings

Table 1-3 Numeric Codes for Errors and Warnings in the Program and File Management APIs

Error or Warning Code	Description
101	Unable to fetch the file
102	File syntax error
103	Invalid value in the file
104	Related system error
105	Program file unused input - Warning

For example, when you enter the following URL to execute an API to delete a selected type of program:

https://<cdsm:port>/servlet/com.cisco.unicorn.ui.ProgramApiServlet?action=deletePrograms&program=type=wmt

and no programs of that type exist, the API returns the subsequent warning.

```
<?xml version="1.0" ?>
<programApi action="deletePrograms">
<message status="success" message="The program(s) are deleted." />
<warning code="4" message="No Program(s) that matched the request were found" />
</programApi>
```

Similarly, when you enter the following URL to execute an API to delete a delivery service:

https://<cdm:port>/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteDeliveryServices&deliveryService=Channel_333

with an invalid delivery service ID, the API returns the subsequent error.

```
<?xml version="1.0" ?>
<channelProvisioning action="deleteDeliveryServices">
<message status="fail" message="Input Error: Cannot locate delivery service using delivery
service ID Channel_333" />
<error code="2" message="Input Error: Cannot locate delivery service using delivery
service ID Channel_333" />
</channelProvisioning>
```

API Tasks

The following sections provide a brief list of tasks performed by the Replication Status, Provisioning, Listing, and Statistics APIs.

Replication Status API

The Replication Status API performs one or more of the following tasks when executed:

- Obtains the replication status of content on specified delivery services
- Obtains the replication status of content for all Service Engines assigned to the specified delivery service
- Obtains the replication status of content for all delivery services assigned to the specified Service Engine
- Lists all replicated items of a specified Service Engine on a specified delivery service, with or without search criteria
- Lists all nonreplicated items of a specified Service Engine on a specified delivery service, with or without search criteria
- Lists all content items of a Service Engine on a specified delivery service, with or without search criteria

Provisioning APIs

The Provisioning APIs include the Delivery Service Provisioning API, Location Provisioning API, Service Engine Provisioning API, and Program API.

Delivery Service Provisioning API

The Delivery Service Provisioning API performs one or more of the following tasks when executed:

- Creates, modifies, and deletes delivery services
- Creates, modifies, and deletes delivery service content
- Creates, modifies, and deletes delivery service settings
- Adds and modifies a Manifest file to a specified delivery service
- Immediately fetches the Manifest file
- Assigns Service Engines to, and removes Service Engines from a specified delivery service
- Assigns an IP address of a Service Engine to a specified delivery service
- Removes IP addresses of a Service Engine from a specified delivery service
- Creates, modifies, and deletes content origins
- Applies Service Rule files and Geo/IP files to delivery services

Location Provisioning API

The Location Provisioning API performs one or more of the following tasks when executed:

- Creates a specified location
- Modifies a specified location

- Deletes a specified location

Service Engine Provisioning API

The Service Engine Provisioning API performs one or more of the following tasks when executed:

- Activates a specified Service Engine
- Changes the location of a specified Service Engine
- Deletes a specified Service Engine

Program API

The Program API performs one or more of the following tasks when executed:

- Creates a program file
- Validates a program file
- Assigns delivery services to a specified program
- Assigns Service Engines to a specified program
- Fetches a program file
- Modifies a program file
- Removes delivery services from a specified program
- Removes Service Engines from a specified program

Media API

The Media API performs one or more of the following tasks when executed:

- Adds or deletes a media file from a Movie Streamer rebroadcast program
- Updates the media file order of a Movie Streamer rebroadcast program

URL Management API

The URL Management API performs one or more of the following tasks when executed:

- Removes content items of delivery service by single URL
- Removes content items or delivery service by URL batch file, which contains a set of URLs

Cache Storage Priority Class API

The Cache Storage Priority Class API performs one or more of the following tasks when executed:

- Creates a storage priority class
- Modifies a storage priority class
- Deletes a storage priority class

Multicast Cloud API

The Multicast Cloud API performs one or more of the following tasks when executed:

- Creates a multicast cloud
- Modifies a multicast cloud

- Deletes a multicast cloud
- Assigns receiver SEs to a multicast cloud
- Removes receiver SEs from a multicast cloud
- Assigns a multicast cloud to a delivery service
- Removes a multicast cloud from a delivery service

External System API

The External System API performs one or more of the following tasks when executed:

- Creates an external system
- Modifies an external system
- Deletes an external system

Listing API

The Listing API performs one or more of the following tasks when executed:

- Lists selected content origin names or lists every content origin
- Lists selected delivery service names and related content origin IDs or lists every delivery service
- Lists selected Service Engine names or lists every Service Engine
- Lists the location of the specified Service Engines
- Lists selected cluster names or lists every cluster (cluster is the same thing as Service Engine)
- Lists selected device group names or lists every device group
- Lists the status of a device or device group
- Lists an object, based on its string ID
- Lists an object, based on its name
- Lists all programs specified
- Lists all multicast addresses currently in use by programs
- Lists all multicast addresses currently in use
- Lists the multicast address range reserved for programs
- Lists all the multicast clouds.
- Lists the cache storage priority classes
- Lists the external systems configured

Statistics API

The Statistics APIs include the Monitoring Statistics API and the Streaming Statistics API.

Monitoring Statistics API

The Monitoring Statistics API performs one or more of the following tasks when executed:

- Obtains monitoring statistics for each Service Engine
- Obtains monitoring statistics for all the Service Engines in a location

- Obtains monitoring statistics for all the Service Engines in the CDS network

Streaming Statistics API

The Streaming Statistics API performs one or more of the following tasks when executed:

- Reports HTTP statistics for each Service Engine or device group
- Reports Movie Streamer statistics for each Service Engine or device group
- Reports WMT statistics for each Service Engine or device group

File Management API

The File Management APIs include the File Management API and the Certificate and Key File Management API.

File Management API

The File Management API performs one or more of the following tasks when executed:

- Displays a list of all the file types that can be registered with the CDSM
- Registers an external file with the CDSM by either uploading a file from any location that is accessible from your PC or by importing a file from an external server
- Validates a file before or after registering it with the CDSM
- Modifies the metadata associated with a registered file
- Immediately refetches a registered file from an external server
- Deletes a registered file from the CDSM
- Lists the details of a specific file or lists all files of a specific file type
- Assigns a Coverage Zone file to an SR or unassigns a Coverage Zone file from an SR
- Associates a CDN Selector file with an SR or disassociates a CDN Selector file from an SR

Certificate and Key File Management API

The Certificate and Key File Management API performs one or more of the following tasks when executed:

- Registers the Certificate and Key file for HTTPS Streaming
- Modifies (updates) the Certificate and Key file for HTTPS Streaming
- Deletes the Certificate and Key file for HTTPS Streaming
- Lists the details of the Certificate and Key file for HTTPS Streaming

Request Routing Engine APIs

Request Routing Engine API

The Request Routing Engine API returns the name of the Service Engine which the Request Routing Engine selects as the best Service Engine on the basis of a Client IP address and a URL provided in the calling API.

Note The Request Routing Engine API does not support service-aware routing.

Last-Resort URL Translator Web Services API

The last-resort URL Translator API uses a Web Services Description Language (WSDL) for communicating between the Request Routing Engine and the third-party URL translator server.

Proximity Engine SOAP API

The Proximity Engine exposes a rating API on its SOAP interface to a proximity client (SR). SOAP is an XML-based messaging protocol for invoking remote procedures by sending XML messages over application layer protocols (for example, HTTP). The SR leverages the rate API to determine the network difference between a PSA and a PTA in order to choose the PTA within closest proximity to the PSA. PTAs returned by the rate API are ranked in ascending order based on the rating each has received. If an error occurs while the rating API is invoked, an XML-formatted fault message is returned. When the Proximity Engine does not consider itself the most appropriate Proximity Engine to service the request, an XML-formatted redirect fault messages is returned. In this message, the Proximity Engine redirects the proximity client to a set of Proximity Engines it considers more appropriate.

Note The Proximity Engine API is available on the CDE205 and CDE220-2G2 platforms.

CDSM RESTful APIs

This chapter describes the CDSM RESTful APIs, which consist of the following:

- Device APIs
- Service APIs
- Other APIs

Device APIs

Device Groups

This API provides parity for the following CDSM GUI page:

Devices>Device Groups

Table 2-1 Device Groups API Calls

Resource URL	Method	Function Description
/api/groups	GET	Get the list of all Device Groups.
	POST	Create a new Device Group.
/api/groups/{id}	GET	Get a Device Group identified by {id}.
	PUT	Modify a Device Group identified by {id}.
	DELETE	Remove a Device Group identified by {id}.

Examples

- Get a Device Group

  Get a Device Group identified by id 3185

Request URL: /api/groups/3185

Request Method: GET

Request XML: None

Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<group id="3185" uri="/api/groups/3185">
  <Name>dg1</Name>
  <Description></Description>
  <Type>1</Type>
  <AutoAssignCe>false</AutoAssignCe>
</group>
```

- Create a Device Group

  Create a Device Group.

  Request URL: /api/groups

  Request Method: POST

  Request XML:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <group>
      <Name>dg1</Name>
      <Description></Description>
      <Type>1</Type>
      <AutoAssignCe>false</AutoAssignCe>
  </group>
  ```

  Response XML: None

- Modify a Device Group

  Modify a Device Group identified by id 3185.

  Request URL: /api/groups/3185

  Request Method: PUT

  Request XML:

  ```
  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <group>
      <Name>dg4</Name>
      <Description></Description>
      <Type>1</Type>
      <AutoAssignCe>false</AutoAssignCe>
  </group>
  ```

  Response XML: None

- Remove a Device Group

  Remove a Device Group identified by id 3185.

  Request URL: /api/groups/3185

  Request Method: DELETE

  Request XML: None

Device Group Assignments

This API provides parity for the following CDSM GUI page:

Devices>Device Groups>{DG}>Assignments>Devices

Table 2-2 Device Group Assignments API Calls

Resource URL	Method	Function Description
/api/groups/{id}/SEs	GET	Get all SEs assigned to a Device Group.
	POST	Assigns or Unassigns SEs to a Device Group.

Examples

- Get all SEs assigned to a Device Group

  Get all SEs assigned to a Device Group identified by id 304.

  Request URL: api/groups/304/SEs

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SEs uri="/api/groups/304/SEs">
  <SE id="234" uri="/api/groups/304/SEs/234">
    <Name></Name>
    <Description></Description>
    <name>SE_1</name>
    <status>Online</status>
    <isServerOffload>false</isServerOffload>
    <ipAddress>10.79.53.183</ipAddress>
    <location>SE-location</location>
    <version>3.3.0.b.33</version>
  </SE>
</SEs>
```

- Assign or un-assign SEs to a Device Group.

  Assign or un-assign SEs to a Device Group identified by 304.

  Request URL: /api/groups/304/SEs

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<groups>
    <assignIds>234</assignIds>
    <removeIds>235</removeIds>
</groups>
```

  Response XML: None

Locations

This API provides parity for the following CDSM GUI page.

Devices>Locations

Table 2-3 Locations API Calls

Resource URL	Method	Function Description
/api/locations	GET	Get all Locations.
	POST	Create a Location.
/api/locations/{id}	GET	Get a Location identified by {id}.
	DELETE	Remove a Location identified by {id}.

Examples

- Get Locations

  Get all Locations.

  Request URL: /api/locations

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<locations uri="/api/locations">
  <locations-row id="405" uri="/api/locations/405">
     <name>SE-location2</name>
     <description>None</description>
     <level>1</level>
     <parentLocation></parentLocation>
  </locations-row>
  <locations-row id="239" uri="/api/locations/239">
     <name>SE-location</name>
     <description>default location for SE</description>
     <level>1</level>
     <parentLocation></parentLocation>
  </locations-row>
</locations>
```

- Create Locations

  Create a Location.

  Request URL: /api/locations

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<location>
```

```
    <name>SE-location4</name>
    <description>none</description>
    <level>1</level>
    <parentLocation></parentLocation>
</location>
```

Response XML: None

- Remove Locations

  Remove a location identified by id 239.

  Request URL: /api/locations/239

  Request Method: DELETE

  Request XML: None

  Response XML: None

Replication

Default Bandwidth

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Replication>Default Bandwidth

Table 2-4 Default Bandwidth API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/defaultRepBw /api/groups/{DG_id}/defaultRepBw	GET	Get the Acquisition and Distribution Default Bandwidth settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Acquisition and Distribution Default Bandwidth settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Acquisition and Distribution Default Bandwidth settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify a Default Bandwidth

  Modify Default Bandwidth of an SE identified by id 234.

  Request URL: /api/SEs/234/defaultRepBw

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<defaultRepBw>
```

```
  <AcquisitionIn>100000</AcquisitionIn>
  <DistributionIn>100000</DistributionIn>
  <DistributionOut>100000</DistributionOut>
</defaultRepBw>
```

Response XML: None

Distribution

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Replication>Distribution

Table 2-5 Distribution API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/distributionSetting /api/groups/{DG_id}/distributionSetting	GET	Get the Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Distribution settings

  Modify Distribution Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/distributionSetting

  Request Method: PUT

  Request XML: None

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<distributionSetting>

<System_AdMul_NextNACKIntervalMultiplier>10</System_AdMul_NextNACKIntervalMultiplier>
</distributionSettingdistributionSetting>
```

  Response XML: None

Multicast Distribution

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Replication>Multicast Distribution

Table 2-6 Multicast Distribution API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/mcastDistribution /api/groups/{DG_id}/mcastDistribution	GET	Get the Multicast Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Multicast Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Multicast Distribution settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Multicast Distribution settings.

  Modify Multicast Distribution settings of an SE identified by id 234.

  Request URL: /api/SEs/234/mcastDistribution

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mcastDistribution>
    <System_mcast_receiver_enable>true</System_mcast_receiver_enable>
    <System_mcast_sender_enable>false</System_mcast_sender_enable>
</mcastDistribution>
```

  Response XML: None

Service Control

Enable Rules

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Service Control>Enable Rules

Table 2-7 Enable Rules API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/ruleEnable /api/groups/{DG_id}/ruleEnable	GET	Get the Enable Rules settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Enable Rules settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Enable Rules settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Enable Rules settings

  Modify Enable Rules Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/ruleEnable

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ruleEnable>
    <System_servicerule_enable>true</System_servicerule_enable>
</ruleEnable>
```

  Response XML: None

Authorization Service

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Service Control>Authorization Service

Table 2-8 Authorization Service API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/authService /api/groups/{DG_id}/authService	GET	Get the Authorization Service settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Authorization Service settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Authorization Service settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Authorization Service settings

  Modify Authorization Service settings of an SE identified by id 234.

  Request URL: /api/SEs/234/authService

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<authService>
  <System_authserv_enable>true</System_authserv_enable>
  <System_authserv_locationserver_enable>true</System_authserv_locationserver_enable>
  <System_authserv_locationserver_primary_host>1.1.1.1</System_authserv_locationserver
   _primary_host>
  <System_authserv_locationserver_primary_port>4333</System_authserv_locationserver_pr
   imary_port>
  <System_authserv_locationserver_secondary_host></System_authserv_locationserver_seco
   ndary_host>
  <System_authserv_locationserver_secondary_port></System_authserv_locationserver_seco
   ndary_port>
  <System_authserv_unknowndomainproxy_enable>false</System_authserv_unknowndomainproxy
   _enable>
</authService>
```

  Response XML: None

Transaction Logging

This API provides parity for the following CDSM GUI pages:

Devices>Devices>Device Groups>{SE|DG}>Service Control>Transaction Logging

Devices>Devices>{SR}>General Settings>Notification and Tracking>Transaction Logging

Table 2-9 Transaction Logging API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/transacLog /api/SRs/{SR_id}/transacLog /api/groups/{DG_id}/transacLog	GET	Get the Transaction Logging settings for device identified by {SE_id}, {SR_id} or {DG_id}.
	PUT	Modify the Transaction Logging settings for device identified by {SE_id}, {SR_id} or {DG_id}.
	POST	Create the Transaction Logging settings for device identified by {SE_id}, {SR_id} or {DG_id}.
	DELETE	Remove the Transaction Logging settings for device identified by {SE_id}, {SR_id} or {DG_id}.
/api/SEs/{SE_id}/transacLog/applyDefault /api/SRs/{SR_id}/transacLog/applyDefault /api/groups/{DG_id}/transacLog/applyDefault	POST	Apply default values to the Transaction Logging settings for device identified by {SE_id}, {SR_id} or {DG_id}. Request XML is not required.
/api/SEs/{SE_id}/transacLog/applyDG/{DG_id}	POST	Apply Device Groups transaction log settings (identified by {DG_Id}) on the transaction log settings of the device identified by {SE_id}. Request XML is not required.
/api/groups/{DG_id}/transacLog/forceSEs	POST	Force transaction settings on SEs in the group Identified by {DG_id}. Request XML is not required.

Examples

- Get Transaction Logging

  Get Transaction Logging of an SE identified by id 217.

  Request URL: /api/SEs/217/transacLog

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<transacLog uri="SEs/217/transacLog">
  <System_transacLog_enable>false</System_transacLog_enable>
  <System_transacLog_dsSnapCounter>false</System_transacLog_dsSnapCounter>
  <System_transacLog_logWindowsDomain>false</System_transacLog_logWindowsDomain>
  <System_transacLog_compressEnable>false</System_transacLog_compressEnable>
  <System_transacLog_archiveFileSize>500000</System_transacLog_archiveFileSize>
  <System_transacLog_maxArchiveFileNumber>10</System_transacLog_maxArchiveFileNumber>
  <System_transacLog_format>apache</System_transacLog_format>
  <System_transacLog_archiveStr>every-day every 1</System_transacLog_archiveStr>
  <System_transacLog_hostEnable>false</System_transacLog_hostEnable>
  <System_transacLog_host_port>514</System_transacLog_host_port>
  <System_transacLog_host_ratelimit>0</System_transacLog_host_ratelimit>
  <System_transacLog_splunkUF_exportEnable>false</System_transacLog_splunkUF_exportEna
   ble>
  <System_transacLog_exportEnable>true</System_transacLog_exportEnable>
  <System_transacLog_exportStr>every-day every 3</System_transacLog_exportStr>
  <System_transacLog_wmtEnable>true</System_transacLog_wmtEnable>
  <System_transacLog_wmtFormat>wms-41</System_transacLog_wmtFormat>
  <System_transacLog_weABRSession>disable</System_transacLog_weABRSession>
  <System_transacLog_export_skipLogTypes_serviceMonitor>true</System_transacLog_export
   _skipLogTypes_serviceMonitor>
  <System_transacLog_export_skipLogTypes_snapshotCounter>true</System_transacLog_expor
   t_skipLogTypes_snapshotCounter>
  <System_transacLog_export_skipLogTypes_ad>true</System_transacLog_export_skipLogType
   s_ad>
  <System_transacLog_export_skipLogTypes_authSvr>true</System_transacLog_export_skipLo
   gTypes_authSvr>
  <System_transacLog_export_skipLogTypes_contentMgr>false</System_transacLog_export_sk
   ipLogTypes_contentMgr>
  <System_transacLog_export_skipLogTypes_fmsAccess>true</System_transacLog_export_skip
   LogTypes_fmsAccess>
  <System_transacLog_export_skipLogTypes_fmsAuth>true</System_transacLog_export_skipLo
   gTypes_fmsAuth>
  <System_transacLog_export_skipLogTypes_movieStreamer>true</System_transacLog_export_
   skipLogTypes_movieStreamer>
  <System_transacLog_export_skipLogTypes_weABR>true</System_transacLog_export_skipLogT
   ypes_weABR>
  <System_transacLog_export_skipLogTypes_weApache>true</System_transacLog_export_skipL
   ogTypes_weApache>
  <System_transacLog_export_skipLogTypes_weCLF>true</System_transacLog_export_skipLogT
   ypes_weCLF>
  <System_transacLog_export_skipLogTypes_weIngest>true</System_transacLog_export_skipL
   ogTypes_weIngest>
  <System_transacLog_export_skipLogTypes_weKCM>true</System_transacLog_export_skipLogT
   ypes_weKCM>
  <System_transacLog_export_skipLogTypes_weSquid>true</System_transacLog_export_skipLo
   gTypes_weSquid>
  <System_transacLog_export_skipLogTypes_weUrlStats>false</System_transacLog_export_sk
   ipLogTypes_weUrlStats>
  <System_transacLog_export_skipLogTypes_wms41>true</System_transacLog_export_skipLogT
   ypes_wms41>
  <System_transacLog_export_skipLogTypes_wms41Ext>true</System_transacLog_export_skipL
   ogTypes_wms41Ext>
  <System_transacLog_export_skipLogTypes_wms90>true</System_transacLog_export_skipLogT
   ypes_wms90>
  <System_transacLog_export_skipLogTypes_wms90Ext>true</System_transacLog_export_skipL
   ogTypes_wms90Ext>
  <System_transacLog_export_skipLogTypes_wmtIngest>true</System_transacLog_export_skip
   LogTypes_wmtIngest>
</transacLog>
```

- Create or modify Transaction Logging

  Create or modify Transaction Logging of an SE identified by id 217.

Request URL: /api/SEs/217/transacLog

Request Method: POST or PUT

Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<transacLog>
  <System_transacLog_enable>false</System_transacLog_enable>
  <System_transacLog_dsSnapCounter>false</System_transacLog_dsSnapCounter>
  <System_transacLog_logWindowsDomain>false</System_transacLog_logWindowsDomain>
  <System_transacLog_compressEnable>false</System_transacLog_compressEnable>
  <System_transacLog_archiveFileSize>500000</System_transacLog_archiveFileSize>
  <System_transacLog_maxArchiveFileNumber>10</System_transacLog_maxArchiveFileNumber>
  <System_transacLog_format>apache</System_transacLog_format>
  <System_transacLog_archiveStr>every-day every 1</System_transacLog_archiveStr>
  <System_transacLog_hostEnable>false</System_transacLog_hostEnable>
  <System_transacLog_host_port>514</System_transacLog_host_port>
  <System_transacLog_host_ratelimit>0</System_transacLog_host_ratelimit>
  <System_transacLog_splunkUF_exportEnable>false</System_transacLog_splunkUF_exportEna
   ble>
  <System_transacLog_exportEnable>true</System_transacLog_exportEnable>
  <System_transacLog_exportStr>every-day every 3</System_transacLog_exportStr>
  <System_transacLog_wmtEnable>true</System_transacLog_wmtEnable>
  <System_transacLog_wmtFormat>wms-41</System_transacLog_wmtFormat>
  <System_transacLog_weABRSession>disable</System_transacLog_weABRSession>
  <System_transacLog_export_ftp1_server>1.1.1.1</System_transacLog_export_ftp1_server>
  <System_transacLog_export_ftp1_name>aaa</System_transacLog_export_ftp1_name>
  <System_transacLog_export_ftp1_password>aaa</System_transacLog_export_ftp1_password>
  <System_transacLog_export_ftp1_dirpath>aaa</System_transacLog_export_ftp1_dirpath>
  <System_transacLog_export_ftp1_option>ftp-server</System_transacLog_export_ftp1_opti
   on>
  <System_transacLog_export_ipv6_ftp1_server>fec0::1</System_transacLog_export_ipv6_ft
   p1_server>
  <System_transacLog_export_ipv6_ftp1_name>bbb</System_transacLog_export_ipv6_ftp1_nam
   e>
  <System_transacLog_export_ipv6_ftp1_password>bbb</System_transacLog_export_ipv6_ftp1
   _password>
  <System_transacLog_export_ipv6_ftp1_dirpath>bbb</System_transacLog_export_ipv6_ftp1_
   dirpath>
  <System_transacLog_export_ipv6_ftp1_option>ftp-server</System_transacLog_export_ipv6
   _ftp1_option>
  <System_transacLog_export_skipLogTypes_serviceMonitor>true</System_transacLog_export
   _skipLogTypes_serviceMonitor>
  <System_transacLog_export_skipLogTypes_snapshotCounter>true</System_transacLog_expor
   t_skipLogTypes_snapshotCounter>
  <System_transacLog_export_skipLogTypes_ad>true</System_transacLog_export_skipLogType
   s_ad>
  <System_transacLog_export_skipLogTypes_authSvr>true</System_transacLog_export_skipLo
   gTypes_authSvr>
  <System_transacLog_export_skipLogTypes_contentMgr>false</System_transacLog_export_sk
   ipLogTypes_contentMgr>
  <System_transacLog_export_skipLogTypes_fmsAccess>true</System_transacLog_export_skip
   LogTypes_fmsAccess>
  <System_transacLog_export_skipLogTypes_fmsAuth>true</System_transacLog_export_skipLo
   gTypes_fmsAuth>
  <System_transacLog_export_skipLogTypes_movieStreamer>true</System_transacLog_export_
   skipLogTypes_movieStreamer>
  <System_transacLog_export_skipLogTypes_weABR>true</System_transacLog_export_skipLogT
   ypes_weABR>
  <System_transacLog_export_skipLogTypes_weApache>true</System_transacLog_export_skipL
   ogTypes_weApache>
  <System_transacLog_export_skipLogTypes_weCLF>true</System_transacLog_export_skipLogT
   ypes_weCLF>
```

```
  <System_transacLog_export_skipLogTypes_weIngest>true</System_transacLog_export_skipL
   ogTypes_weIngest>
  <System_transacLog_export_skipLogTypes_weKCM>true</System_transacLog_export_skipLogT
   ypes_weKCM>
  <System_transacLog_export_skipLogTypes_weSquid>true</System_transacLog_export_skipLo
   gTypes_weSquid>
  <System_transacLog_export_skipLogTypes_weUrlStats>false</System_transacLog_export_sk
   ipLogTypes_weUrlStats>
  <System_transacLog_export_skipLogTypes_wms41>true</System_transacLog_export_skipLogT
   ypes_wms41>
  <System_transacLog_export_skipLogTypes_wms41Ext>true</System_transacLog_export_skipL
   ogTypes_wms41Ext>
  <System_transacLog_export_skipLogTypes_wms90>true</System_transacLog_export_skipLogT
   ypes_wms90>
  <System_transacLog_export_skipLogTypes_wms90Ext>true</System_transacLog_export_skipL
   ogTypes_wms90Ext>
  <System_transacLog_export_skipLogTypes_wmtIngest>true</System_transacLog_export_skip
   LogTypes_wmtIngest>
</transacLog>
```

Application Control

Default and Maximum Bandwidth

Devices>Devices>Device Groups>{SE|DG}>Application Control>Default and Maximum Bandwidth

Table 2-10 Default and Maximum Bandwidth API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/appBw /api/groups/{DG_id}/appBw	GET	Get the Default and Maximum Bandwidth settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Default and Maximum Bandwidth settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Default and Maximum Bandwidth for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Default and Maximum Bandwidth

  Modify Default and Maximum Bandwidth settings of an SE identified by id 204.

  Request URL: /api/SEs/204/appBw

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<appBw>
```

```
  <dWmtIn>200000</dWmtIn>
  <dWmtOut>200000</dWmtOut>
  <dIptvIn>200000</dIptvIn>
  <dIptvOut>200000</dIptvOut>
  <mWmtIn>200000</mWmtIn>
  <mWmtOut>200000</mWmtOut>
  <mIptvIn>200000</mIptvIn>
  <mIptvOut>200000</mIptvOut>
</appBw>
```

Response XML: None

Windows Media Streaming General Settings

This API provides parity for the following CDSM GUI page.

Devices>Devices>{SE}>Application Control>Windows Media Streaming>General Settings

Table 2-11 Windows Media Streaming General Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/wmtGenSetting	GET	Get Windows Media Streaming General Settings for an SE identified by {SE_id}.
	POST	Create Windows Media Streaming General Settings for an SE identified by {SE_id}.
	PUT	Modify Windows Media Streaming General Settings for an SE identified by {SE_id}.
	DELETE	Remove Windows Media Streaming General Settings for an SE identified by {SE_id}.
/api/SEs/{SE_id}/wmtGenSetting/applyDefault /api/groups/{DG_id}/wmtGenSetting/applyDefault	POST	Apply default values to the Windows Media Streaming General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}. Request XML is not required.
/api/SEs/{SE_id}/wmtGenSetting/applyDG/{DG_id}	POST	Apply Device Group's Windows Media Streaming General Settings (identified by {DG_id}) on the Windows Media Streaming General Settings of the SE identified by {SE_id}. Request XML is not required.
/api/groups/{DG_id}/wmtGenSetting/forceSEs	POST	Force Windows Media Streaming General Settings on SEs in the Device Group identified by {DG_id}. Request XML is not required.

Examples

- Get Windows Media Streaming General Settings

  Get Windows Media Streaming General Settings for an SE identified by id 234.

  Request URL: /api/SEs/234/wmtGenSetting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wmtGenSetting uri="/api/SEs/234/wmtGenSetting">
  <System_wmt_disallowClientProtocols_http>false</System_wmt_disallowClientProtocols_h
   ttp>
  <System_wmt_disallowClientProtocols_rtspt>false</System_wmt_disallowClientProtocols_
   rtspt>
  <System_wmt_disallowClientProtocols_rtspu>false</System_wmt_disallowClientProtocols_
   rtspu>
  <System_wmt_maxConnectionsEnable>false</System_wmt_maxConnectionsEnable>
  <System_wmt_maxConnections></System_wmt_maxConnections>
  <System_wmt_multicastTimeToLive>5</System_wmt_multicastTimeToLive>
  <System_wmt_enforceMaxBitrate>false</System_wmt_enforceMaxBitrate>
  <System_wmt_maxBitrate></System_wmt_maxBitrate>
  <System_wmt_enforceIncomingMaxBitrate>false</System_wmt_enforceIncomingMaxBitrate>
  <System_wmt_maxIncomingBitrate></System_wmt_maxIncomingBitrate>
  <System_wmt_proxyMaxObjectSize>25600</System_wmt_proxyMaxObjectSize>
  <System_wmt_serviceEnable>true</System_wmt_serviceEnable>
  <System_wmt_proxyOutgoingHttpEnable>false</System_wmt_proxyOutgoingHttpEnable>
  <System_wmt_proxyOutgoingHttpHost_hostname></System_wmt_proxyOutgoingHttpHost_hostna
   me>
  <System_wmt_proxyOutgoingHttpHost_port></System_wmt_proxyOutgoingHttpHost_port>
  <System_wmt_proxyOutgoingRtspEnable>false</System_wmt_proxyOutgoingRtspEnable>
  <System_wmt_proxyOutgoingRtspHost_hostname></System_wmt_proxyOutgoingRtspHost_hostna
   me>
  <System_wmt_proxyOutgoingRtspHost_port></System_wmt_proxyOutgoingRtspHost_port>
  <System_wmt_fastLiveSplitEnable>true</System_wmt_fastLiveSplitEnable>
  <System_wmt_fastProxyCacheEnable>true</System_wmt_fastProxyCacheEnable>
  <System_wmt_vodEnable>true</System_wmt_vodEnable>
  <System_wmt_mmsAllowedExtnsEnable>false</System_wmt_mmsAllowedExtnsEnable>
  <System_wmt_mmsAllowedExtns>asf none nsc wma wmv nsclog</System_wmt_mmsAllowedExtns>
  <System_wmt_fastStartEnable>true</System_wmt_fastStartEnable>
  <System_wmt_fastStartMaxBandwidth>3500</System_wmt_fastStartMaxBandwidth>
  <System_wmt_fastCacheEnable>true</System_wmt_fastCacheEnable>
  <System_wmt_fastCacheMaxDeliveryRate>5</System_wmt_fastCacheMaxDeliveryRate>
  <System_wmt_advancedClientIdleTimeout>60</System_wmt_advancedClientIdleTimeout>
  <System_wmt_advancedClientMaxPacketSize>1500</System_wmt_advancedClientMaxPacketSize
   >
  <System_wmt_advancedServerLogForwardingEnable>true</System_wmt_advancedServerLogForw
   ardingEnable>
  <System_wmt_advancedServer_timeout>65535</System_wmt_advancedServer_timeout>
  <System_wmt_cacheEnable>true</System_wmt_cacheEnable>
  <System_wmt_cacheAgeMultiplier>30</System_wmt_cacheAgeMultiplier>
  <System_wmt_cacheMaxTtl_time>1</System_wmt_cacheMaxTtl_time>
  <System_wmt_cacheMaxTtl_scale>days</System_wmt_cacheMaxTtl_scale>
  <System_wmt_cacheMinTtl>60</System_wmt_cacheMinTtl>
  <System_wmt_cacheReEvalRequest>false</System_wmt_cacheReEvalRequest>
</wmtGenSetting>
```

Windows Media Streaming Bypass List

This API provides parity for the following CDSM GUI page.

Devices>Devices>{SE}>Application Control>Windows Media Streaming>Bypass List

Table 2-12 Windows Media Streaming Bypass List API Calls

Resource URL	Method	Function Description
/api/SEs/{id}/wmtBypassList	GET	Get Windows Media BW Incoming Bypass List for an SE identified by {id}.
	POST	Create Windows Media BW Incoming Bypass List for an SE identified by {id}.
	PUT	Modify Windows Media BW Incoming Bypass List for an SE identified by {id}.
	DELETE	Remove Windows Media BW Incoming Bypass List for an SE identified by {id}.

Examples

- Get Windows Media Streaming Bypass List

  Get Windows Media Streaming Bypass List for an SE identified by id 234.

  Request URL: /api/SEs/234/wmtBypassList

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wmtBypassList uri="/api/SEs/234/wmtBypassList">
  <System_LocalContentServices_wmtBwIncomingBypassList>
    10.73.43.123 10.73.41.234
  </System_LocalContentServices_wmtBwIncomingBypassList>
</wmtBypassList>
```

- Create Windows Media Streaming Bypass List

  Create Windows Media Streaming Bypass List for an SE with id 234. The IP address that 10.73.43.123 and 10.74.41.234 will be added in bypass list.

  Request URL: /api/SEs/234/wmtBypassList

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wmtBypassList>
  <System_LocalContentServices_wmtBwIncomingBypassList>
    10.73.43.123 10.73.41.234
  </System_LocalContentServices_wmtBwIncomingBypassList>
</wmtBypassList>
```

  Response XML: None

- Modify Windows Media Streaming Bypass List

  Modify Windows Media Streaming Bypass List for an SE with id 234. Change the bypass list with 10.73.43.123.

  Request URL: /api/SEs/234/wmtBypassList

Request Method: PUT

Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wmtBypassList>
  <System_LocalContentServices_wmtBwIncomingBypassList>
    10.73.43.123
  </System_LocalContentServices_wmtBwIncomingBypassList>
</wmtBypassList>
```

Response XML: None

- Remove Windows Media BW Incoming Bypass List

  Remove Windows Media Streaming Bypass List. That means empty the bypass list.

  Request URL: /api/SEs/234/wmtBypassList

  Request Method: DELETE

  Request XML: None

  Response XML: None

Movie Streamer General Settings

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>Movie Streamer>General Settings

Table 2-13 Movie Streamer General Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/msGenSetting /api/groups/{DG_id}/msGenSetting	GET	Get the Movie Streamer General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Movie Streamer General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Movie Streamer General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify Movie Streamer General Settings

  Modify Movie Streamer of an SE identified by id 204.

  Request URL: /api/SEs/204/msGenSetting

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<msGenSetting>
  <System_iptv_maxConnectionsEnable>false</System_iptv_maxConnectionsEnable>
  <System_iptv_maxConnections></System_iptv_maxConnections>
```

```
  <System_iptv_enforceMaxBitrate>false</System_iptv_enforceMaxBitrate>
  <System_iptv_maxBitrate></System_iptv_maxBitrate>
  <System_iptv_enforceIncomingMaxBitrate>false</System_iptv_enforceIncomingMaxBitrate>
  <System_iptv_maxIncomingBitrate></System_iptv_maxIncomingBitrate>
  <System_darwin_license_enable>false</System_darwin_license_enable>
  <System_iptv_proxyOutgoingRtspHost_hostname>0.0.0.0</System_iptv_proxyOutgoingRtspHo
   st_hostname>
  <System_iptv_proxyOutgoingRtspHost_port>554</System_iptv_proxyOutgoingRtspHost_port>
  <System_iptv_vodEnable>true</System_iptv_vodEnable>
  <System_iptv_advancedClientIdleTimeout>300</System_iptv_advancedClientIdleTimeout>
  <System_iptv_advancedClientRtpTimeout>180</System_iptv_advancedClientRtpTimeout>
  <System_iptv_cacheEnable>true</System_iptv_cacheEnable>
  <System_iptv_cacheAgeMultiplier>30</System_iptv_cacheAgeMultiplier>
  <System_iptv_cacheMaxTtl_time>1</System_iptv_cacheMaxTtl_time>
  <System_iptv_cacheMaxTtl_scale>days</System_iptv_cacheMaxTtl_scale>
  <System_iptv_cacheReEvalRequest>true</System_iptv_cacheReEvalRequest>
</msGenSetting>
```

Response XML: None

RTSP Advanced Settings

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>RTSP Advanced Settings

Table 2-14 RTSP Advanced Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/rtspGateway /api/groups/{DG_id}/rtspGateway	GET	Get the RTSP Advanced Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the RTSP Advanced Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the RTSP Advanced Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify RTSP Advanced Settings

  Modify RTSP Advanced Settings of an SE identified by id 204.

  Request URL: /api/SEs/204/rtspGateway

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rtspGateway>
    <System_rtsppproxy_maxRequestsPerSecond>40</System_rtsppproxy_maxRequestsPerSecond>
    <System_rtsppproxy_maxInitialSetupDelay>10</System_rtsppproxy_maxInitialSetupDelay>
</rtspGateway>
```

  Response XML: None

FMS Admin Allow Hosts

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>Flash Media Streaming>FMS Admin Allow Hosts

Table 2-15 FMS Admin Allow Hosts API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/fmsAllowHosts /api/groups/{DG_id}/fmsAllowHosts	GET	Get the FMS Admin Allow Hosts settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the FMS Admin Allow Hosts settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the FMS Admin Allow Hosts settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify FMS Admin Allow Hosts

  Modify FMS Admin Allow Hosts of an SE identified by id 204

  Request URL: /api/SEs/204/fmsAllowHosts

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fmsAllowHosts>
    <System_fmsAdminIp_enable>false</System_fmsAdminIp_enable>
    <System_fmsAdminIp_ipList></System_fmsAdminIp_ipList>
</fmsAllowHosts>
```

  Response XML: None

FMS General Settings

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>Flash Media Streaming>General Settings

Table 2-16 FMS General Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/fmsGenSetting /api/groups/{DG_id}/fmsGenSetting	GET	Get the FMS General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the FMS General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the FMS General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify FMS General Settings

  Modify FMS General Settings of an SE by id 204.

  Request URL: /api/SEs/204/fmsGenSetting

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fmsGenSetting>
    <System_fms_serviceEnable>false</System_fms_serviceEnable>
    <System_fms_maxSessions>200</System_fms_maxSessions>
    <System_fms_maxBandwidth>200000</System_fms_maxBandwidth>
</fmsGenSetting>
```

  Response XML: None

FMS Service Monitoring

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>Flash Media Streaming>Service Monitoring

Table 2-17 FMS Service Monitoring API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/fmsServMonitor /api/groups/{DG_id}/fmsServMonitor	GET	Get the FMS Service Monitoring settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the FMS Service Monitoring settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the FMS Service Monitoring settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify FMS Service Monitoring

  Modify FMS Service Monitoring of an SE identified by id 204.

  Request URL: /api/SEs/204/fmsServMonitor

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<fmsServMonitor>
    <System_fms_monitoring_enable>true</System_fms_monitoring_enable>
</fmsServMonitor>
```

  Response XML: None

HTTP Cache Freshness

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Application Control>Web>HTTP>HTTP Cache Freshness

Table 2-18 HTTP Cache Freshness API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/httpCacheFreshness /api/groups/{DG_id}/httpCacheFreshness	GET	Get the HTTP Cache Freshness settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the HTTP Cache Freshness settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the HTTP Cache Freshness settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Modify HTTP Cache Freshness

  Modify HTTP Cache Freshness of an SE identified by id 204.

  Request URL: /api/SEs/204/httpCacheFreshness

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<httpCacheFreshness>
    <System_http_freshnessEnable>false</System_http_freshnessEnable>
    <System_http_ageMultiplier>30</System_http_ageMultiplier>
    <System_http_maxTtl_scale>days</System_http_maxTtl_scale>
    <System_http_maxTtl_object>61</System_http_maxTtl_object>
    <System_http_minTtl>60</System_http_minTtl>
</httpCacheFreshness>
```

Response XML: None

General Settings

Content Management

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Content Management

Table 2-19 Content Management API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/contentMgmt /api/groups/{DG_id}/contentMgmt	GET	Get the Content Management settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the Content Management settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	POST	Create the Content Management settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the Content Management settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
/api/SEs/{SE_id}/contentMgmt/applyDefault /api/groups/{DG_id}/contentMgmt/applyDefault	POST	Apply default values to the Content Management settings for an SE or a Device Group identified by {SE_id} or {DG_id}. Request XML is not required.
/api/SEs/{SE_id}/contentMgmt/applyDG/{DG_id}	POST	Apply Device Group's Content Management settings (identified by {DG_id}) on the Content Management settings of the SE identified by {SE_id}. Request XML is not required.
/api/groups/{DG_id}/contentMgmt/forceSEs	POST	Force Content Management settings on SEs in the Device Group identified by {DG_id}. Request XML is not required.

Examples

- Get Content Management

  Get Content Management of an SE identified by id 234.

  Request URL: /api/SEs/234/contentMgmt

Request Method: GET

Request XML: None

Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentMgmt uri="/api/SEs/234/contentMgmt">
  <System_contentMgmt_cacheMaxEntries>20000000</System_contentMgmt_cacheMaxEntries>
  <System_contentMgmt_evictionprotection_size></System_contentMgmt_evictionprotection_
   size>
  <System_contentMgmt_evictionprotection_duration></System_contentMgmt_evictionprotect
   ion_duration>
  <System_contentMgmt_smallFileEvictionProtection_size></System_contentMgmt_smallFileE
   victionProtection_size>
  <System_contentMgmt_smallFileEvictionProtection_duration></System_contentMgmt_smallF
   ileEvictionProtection_duration>
  <System_contentMgmt_evictionsize>large</System_contentMgmt_evictionsize>
  <System_contentMgmt_hitcntDecayHalfLife>14</System_contentMgmt_hitcntDecayHalfLife>
  <System_contentMgmt_diskbktFailThreshold>30</System_contentMgmt_diskbktFailThreshold
   >
  <System_contentMgmt_slowscan_starttime>00:00</System_contentMgmt_slowscan_starttime>
  <System_contentMgmt_transactionLog_disable>false</System_contentMgmt_transactionLog_
   disable>
</contentMgmt>
```

- Modify Content Management

  Modify Content Management of an SE identified by id 234.

  Request URL: /api/SEs/234/contentMgmt

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentMgmt>
  <System_contentMgmt_cacheMaxEntries>20000000</System_contentMgmt_cacheMaxEntries>
  <System_contentMgmt_evictionsize>1</System_contentMgmt_evictionsize>
  <System_contentMgmt_evictionprotection>false</System_contentMgmt_evictionprotection>
  <System_contentMgmt_evictionprotection_size>0</System_contentMgmt_evictionprotection
   _size>
  <System_contentMgmt_evictionprotection_duration>0</System_contentMgmt_evictionprotec
   tion_duration>
  <System_contentMgmt_hitcntDecayHalfLife>14</System_contentMgmt_hitcntDecayHalfLife>
  <System_contentMgmt_diskbktFailThreshold>30</System_contentMgmt_diskbktFailThreshold
   >
  <System_contentMgmt_slowscan_starttime_primary>00:00</System_contentMgmt_slowscan_st
   arttime_primary>
  <System_contentMgmt_slowscan_starttime_secondary></System_contentMgmt_slowscan_start
   time_secondary>
</contentMgmt>
```

  Response XML: None

Login Access Control

Login Authentication

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Login Access Control>Login Authentication

Table 2-20 Login Authentication API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/loginAuth /api/groups/{DG_id}/loginAuth	GET	Get Login Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Login Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Login Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get Login Authentication settings

  Get Login Authentication settings of an SE identified by id 234

  Request URL: /api/SEs/234/loginAuth

  Request Method: GET

  Request XML: None

  Response XML: None

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loginAuth id="234" uri="/api/SEs/234/loginAuth">
  <Name></Name>
  <Description></Description>
  <LoginEnable>false</LoginEnable>
  <LoginConfigFailoverServerUnreach>false</LoginConfigFailoverServerUnreach>
  <EnablePassword></EnablePassword>
  <LoginPrimary>1</LoginPrimary>
  <LoginSecondary>0</LoginSecondary>
  <LoginTertiary>0</LoginTertiary>
  <EnablePrimary>0</EnablePrimary>
  <EnableSecondary>0</EnableSecondary>
  <EnableTertiary>0</EnableTertiary>
</loginAuth>
```

- Modify Login Authentication settings

  Modify Login Authentication settings of an SE identified by id 234

  Request URL: /api/SEs/234/loginAuth

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<loginAuth>
  <Name></Name>
  <Description></Description>
  <LoginEnable>True</LoginEnable>
  <LoginConfigFailoverServerUnreach>false</LoginConfigFailoverServerUnreach>
  <EnablePassword></EnablePassword>
  <!-- Value for servers
       Local = 1, Radius - 2, Tacacs = 3, Do Not Set = 0 -->
```

```
  <LoginPrimary>1</LoginPrimary>
  <LoginSecondary>0</LoginSecondary>
  <LoginTertiary>0</LoginTertiary>
  <EnablePrimary>0</EnablePrimary>
  <EnableSecondary>0</EnableSecondary>
  <EnableTertiary>0</EnableTertiary>
</loginAuth>
```

Response XML: None

Exec Authentication

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Login Access Control>Exec Authentication

Table 2-21 Exec Authentication API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/execAuth /api/groups/{DG_id}/execAuth	GET	Get Exec Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Exec Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Exec Authentication settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

SSH

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Login Access Control>SSH

Table 2-22 SSH API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/ssh /api/groups/{DG_id}/ssh	GET	Get SSH settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify SSH settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Delete SSH settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Telnet

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Login Access Control>Telnet

Table 2-23 Telnet API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/telnet /api/groups/{DG_id}/telnet	GET	Get Telnet settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Telnet settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Telnet settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Message of the Day

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups >{CDM|SR|SE|DG}>General Settings>Login Access Control>Message of the Day

Table 2-24 Message of the Day API Calls

Resource URL	Method	Function Description
/api/SRs/{SR_id}/motd /api/SEs/{SE_id}/motd /api/groups/{DG_id}/motd	GET	Get Message of the Day settings for a device or a Device Group identified by {id}.
	POST	Create Message of the Day settings for a device or a Device Group identified by {id}.
	PUT	Modify Message of the Day settings for a device or a Device Group identified by {id}.
	DELETE	Remove Message of the Day settings for a device or a Device Group identified by {id}.
/api/SEs/{SE_id}/motd/applyDefault /api/groups/{DG_id}/motd/applyDefault	POST	Apply default values to the Message of the Day settings for an SE or a Device Group identified by {SE_id} or {DG_id}. Request XML is not required.

Table 2-24 Message of the Day API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/motd/applyDG/{DG_id}	POST	Apply Device Group's Message of the Day settings (identified by {DG_Id}) on the Message of the Day settings of the SE identified by {SE_id}. Request XML is not required.
/api/groups/{DG_id}/motd/forceSEs	POST	Force Message of the Day settings on SEs in the group identified by {DG_id}. Request XML is not required.

Examples

- Get Message of the Day settings

  Modify Message of the Day settings of an SE identified by id 234

  Request URL: /api/SEs/234/motd

  Request Method: GET

  Request XML: None

  Response XML: None

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<motd uri="/api/SEs/234/motd">
  <System_motd_enable>false</System_motd_enable>
  <System_motd_motdMessage></System_motd_motdMessage>
  <System_motd_loginMessage></System_motd_loginMessage>
  <System_motd_execMessage></System_motd_execMessage>
  <System_motd_login_banner_enable>true</System_motd_login_banner_enable>
</motd>
```

CLI Session Time

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE| DG}>General Settings>Login Access Control>CLI Session Time

Table 2-25 CLI Session Time API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/cliSessionTime /api/groups/{DG_id}/cliSessionTime	GET	Get CLI Session Time settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify CLI Session Time settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove CLI Session Time settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Usernames

This API provides parity for the following CDSM GUI page.

Devices>Devices>Device Groups>{SE|SR|DG}>General Settings>Login Access Control> Users>Usernames

Table 2-26 Usernames API calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/localUsers /api/SRs/{SR_id}/localUsers /api/groups/{DG_id}/localUsers	GET	Get all Usernames for a device or a Device Group identified by {id}.
/api/SEs/{SE_id}/localUser/{USER_id} /api/SRs/{SR_id}/localUser/{USER_id} /api/groups/{DG_id}/localUser/{USER_id}	GET	Get a Usernames with {USER_id} for a device or a Device Group identified by {id}.
	DELETE	Remove a Usernames with {USER_id} identified by {USER_id} for a device or a Device Group identified by {id}.

Examples

- Get Usernames

  Get a Usernames with USER_id 303 from SE identified by SE_id 234.

  Request URL: /api/SEs/234/localUsers/303

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<localUsers uri="/api/SEs/234/localUsers">
  <localUsers-row id="303" uri="/api/SEs/234/localUsers/303">
    <username>testUser6</username>
    <password>$5$cRgyTXbh$9Y0H1g2rZwp36M9K.TTAEYzdJVG.EwDcdfTkSqV8dU2</password>
    <cifsPassword></cifsPassword>
    <privilege>0</privilege>
  </localUsers-row>
</localUsers>
```

- Remove Usernames

  Remove a Username with USER_id 303 from SE identified by SE_id 234.

  Request URL: /api/SEs/234/localUsers/303

  Request Method: DELETE

  Request XML: None

  Response XML: None

Authentication

RADIUS Server

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Authentication>RADIUS Server

Table 2-27 RADIUS Server settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/radius /api/groups/{DG_id}/radius	GET	Get RADIUS Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify RADIUS Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove RADIUS Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get REDIUS Server settings

  Get RADIUS Server settings of an SE identified by id 234.

  Request URL: /api/SEs/234/radius

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<radius id="234" uri="/api/SEs/234/radius">
  <Name></Name>
```

```
  <Description></Description>
  <RadiusContainerEnable>true</RadiusContainerEnable>
  <RadiusContainerRedirectEnable>true</RadiusContainerRedirectEnable>
  <RadiusContainerRedirectMsg0>two</RadiusContainerRedirectMsg0>
  <RadiusContainerRedirectLocation0>http://two.se</RadiusContainerRedirectLocation0>
  <RadiusContainerEncryptKey>1234</RadiusContainerEncryptKey>
  <RadiusContainerTimeout>5</RadiusContainerTimeout>
  <RadiusContainerRetransmit>2</RadiusContainerRetransmit>
  <RadiusContainerServer1>1.1.1.1</RadiusContainerServer1>
  <RadiusContainerPort1>1645</RadiusContainerPort1>
  <RadiusContainerPort2>1645</RadiusContainerPort2>
  <RadiusContainerPort3>1645</RadiusContainerPort3>
  <RadiusContainerPort4>1645</RadiusContainerPort4>
  <RadiusContainerPort5>1645</RadiusContainerPort5>
</radius>
```

TACACS+ Server

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Authentication>TACACS+ Server

Table 2-28 TACACS+ Server settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/tacacs /api/groups/{DG_id}/tacacs	GET	Get TACACS+ Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify TACACS+ Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove TACACS+ Server settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get TACACS+ Server settings

  Get TACACS+ Server settings of an SE identified by id 234.

  Request URL: /api/SEs/234/tacacs

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tacacs id="234" uri="/api/SEs/234/tacacs">
  <Name></Name>
  <Description></Description>
  <TacacsContainerEnable>true</TacacsContainerEnable>
  <TacacsContainerUseAsciiPassword>false</TacacsContainerUseAsciiPassword>
  <TacacsContainerTimeout>5</TacacsContainerTimeout>
  <TacacsContainerRetransmit>2</TacacsContainerRetransmit>
  <TacacsContainerPrimaryServer>1.1.1.1</TacacsContainerPrimaryServer>
</tacacs>
```

AAA Accounting

This API provides parity for the following CDSM GUI page.

Devices>Devices>Device Groups>{SE|DG}>General Settings>Authentication>AAA Accounting

Table 2-29 AAA Accounting settings API calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/aaaAcounting /api/groups/{DG_id}/aaaAcounting	GET	Get AAA Accounting settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	POST	Create AAA Accounting settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify AAA Accounting settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove AAA Accounting settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get AAA Accounting settings

  Get all AAA Accounting Settings for an SE identified by id 234.

  Request URL: /api/SEs/234/aaaAcounting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<aaaAcounting uri="/api/SEs/234/aaaAcounting">
  <System_aaa_tacacs_systemOptions>
    start-stop
  </System_aaa_tacacs_systemOptions>
  <System_aaa_tacacs_execOptions>
    stop-only
  </System_aaa_tacacs_execOptions>
  <System_aaa_tacacs_normalUserCommandsOptions>
    start-stop
  </System_aaa_tacacs_normalUserCommandsOptions>
  <System_aaa_tacacs_adminUserCommandsOptions>
    start-stop
  </System_aaa_tacacs_adminUserCommandsOptions>
</aaaAcounting>
```

- Create AAA Accounting settings

  Create AAA Accounting Settings for an SE with id 234. The TACACS+ server need to be enabled before creating it.

Request URL: /api/SEs/234/aaaAcounting

Request Method: POST

Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<aaaAcounting>
    <System_aaa_tacacs_systemOptions>
        start-stop
    </System_aaa_tacacs_systemOptions>
    <System_aaa_tacacs_execOptions>
        stop-only
    </System_aaa_tacacs_execOptions>
    <System_aaa_tacacs_normalUserCommandsOptions>
        start-stop
    </System_aaa_tacacs_normalUserCommandsOptions>
    <System_aaa_tacacs_adminUserCommandsOptions>
        start-stop
    </System_aaa_tacacs_adminUserCommandsOptions>
</aaaAcounting>
```

Response XML: None

- Modify AAA Accounting settings

  Modify AAA Accounting settings for an SE identified by id 234. Change all values of field to stop-only.

  Request URL: /api/SEs/234/aaaAcounting

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<aaaAcounting>
  <System_aaa_tacacs_systemOptions>
    stop-only
  </System_aaa_tacacs_systemOptions>
  <System_aaa_tacacs_execOptions>
    stop-only
  </System_aaa_tacacs_execOptions>
  <System_aaa_tacacs_normalUserCommandsOptions>
    stop-only
  </System_aaa_tacacs_normalUserCommandsOptions>
  <System_aaa_tacacs_adminUserCommandsOptions>
    stop-only
  </System_aaa_tacacs_adminUserCommandsOptions>
</aaaAcounting>
```

  Response XML: None

- Remove AAA Accounting Settings

  Remove configurations of AAA Accounting Settings. That means set all fields to “Do Not Set”.

  Request URL: /api/SEs/234/aaaAcounting

  Request Method: DELETE

  Request XML: None

  Response XML: None

Enable Access Control List

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Authentication>Access Control List>Enable Access Control List

Table 2-30 Enable Access Control List settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/enableAcl /api/SRs/{DG_id}/enableAcl	GET	Get the status of Enable Access Control List for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify the status of Enable Access Control List for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove the status of Enable Access Control List for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get Enable Access Control List

  Get the status of Enable Access Control List identified by id 234.

  Request URL: /api/SEs/234/enableAcl

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enableAcl uri="/api/SEs/234/enableAcl">
  <System_accesslist_enable>false</System_accesslist_enable>
</enableAcl>
```

Configure Access Control List

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Authentication>Access Control List>Configure Access Control List

Table 2-31 ACL API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/acl /api/SRs/{DG_id}/acl	GET	Get Access Control List settings for an SE or a Device Group identified by {SE_id}, or {DG_id}.
	PUT	Modify Access Control List settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Access Control List settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get Access Control List settings

  Get Access Control List settings of an SE identified by id 234.

  Request URL: /api/SEs/234/acl

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<acl uri="/api/SEs/234/acl">
  <acl-row id="587" uri="/api/SEs/234/acl/587">
    <action>0</action>
    <groupName>1</groupName>
    <position>1</position>
  </acl-row>
  <acl-row id="588" uri="/api/SEs/234/acl/588">
    <action>0</action>
    <groupName>&amp;quot;group 3&amp;quot;</groupName>
    <position>2</position>
  </acl-row>
  <acl-row id="586" uri="/api/SEs/234/acl/586">
    <action>0</action>
    <groupName>&amp;quot;group 1&amp;quot;</groupName>
    <position>3</position>
  </acl-row>
</acl>
```

Storage

Disk Error Handing

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Storage>Disk Error Handing

Table 2-32 Disk Error Handing settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/diskError /api/groups/{DG_id}/diskError	GET	Get Disk Error Handling settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Disk Error Handling settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Disk Error Handling settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get Disk Error Handing settings

  Get Disk Error Handing settings of an SE identified by id 234.

  Request URL: /api/SEs/234/diskError

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<diskError uri="/api/SEs/234/diskError">
    <System_diskErrorHandling_enable>false</System_diskErrorHandling_enable>

<System_diskErrorHandling_reloadEnable>false</System_diskErrorHandling_reloadEnable>

<System_diskErrorHandling_thresholdEnable>false</System_diskErrorHandling_thresholdEna
     ble>

<System_diskErrorHandling_badsectors_threshold>15</System_diskErrorHandling_badsectors
     _theshold>

<System_diskErrorHandling_errors_threshold>500</System_diskErrorHandling_errors_thresh
     old>
</diskError>
```

Network

FTP

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>FTP

Table 2-33 FTP settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/ftp /api/groups/{DG_id}/ftp	GET	Get FTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify FTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove FTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get FTP settings

  Get FTP settings of an SE identified by id 234.

  Request URL: /api/SEs/234/ftp

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ftp uri="/api/SEs/234/ftp">
  <System_ftpInetd_enable>true</System_ftpInetd_enable>
</ftp>
```

DNS

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>DNS

Table 2-34 DNS settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/dns /api/groups/{DG_id}/dns	GET	Get DNS settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify DNS settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove DNS settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get DNS settings

  Get DNS settings of an SE identified by id 234.

  Request URL: /api/SEs/234/dns

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dns uri="/api/SEs/234/dns">
  <System_dns_enable>false</System_dns_enable>
  <System_dns_nameServer></System_dns_nameServer>
  <System_dns_domainName></System_dns_domainName>
</dns>
```

RCP

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>RCP

Table 2-35 RCP settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/rcp /api/groups/{DG_id}/rcp	GET	Get RCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify RCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove RCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get RCP settings

  Get RCP settings of an SE identified by id 234.

  Request URL: /api/SEs/234/rcp

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rcp uri="/api/SEs/234/rcp">
  <System_inetdRcp_enable>false</System_inetdRcp_enable>
</rcp>
```

NTP

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>NTP

Table 2-36 NTP settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/ntp /api/groups/{DG_id}/ntp	GET	Get NTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify NTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove NTP settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get NTP settings

  Get NTP settings of an SE identified by id 234.

  Request URL: /api/SEs/234/ntp

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ntp uri="/api/SEs/234/ntp">
  <System_ntp_enable>false</System_ntp_enable>
  <System_ntp_server></System_ntp_server>
</ntp>
```

TCP

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>TCP

Table 2-37 TCP settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/tcp /api/groups/{DG_id}/tcp	GET	Get TCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify TCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove TCP settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get TCP settings

  Get TCP settings of an SE identified by id 234.

  Request URL: /api/SEs/234/tcp

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<tcp uri="/api/SEs/234/tcp">
  <System_tcp_timeStamp>true</System_tcp_timeStamp>
  <System_tcp_twRecycle>true</System_tcp_twRecycle>
  <System_tcp_twReuse>true</System_tcp_twReuse>
</tcp>
```

Time Zone

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Group>{SE|DG}>Network>Time Zone

Table 2-38 Time Zone Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/timeZone /api/groups/{DG_id}/timeZone	GET	Get Time Zone settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
/api/SEs/{SE_id}/timeZone/{id} /api/groups/{DG_id}/timeZone/{id}	PUT	Modify Time Zone settings with {id} for an SE or a Device Group identified by {SE_id} or {DG_id}.

Table 2-38 Time Zone Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/timeZone /api/groups/{DG_id}/timeZone	POST	Create Time Zone settings for an SE or a Device Group identified by {SE_id} or {DG_id}
/api/SEs/{SE_id}/timeZone/{id} /api/groups/{DG_id}/timeZone/{id}	DELETE	Remove Time Zone settings with {id} for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Create Customized Time Zone settings

  Create Customized Time Zone settings of an SE identified by id 234.

  Request URL: /api/SEs/234/timeZone

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<timeZone>
  <timeZoneName>newTimeZone</timeZoneName>
  <timeZoneUtcOffset>-150</timeZoneUtcOffset>
  <summerTimeType>2</summerTimeType>
  <summerTimeOffset>12</summerTimeOffset>
  <summerTimeStartTime>10/16/2013</summerTimeStartTime>
  <summerTimeEndTime>11/16/2013</summerTimeEndTime>
  <recStartDay>1</recStartDay>
  <recStartWeek>1</recStartWeek>
  <recStartMonth>1</recStartMonth>
  <recStartMin>1</recStartMin>
  <recEndDay>2</recEndDay>
  <recEndWeek>2</recEndWeek>
  <recEndMonth>2</recEndMonth>
  <recEndMin>2</recEndMin>
  <isCustomTz>custom</isCustomTz>
</timeZone >
```

  Response XML: None

- Modify Customized Time Zone settings

  Modify Customized Time Zone settings with id 572 for an SE identified by id 234.

  Request URL: /api/SEs/234/timeZone/572

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<timeZone>
  <timeZoneName>newTimeZone</timeZoneName>
  <timeZoneUtcOffset>-150</timeZoneUtcOffset><!-Value is given in hours -->
  <!-- Summer Time Type
  No Value for no summer time
  2 = Absolute Dates
  3 = Recurring Dates
  -->
  <summerTimeType>2</summerTimeType>
  <summerTimeOffset>12</summerTimeOffset>
```

```
  <!-- Date format is MM/DD/YYYY -->
  <summerTimeStartTime>10/16/2013</summerTimeStartTime>
  <summerTimeEndTime>11/16/2013</summerTimeEndTime>
  <!-- Day value will be 1=MON,2=TUE etc. till 7=SUN -->
  <recStartDay></recStartDay>
  <!-- Week value will be 1=1st Week,2=2nd Week etc. till 4=last Week -->
  <recStartWeek></recStartWeek>
  <!-- Month value must be specified in number -->
  <recStartMonth></recStartMonth>
  <recStartMin></recStartMin>
  <recEndDay></recEndDay>
  <recEndWeek></recEndWeek>
  <recEndMonth></recEndMonth>
  <recEndMin></recEndMin>
  <!-- For isCustomTz value must be "standard" or "custom" -->
  <isCustomTz>custom</isCustomTz>
</timeZone>
```

Response XML: None

- Create Standard Time Zone settings

  Create Standard Time Zone settings of an SE identified by id 234.

  Request URL: /api/SEs/234/timeZone

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<timeZone>
    <timeZoneName>MST</timeZoneName>
    <isCustomTz>standard</isCustomTz>
</timeZone >
```

  Response XML: None

- Modify Standard Time Zone settings

  Modify Standard Time Zone Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/timeZone/572

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Data Model for modification of Standard TimeZone -->
<timeZone>
<!-- For subtimezone the value must be like "Europe/Amsterdam" -->
<timeZoneName>MST</timeZoneName>
<!-- For isCustomTz value must be "standard" or "custom" -->
<isCustomTz>standard</isCustomTz>
</timeZone >
```

  Response XML: None

Port Channel Settings

This API provides parity for the following CDSM GUI page:

Devices> Devices>Device Groups>{SE|DG}>Network>Port Channel Settings

Table 2-39 Port Channel Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/portChannelSetting /api/groups/{DG_id}/portChannelSetting	GET	Get Port Channel Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Port Channel Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Port Channel Settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get Port Channel Settings

  Get Port Channel Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/portChannelSetting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<portChannelSetting uri="/api/SEs/234/portChannelSetting">
  <System_portChannel_loadBalanceMethod>dst-ip</System_portChannel_loadBalanceMethod>
</portChannelSetting>
```

IP General Settings

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>IP General Settings

Table 2-40 IP General Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/ipGenSetting /api/groups/{DG_id}/ipGenSetting	GET	Get IP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify IP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove IP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get IP General Settings

Get IP General Settings of an SE identified by id 234.

Request URL: /api/SEs/234/ipGenSetting

Request Method: GET

Request XML: None

Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ipGenSetting uri="/api/SEs/234/ipGenSetting">

<System_ipGeneralSettings_pathMtuDiscovery>true</System_ipGeneralSettings_pathMtuDisco
     very>
</ipGenSetting>
```

DSR VIP

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>Network>DSR VIP

Table 2-41 DSR VIP settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/dsrVIP /api/groups/{DG_id}/dsrVIP	GET	Get DSR VIP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify DSR VIP settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove DSR VIP settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get DSR VIP settings

  Get DSR VIP settings of an SE identified by id 234.

  Request URL: /api/SEs/234/dsrVIP

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dsrVIP uri="/api/SEs/234/dsrVIP">
  <System_dsr_vip0></System_dsr_vip0>
  <System_dsr_vipv6_0></System_dsr_vipv6_0>
  <System_dsr_vip1></System_dsr_vip1>
  <System_dsr_vipv6_1></System_dsr_vipv6_1>
  <System_dsr_vip2></System_dsr_vip2>
  <System_dsr_vipv6_2></System_dsr_vipv6_2>
  <System_dsr_vip3></System_dsr_vip3>
  <System_dsr_vipv6_3></System_dsr_vipv6_3>
</dsrVIP>
```

Notification and Tracking

Alarm Settings

This API provides parity for the following CDSM GUI page:

Devices> Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking> Alarm Settings

Table 2-42 Alarm Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/alarmSetting /api/groups/{DG_id}/alarmSetting	GET	Get Alarm Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Alarm Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Alarm Settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get Alarm Settings

  Get Alarm Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/alarmSetting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<alarmSetting uri="/api/SEs/234/alarmSetting">
  <System_alarmOverloadDetect_enable>true</System_alarmOverloadDetect_enable>
  <System_alarmOverloadDetect_waterMark_raise>10</System_alarmOverloadDetect_waterMark
    _raise>
  <System_alarmOverloadDetect_waterMark_clear>1</System_alarmOverloadDetect_waterMark_
    clear>
  <System_alarmGeneration_adminShutdownIntf>false</System_alarmGeneration_adminShutdow
    nIntf>
</alarmSetting>
```

Service Monitor

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>Service Monitor

Table 2-43 Service Monitor settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/serviceMon /api/groups/{DG_id}/serviceMon	GET	Get Service Monitor settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Service Monitor settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Service Monitor settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get Service Monitor settings

  Get Service Monitor settings of an SE identified by id 234.

  Request URL: /api/SEs/234/serviceMon

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<serviceMon uri="/api/SEs/234/serviceMon">
  <System_serviceMonitor_wmt_enable>true</System_serviceMonitor_wmt_enable>
  <System_serviceMonitor_wmt_threshold>90</System_serviceMonitor_wmt_threshold>
  <System_serviceMonitor_wmt_period>1</System_serviceMonitor_wmt_period>
  <System_serviceMonitor_wmt_numberofsamples>2</System_serviceMonitor_wmt_numberofsamp
   les>
  <System_serviceMonitor_memory_enable>true</System_serviceMonitor_memory_enable>
  <System_serviceMonitor_memory_threshold>80</System_serviceMonitor_memory_threshold>
  <System_serviceMonitor_memory_period>1</System_serviceMonitor_memory_period>
  <System_serviceMonitor_memory_numberofsamples>2</System_serviceMonitor_memory_number
   ofsamples>
  <System_serviceMonitor_kmemory_enable>true</System_serviceMonitor_kmemory_enable>
  <System_serviceMonitor_kmemory_threshold>50</System_serviceMonitor_kmemory_threshold
   >
  <System_serviceMonitor_kmemory_period>1</System_serviceMonitor_kmemory_period>
  <System_serviceMonitor_kmemory_numberofsamples>2</System_serviceMonitor_kmemory_numb
   erofsamples>
  <System_serviceMonitor_cpu_enable>true</System_serviceMonitor_cpu_enable>
  <System_serviceMonitor_cpu_threshold>80</System_serviceMonitor_cpu_threshold>
  <System_serviceMonitor_cpu_period>1</System_serviceMonitor_cpu_period>
  <System_serviceMonitor_cpu_numberofsamples>2</System_serviceMonitor_cpu_numberofsamp
   les>
  <System_serviceMonitor_disk_enable>true</System_serviceMonitor_disk_enable>
  <System_serviceMonitor_disk_threshold>80</System_serviceMonitor_disk_threshold>
  <System_serviceMonitor_disk_period>1</System_serviceMonitor_disk_period>
  <System_serviceMonitor_disk_numberofsamples>2</System_serviceMonitor_disk_numberofsa
   mples>
  <System_serviceMonitor_fms_enable>true</System_serviceMonitor_fms_enable>
  <System_serviceMonitor_fms_threshold>90</System_serviceMonitor_fms_threshold>
  <System_serviceMonitor_fms_period>1</System_serviceMonitor_fms_period>
```

```
  <System_serviceMonitor_fms_numberofsamples>2</System_serviceMonitor_fms_numberofsamp
   les>
  <System_serviceMonitor_moviestreamer_enable>true</System_serviceMonitor_moviestreame
   r_enable>
  <System_serviceMonitor_moviestreamer_threshold>90</System_serviceMonitor_moviestream
   er_threshold>
  <System_serviceMonitor_burstcnt_threshold>1</System_serviceMonitor_burstcnt_threshol
   d>
  <System_serviceMonitor_diskFailurePercentage_threshold>75</System_serviceMonitor_dis
   kFailurePercentage_threshold>
  <System_serviceMonitor_nic_enable>true</System_serviceMonitor_nic_enable>
  <System_serviceMonitor_nic_threshold>90</System_serviceMonitor_nic_threshold>
  <System_serviceMonitor_nic_period>3</System_serviceMonitor_nic_period>
  <System_serviceMonitor_nic_numberofsamples>2</System_serviceMonitor_nic_numberofsamp
   les>
  <System_serviceMonitor_augmentAlarm_enable>false</System_serviceMonitor_augmentAlarm
   _enable>
  <System_serviceMonitor_augmentAlarm_threshold>80</System_serviceMonitor_augmentAlarm
   _threshold>
  <System_serviceMonitor_transactionLog_enable>false</System_serviceMonitor_transactio
   nLog_enable>
</serviceMon>
```

System Monitor

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>System Monitor

Table 2-44 System Monitor settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/sysMon /api/SRs/{SR_id}/sysMon /api/groups/{DG_id}/sysMon	GET	Get System Monitor settings for a device or Device Group identified by {SE_id}, {SR_ID} or {DG_id}.
	PUT	Modify Service Monitor settings for a device or Device Group identified by {SE_id}, {SR_ID} or {DG_id}.
	DELETE	Remove System Monitor settings for a device or Device Group identified by {SE_id}, {SR_ID} or {DG_id}.

Examples

- Get System Monitor settings

  Get System Monitor settings of an SE identified by id 234.

  Request URL: /api/SEs/234/sysMon

  Request Method: GET

  Request XML: None

  Response XML:

```
<<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sysMon uri="/api/SEs/239/sysMon">
```

```
<System_systemMonitor_uninterruptible_sleep_enable>false</System_systemMonitor_uninter
ruptible_sleep_enable>
</sysMon>
```

SNMP General Settings

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>SNMP>General Settings

Table 2-45 SNMP General Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/snmpGenSetting /api/groups/{DG_id}/snmpGenSetting	GET	Get SNMP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify SNMP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove SNMP General Settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get SNMP General Settings

  Get SNMP General Settings of an SE identified by id 234.

  Request URL: /api/SEs/234/snmpGenSetting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<snmpGenSetting uri="/api/SEs/234/snmpGenSetting">
  <snmpGenSetting-row id="595" uri="/api/SEs/234/snmpGenSetting/595">
    <contact></contact>
    <location></location>
    <ceTrapFlag>4</ceTrapFlag>
    <entity></entity>
    <event></event>
    <config></config>
    <authentication>1</authentication>
    <coldStart></coldStart>
    <alarmTrapFlag>5</alarmTrapFlag>
    <enableTraps></enableTraps>
    <mibPersistentEvent></mibPersistentEvent>
    <notifyInform>1</notifyInform>
    <writeMib></writeMib>
    <crConfig></crConfig>
    <cdmConfig></cdmConfig>
    <ceConfig>CeConfig_234</ceConfig>
    <deviceGroup></deviceGroup>
  </snmpGenSetting-row>
</snmpGenSetting>
```

SNMP Asset Tag

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>SNMP>Asset Tag

Table 2-46 SNMP Asset Tag settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/snmpAsset /api/groups/{DG_id}/snmpAsset	GET	Get SNMP Asset Tag settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify SNMP Asset Tag settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove SNMP Asset Tag settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get SNMP Asset Tag settings

  Get Asset Tag settings of an SE identified by id 234.

  Request URL: /api/SEs/234/snmpAsset

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<snmpAsset uri="/api/SEs/234/snmpAsset">
    <System_snmpAsset_assetTagName>tagName</System_snmpAsset_assetTagName>
</snmpAsset>
```

SNMP Contact

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>SNMP>Contact

Table 2-47 SNMP Contact settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/snmpContact /api/groups/{DG_id}/snmpContact	GET	Get SNMP Contact settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify SNMP Contact settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove SNMP Contact settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get SNMP Contact settings

  Get SNMP Contact settings of an SE identified by id 234.

  Request URL: /api/SEs/234/snmpContact

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<snmpContact uri="/api/SEs/234/snmpContact">
    <System_snmpContact_contact>sysContact</System_snmpContact_contact>
    <System_snmpContact_location>Lo_1</System_snmpContact_location>
</snmpContact>
```

System Log

This API provides parity for the following CDSM GUI page:

Devices>Devices>Device Groups>{SE|DG}>General Settings>Notification and Tracking>System Log

Table 2-48 System Log Settings API Calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/syslog /api/groups/{DG_id}/syslog	GET	Get System Log settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify System Log settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove System Log settings for an SE or a Device Group identified by {SE_id} or {DG_id}

Examples

- Get System Log settings

  Get System Log settings of an SE identified by id 234.

  Request URL: /api/SEs/234/syslog

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<syslog uri="/api/SEs/234/syslog">
  <System_syslog_enable>false</System_syslog_enable>
  <System_syslog_consoleEnable>false</System_syslog_consoleEnable>
  <System_syslog_consolePriority>warning</System_syslog_consolePriority>
  <System_syslog_diskEnable>true</System_syslog_diskEnable>
  <System_syslog_diskPriority>notice</System_syslog_diskPriority>
  <System_syslog_fileName>/local1/syslog.txt</System_syslog_fileName>
  <System_syslog_recycle>10000000</System_syslog_recycle>
  <System_syslog_hostEnable>false</System_syslog_hostEnable>
  <System_syslog_host0_hostname></System_syslog_host0_hostname>
  <System_syslog_host0_priority>warning</System_syslog_host0_priority>
  <System_syslog_host0_port>514</System_syslog_host0_port>
  <System_syslog_host0_ratelimit>0</System_syslog_host0_ratelimit>
  <System_syslog_host1_hostname></System_syslog_host1_hostname>
  <System_syslog_host1_priority>warning</System_syslog_host1_priority>
  <System_syslog_host1_port>514</System_syslog_host1_port>
  <System_syslog_host1_ratelimit>0</System_syslog_host1_ratelimit>
  <System_syslog_host2_hostname></System_syslog_host2_hostname>
  <System_syslog_host2_priority>warning</System_syslog_host2_priority>
  <System_syslog_host2_port>514</System_syslog_host2_port>
  <System_syslog_host2_ratelimit>0</System_syslog_host2_ratelimit>
  <System_syslog_host3_hostname></System_syslog_host3_hostname>
  <System_syslog_host3_priority>warning</System_syslog_host3_priority>
  <System_syslog_host3_port>514</System_syslog_host3_port>
  <System_syslog_host3_ratelimit>0</System_syslog_host3_ratelimit>
  <System_syslog_facility></System_syslog_facility>
</syslog>
```

Service Router Settings

This API provides parity for the following CDSM GUI page.

Devices>Devices>Device Groups>{SE|DG}>General Settings>Service Router Settings

Table 2-49 Service Router Settings API calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/srSetting /api/groups/{DG_id}/srSetting	GET	Get Service Router Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	POST	Create Service Router Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Service Router Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Service Router Settings for an SE or a Device Group identified by {SE_id} or {DG_id}.

Examples

- Get Service Router Settings

  Get Service Router Settings for an SE identified by id 234.

  Request URL: /api/SEs/234/srSetting

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<srSetting uri="/api/SEs/234/srSetting">
  <System_keepAlive_interval>
    2
  </System_keepAlive_interval>
  <System_keepAlive_snapshotCounterReportInterval>
    10
  </System_keepAlive_snapshotCounterReportInterval>
</srSetting>
```

- Create Service Router Settings

  Create Service Router Settings for an SE with id 234.

  Request URL: /api/SEs/234/srSetting

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<srSetting>
  <System_keepAlive_interval>
    2
  </System_keepAlive_interval>
  <System_keepAlive_snapshotCounterReportInterval>
    120
  </System_keepAlive_snapshotCounterReportInterval>
```

```
</srSetting>
```

Response XML: None

- Modify Service Router Settings

  Modify Service Router Settings for an SE with id 234. Change the "Snapshot Counter Report Interval" with 80.

  Request URL: /api/SEs/234/srSetting

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<srSetting>
  <System_keepAlive_interval>
    2
  </System_keepAlive_interval>
  <System_keepAlive_snapshotCounterReportInterval>
    80
  </System_keepAlive_snapshotCounterReportInterval>
</srSetting>
```

  Response XML: None

- Remove Service Router Settings

  Remove Service Router Settings. That means configurations will be reset to factory defaults.

  Request URL: /api/SEs/234/srSetting

  Request Method: DELETE

  Request XML: None

  Response XML: None

Cache Router

This API provides parity for the following CDSM GUI page.

Devices>Devices>Device Groups>{SE|DG}>General Settings>Cache Router

Table 2-50 Cache Router settings API calls

Resource URL	Method	Function Description
/api/SEs/{SE_id}/cacheRouter /api/groups/{DG_id}/cacheRouter	GET	Get Cache Router settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	POST	Create Cache Router settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	PUT	Modify Cache Router settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
	DELETE	Remove Cache Router settings for an SE or a Device Group identified by {SE_id} or {DG_id}.
/api/SEs/{SE_id}/cacheRouter/applyDefault /api/groups/{DG_id}/cacheRouter/applyDefault	POST	Reset Cache Router settings for an SE or a Device Group identified by {SE_id} or {DG_id} to Factory Defaults. Request XML is not required.
/api/SEs/{SE_id}/cacheRouter/applyDG/{DG_id}	POST	Set Cache Router settings for an SE identified by {SE_id} to a Device Group with {DG_id} settings. Request XML is not required.
/api/groups/{DG_id}/cacheRouter/forceSEs	POST	Set Cache Router settings for all SEs those are belonging with a Device Group identified by {DG_id}. Request XML is not required.

Examples

- Get Cache Router settings

  Get Cache Router settings for an SE identified by id 234.

  Request URL: /api/SEs/234/cacheRouter

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cacheRouter uri="/api/SEs/234/cacheRouter">
  <System_cacheRouter_timeout>
    222
  </System_cacheRouter_timeout>
</cacheRouter>
```

- Create Cache Router settings

  Create Cache Router settings for an SE with id 234.

Request URL: /api/SEs/234/cacheRouter

Request Method: POST

Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cacheRouter >
  <System_cacheRouter_timeout>
    200
  </System_cacheRouter_timeout>
</cacheRouter>
```

Response XML: None

- Modify Cache Router settings

  Modify Cache Router settings for an SE with id 234. Change the “Liveness Query Timeout” with 200.

  Request URL: /api/SEs/234/cacheRouter

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cacheRouter >
  <System_cacheRouter_timeout>200</System_cacheRouter_timeout>
</cacheRouter>
```

  Response XML: None

- Remove Cache Router settings

  Remove Cache Router settings. That means configurations will be overridden by Device Group if SE belongs to a Device Group in which the configurations have been set, otherwise it will be reset to factory defaults.

  Request URL: /api/SEs/234/cacheRouter

  Request Method: DELETE

  Request XML: None

  Response XML: None

DNS Based Redirection

This API provides parity for the following CDSM GUI page:

Devices>Devices>{SR}>Routing Settings>Request Routing Settings>DNS Base Redirection

Table 2-51 DNS Based Redirection settings API Calls

Resource URL	Method	Function Description
/api/SRs/{SR_id}/dnsRedirection	GET	Get DNS Base Redirection settings of SR identified by {SR_id}.
	PUT	Modify DNS Base Redirection settings of SR identified by {SR_id}.
	POST	Create DNS Base Redirection settings of SR identified by {SR_id}.
	DELETE	Remove DNS Base Redirection settings of SR identified by {SR_id}.

Examples

- Get DNS Base Redirection settings

  Get DNS Base Redirection settings of SR identified by SR id 234.

  Request URL: /api/SRs/234/dnsRedirection

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dnsRedirection uri="SRs/234/dnsRedirection">
    <System_dnsRedirect_basedOnWE_enable>false</System_dnsRedirect_basedOnWE_enable>
    <System_dnsRedirect_domain_all_enable>false</System_dnsRedirect_domain_all_enable>
    <domain_list>
        <domain>aaa.com</domain>
        <domain>bbb.com</domain>
        <domain>ccc.com</domain>
    </domain_list>
</dnsRedirection>
```

- Create or Modify DNS Base Redirection settings

  Create or Modify DNS Base Redirection settings of SR identified by SR id 234.

  Request URL: /api/SRs/234/dnsRedirection

  Request Method: POST or PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dnsRedirection uri="SRs/199/dnsRedirection">
    <System_dnsRedirect_basedOnWE_enable>false</System_dnsRedirect_basedOnWE_enable>
    <System_dnsRedirect_domain_all_enable>false</System_dnsRedirect_domain_all_enable>
    <domain_list>
        <domain>aaa.com</domain>
        <domain>bbb.com</domain>
        <domain>ccc.com</domain>
    </domain_list>
</dnsRedirection>
```

Response XML: None

Routing Settings

Request Routing General Settings

This API provides parity for the following CDSM GUI page.

Devices>Devices>{SR}>Routing Settings>Request Routing Settings>General Settings

Table 2-52 Request Routing General Settings API calls

Resource URL	Method	Function Description
/api/SRs/{id}/routing	GET	Get Request Routing General Settings for an SR identified by {id}.
	POST	Create Request Routing General Settings for an SR identified by {id}.
	PUT	Modify Request Routing General Settings for an SR identified by {id}.
	DELETE	Remove Request Routing General Settings for an SR identified by {id}.

Examples

- Get Request Routing General Settings

  Get configurations of Routing Settings for an SR identified by id 421.

  Request URL:/api/SRs/421/routing

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<routing uri="/api/SRs/421/routing">
  <System_locBasedRouting_enable>false</System_locBasedRouting_enable>
  <System_locBasedRouting_cacheTimeout>691211</System_locBasedRouting_cacheTimeout>
  <System_locBasedRouting_serverType_type>quova</System_locBasedRouting_serverType_typ
   e>
  <System_locBasedRouting_serverType_protocol>http</System_locBasedRouting_serverType_
   protocol>
  <System_locBasedRouting_serverType_service></System_locBasedRouting_serverType_servi
   ce>
  <System_locBasedRouting_serverType_licenseKey></System_locBasedRouting_serverType_li
   censeKey>
  <System_locBasedRouting_serverType_apiKey></System_locBasedRouting_serverType_apiKey
   >
  <System_locBasedRouting_serverType_sharedSecretKey></System_locBasedRouting_serverTy
   pe_sharedSecretKey>
  <System_locBasedRouting_geoLocServer1_serverIp>1.1.1.1</System_locBasedRouting_geoLo
   cServer1_serverIp>
```

```
  <System_locBasedRouting_geoLocServer1_port>212</System_locBasedRouting_geoLocServer1
   _port>
  <System_locBasedRouting_geoLocServer1_serviceName></System_locBasedRouting_geoLocSer
   ver1_serviceName>
  <System_locBasedRouting_geoLocServer1_retry>3</System_locBasedRouting_geoLocServer1_
   retry>
  <System_locBasedRouting_geoLocServer1_timeout>300</System_locBasedRouting_geoLocServ
   er1_timeout>
  <System_locBasedRouting_geoLocServer2_serverIp></System_locBasedRouting_geoLocServer
   2_serverIp>
  <System_locBasedRouting_geoLocServer2_port></System_locBasedRouting_geoLocServer2_po
   rt>
  <System_locBasedRouting_geoLocServer2_serviceName></System_locBasedRouting_geoLocSer
   ver2_serviceName>
  <System_locBasedRouting_geoLocServer2_retry>3</System_locBasedRouting_geoLocServer2_
   retry>
  <System_locBasedRouting_geoLocServer2_timeout>300</System_locBasedRouting_geoLocServ
   er2_timeout>
  <System_contentBasedRouting_enable>false</System_contentBasedRouting_enable>
  <System_contentBasedRouting_redundantCopies>1</System_contentBasedRouting_redundantC
   opies>
  <System_proximityBasedRouting_enable>false</System_proximityBasedRouting_enable>
  <System_proximityBasedRouting_cacheTimeout>1800</System_proximityBasedRouting_cacheT
   imeout>
  <System_proximityBasedRouting_Server0_hostname></System_proximityBasedRouting_Server
   0_hostname>
  <System_proximityBasedRouting_Server0_password></System_proximityBasedRouting_Server
   0_password>
  <System_proximityBasedRouting_Server1_hostname></System_proximityBasedRouting_Server
   1_hostname>
  <System_proximityBasedRouting_Server1_password></System_proximityBasedRouting_Server
   1_password>
  <System_proximityBasedRouting_Server2_hostname></System_proximityBasedRouting_Server
   2_hostname>
  <System_proximityBasedRouting_Server2_password></System_proximityBasedRouting_Server
   2_password>
  <System_proximityBasedRouting_Server3_hostname></System_proximityBasedRouting_Server
   3_hostname>
  <System_proximityBasedRouting_Server3_password></System_proximityBasedRouting_Server
   3_password>
  <System_proximityBasedRouting_Server4_hostname></System_proximityBasedRouting_Server
   4_hostname>
  <System_proximityBasedRouting_Server4_password></System_proximityBasedRouting_Server
   4_password>
  <System_proximityBasedRouting_Server5_hostname></System_proximityBasedRouting_Server
   5_hostname>
  <System_proximityBasedRouting_Server5_password></System_proximityBasedRouting_Server
   5_password>
  <System_proximityBasedRouting_Server6_hostname></System_proximityBasedRouting_Server
   6_hostname>
  <System_proximityBasedRouting_Server6_password></System_proximityBasedRouting_Server
   6_password>
  <System_proximityBasedRouting_Server7_hostname></System_proximityBasedRouting_Server
   7_hostname>
  <System_proximityBasedRouting_Server7_password></System_proximityBasedRouting_Server
   7_password>
</routing>
```

- Create Request Routing General Settings

  Create Request Routing General Settings for an SR with id 421, and without Proximity Server Host configurations.

  Request URL: /api/SRs/421/routing

Request Method: POST

Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<routing uri="/api/SRs/421/routing">
  <System_locBasedRouting_enable>false</System_locBasedRouting_enable>
  <System_locBasedRouting_cacheTimeout>691211</System_locBasedRouting_cacheTimeout>
  <System_locBasedRouting_serverType_type>quova</System_locBasedRouting_serverType_typ
   e>
  <System_locBasedRouting_serverType_protocol>http</System_locBasedRouting_serverType_
   protocol>
  <System_locBasedRouting_serverType_service></System_locBasedRouting_serverType_servi
   ce>
  <System_locBasedRouting_serverType_licenseKey></System_locBasedRouting_serverType_li
   censeKey>
  <System_locBasedRouting_serverType_apiKey></System_locBasedRouting_serverType_apiKey
   >
  <System_locBasedRouting_serverType_sharedSecretKey></System_locBasedRouting_serverTy
   pe_sharedSecretKey>
  <System_locBasedRouting_geoLocServer1_serverIp>1.1.1.1</System_locBasedRouting_geoLo
   cServer1_serverIp>
  <System_locBasedRouting_geoLocServer1_port>212</System_locBasedRouting_geoLocServer1
   _port>
  <System_locBasedRouting_geoLocServer1_serviceName></System_locBasedRouting_geoLocSer
   ver1_serviceName>
  <System_locBasedRouting_geoLocServer1_retry>3</System_locBasedRouting_geoLocServer1_
   retry>
  <System_locBasedRouting_geoLocServer1_timeout>300</System_locBasedRouting_geoLocServ
   er1_timeout>
  <System_locBasedRouting_geoLocServer2_serverIp></System_locBasedRouting_geoLocServer
   2_serverIp>
  <System_locBasedRouting_geoLocServer2_port></System_locBasedRouting_geoLocServer2_po
   rt>
  <System_locBasedRouting_geoLocServer2_serviceName></System_locBasedRouting_geoLocSer
   ver2_serviceName>
  <System_locBasedRouting_geoLocServer2_retry>3</System_locBasedRouting_geoLocServer2_
   retry>
  <System_locBasedRouting_geoLocServer2_timeout>300</System_locBasedRouting_geoLocServ
   er2_timeout>
  <System_contentBasedRouting_enable>false</System_contentBasedRouting_enable>
  <System_contentBasedRouting_redundantCopies>1</System_contentBasedRouting_redundantC
   opies>
  <System_proximityBasedRouting_enable>false</System_proximityBasedRouting_enable>
  <System_proximityBasedRouting_cacheTimeout>1800</System_proximityBasedRouting_cacheT
   imeout>
</routing>
```

Response XML: None

- Modify Request Routing General Settings

  Modify Request Routing General Settings for an SR with id 421. Change the value of the “Location Cache Timeout” with 640000.

  Request URL: /api/SRs/421/routing

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<routing uri="/api/SRs/421/routing">
  <System_locBasedRouting_enable>false</System_locBasedRouting_enable>
  <System_locBasedRouting_cacheTimeout>640000</System_locBasedRouting_cacheTimeout>
```

```
<System_locBasedRouting_serverType_type>quova</System_locBasedRouting_serverType_typ
 e>
<System_locBasedRouting_serverType_protocol>http</System_locBasedRouting_serverType_
 protocol>
<System_locBasedRouting_serverType_service></System_locBasedRouting_serverType_servi
 ce>
<System_locBasedRouting_serverType_licenseKey></System_locBasedRouting_serverType_li
 censeKey>
<System_locBasedRouting_serverType_apiKey></System_locBasedRouting_serverType_apiKey
 >
<System_locBasedRouting_serverType_sharedSecretKey></System_locBasedRouting_serverTy
 pe_sharedSecretKey>
<System_locBasedRouting_geoLocServer1_serverIp>1.1.1.1</System_locBasedRouting_geoLo
 cServer1_serverIp>
<System_locBasedRouting_geoLocServer1_port>212</System_locBasedRouting_geoLocServer1
 _port>
<System_locBasedRouting_geoLocServer1_serviceName></System_locBasedRouting_geoLocSer
 ver1_serviceName>
<System_locBasedRouting_geoLocServer1_retry>3</System_locBasedRouting_geoLocServer1_
 retry>
<System_locBasedRouting_geoLocServer1_timeout>300</System_locBasedRouting_geoLocServ
 er1_timeout>
<System_locBasedRouting_geoLocServer2_serverIp></System_locBasedRouting_geoLocServer
 2_serverIp>
<System_locBasedRouting_geoLocServer2_port></System_locBasedRouting_geoLocServer2_po
 rt>
<System_locBasedRouting_geoLocServer2_serviceName></System_locBasedRouting_geoLocSer
 ver2_serviceName>
<System_locBasedRouting_geoLocServer2_retry>3</System_locBasedRouting_geoLocServer2_
 retry>
<System_locBasedRouting_geoLocServer2_timeout>300</System_locBasedRouting_geoLocServ
 er2_timeout>
<System_contentBasedRouting_enable>false</System_contentBasedRouting_enable>
<System_contentBasedRouting_redundantCopies>1</System_contentBasedRouting_redundantC
 opies>
<System_proximityBasedRouting_enable>false</System_proximityBasedRouting_enable>
<System_proximityBasedRouting_cacheTimeout>1800</System_proximityBasedRouting_cacheT
 imeout>
</routing>
```

Response XML: None

- Remove Request Routing General Settings

  Remove Request Routing General Settings for an SR with id 421. That means configurations will be reset to factory defaults.

  Request URL: /api/SRs/421/routing

  Request Method: DELETE

  Request XML: None

  Response XML: None

Service APIs

Deliver Service

Service Engine Settings

This API provides parity for the following CDSM GUI page:

Service>Service Definition>Delivery Services>{DeliveryServices}>Service Engine Settings

Table 2-53 Service Engine Settings API Calls

Resource URL	Method	Function Description
/api/deliveryServices/{DS_id}/dsvcSeSettings	GET	Get all Service Engine Settings of a Delivery Service identified by {DS_id}.
/api/deliveryServices/{DS_id}/dsvcSeSettings/SEs/{SE_id}	GET	Get the Service Engine Settings of the SE identified by {SE_id} for a delivery service identified by {DS_id}.
	PUT	Modify the Service Engine Settings of the SE identified by {SE_id} for a delivery service identified by {DS_id}.
	DELETE	Remove the Service Engine Settings of the SE identified by {SE_id} for a delivery service identified by {DS_id}.

Example

- Get Delivery Service Settings

  Get all Service Engine Settings of Delivery Service identified by id 8830.

  Request URL: /api/deliveryServices/8830/dnsRedirection

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DsvcSeSettingsList>
  <DsvcSeSettings seid="4026" id="9094" dsid="8830">
    <EnableAbrLive>false</EnableAbrLive>
  </DsvcSeSettings>
  <DsvcSeSettings seid="5984" id="9093" dsid="8830">
```

```
    <EnableAbrLive>true</EnableAbrLive>
    <TmpfsDuration>4</TmpfsDuration>
  </DsvcSeSettings>
</DsvcSeSettingsList>
```

- Modify Delivery Service Settings

  Modify a Service Engine Settings of Delivery Service identified by id 8830.

  Request URL: /api/deliveryServices/8830/dnsRedirection/4026

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DsvcSeSettingsInfo>
    <EnableAbrLive>true</EnableAbrLive>
    <TmpfsDuration>5</TmpfsDuration>
</DsvcSeSettingsInfo>
```

  Response XML: None

Content Deletion

This API provides parity for the following CDSM GUI page.

Service>Service Definition>Delivery Services>Content Deletion

Table 2-54 Content Deletion API calls

Resource URL	Method	Function Description
/api/contentDeletion/tasks	GET	Get list of all Content Deletion Tasks.
	POST	Create a Content Deletion Task in the system.
	DELETE	Remove all completed tasks.
/api/contentDeletion/task/{id}	GET	Get the details of a Content Deletion Task identified by {id}.
	DELETE	Remove a Content Deletion Task identified by {id}.
/api/contentDeletion/task/{id}/cancel	POST	Cancel a Content Deletion Task identified by id.

Examples

- Get all Content Deletion Tasks

Request URL: /api/contentDeletion/tasks

Request Method: GET

Request XML: None

Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DeletionTasks>
  <DeletionTask id="6024" type="API" ctime="2013.12.06 04:06:28" status="2" />
  <DeletionTask id="6018" type="GUI" ctime="2013.12.03 10:22:00" status="2"
comments="Deletion Task Comments" />
  <DeletionTask id="5991" type="GUI" ctime="2013.12.03 09:27:24" status="2" />
  <DeletionTask id="5976" type="API" ctime="2013.12.02 10:15:32" status="5"
comments="my comment here" />
</DeletionTasks>
```

- Get a Content Deletion Task identified by id.

  Get a Content Deletion Task identified by id 4018.

  Request URL: /api/contentDeletion/Tasks/4018

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<DeletionTask id="4018" type="GUI" sync="false" rate="200"
  ctime="2013.11.01 01:50:53" status="2" comments="my comments">
  <ContentDeletions>
    <ContentDeletion url="rtsp://128.0.0.1/a/*">
      <URLDeletion seid="3866" status="17" count="10" />
    </ContentDeletion>
  </ContentDeletions>
  <SEDeletions>
    <SEDeletion seid="3866" sename="49-84"
      seip="10.79.49.84" status="2" message="" />
  </SEDeletions>
</DeletionTask>
```

- Create a Content Deletion Task

  Request URL: /api/contentDeletion/Tasks/

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DeletionRequest sync=”true” comments=”my comments” rate=”20”>
  <Content url="http://10.11.11.1/1000.*" >
    <ServiceEngine id="543"/>
  </Content>
  <Content url="http://10.11.11.1/abc.*" >
  </Content>
</DeletionRequest>
```

  Response XML: None

Other APIs

HTTPS General Settings

This API provides parity for the following CDSM GUI page.

System>Configuration>Https Settings>General Settings

Table 2-55 HTTPS General Settings API calls

Resource URL	Method	Function Description
/api/httpsGenSettings	GET	Get all HTTPS General Settings.
	POST	Create a HTTPS General Settings.
	PUT	Modify a HTTPS General Settings.
	DELETE	Remove a HTTPS General Settings.

Examples

- Get HTTPS General Settings

  Get HTTPS General Settings.

  Request URL: /api/httpsGenSettings

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<httpsGenSettings uri="/api/httpsGenSettings">
    <System_httpsgen_deliveryStreamingMutualAuth>
        True
    </System_httpsgen_deliveryStreamingMutualAuth>
    <System_httpsgen_deliveryStreamingCipherList>
        DEFAULT
    </System_httpsgen_deliveryStreamingCipherList>
</httpsGenSettings>
```

- Create HTTPS General Settings

  Create configurations of HTTPS General Settings with true for **Delivery Streaming Mutual Authentication** and DEFAULT for **Delivery Streaming Supported Cipher List**.

  Request URL: /api/httpsGenSettings

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<httpsGenSettings uri="/api/httpsGenSettings">
  <System_httpsgen_deliveryStreamingMutualAuth>
      true
  </System_httpsgen_deliveryStreamingMutualAuth>
  <System_httpsgen_deliveryStreamingCipherList>
    DEFAULT
  </System_httpsgen_deliveryStreamingCipherList>
```

```
</httpsGenSettings>
```

Response XML: None

- Modify HTTPS General Settings

  Modify configurations of HTTPS General Settings with true for **Delivery Streaming Mutual Authentication** and DEFAULT for **Delivery Streaming Supported Cipher List**.

  Request URL: /api/httpsGenSettings

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<httpsGenSettings uri="/api/httpsGenSettings">
  <System_httpsgen_deliveryStreamingMutualAuth>
    true
  </System_httpsgen_deliveryStreamingMutualAuth>
  <System_httpsgen_deliveryStreamingCipherList>
    DEFAULT
  </System_httpsgen_deliveryStreamingCipherList>
</httpsGenSettings>
```

  Response XML: None

- Remove HTTPS General Settings

  Remove configurations of HTTPS General Settings. That means Sets false for **Delivery Streaming Mutual Authentication** and keeps empty for **Delivery Streaming Supported Cipher List**.

  Request URL: /api/httpsGenSettings

  Request Method: DELETE

  Request XML: None

  Response XML: None

CRL File Schedule

This API provides parity for the following CDSM GUI page.

System>Configuration>HTTPS Settings>CRL File Schedule

Table 2-56 CRL File Schedule Tasks API calls

Resource URL	Method	Function Description
/api/crlScheduleTasks	GET	Get all CRL File Schedule Tasks.
	POST	Create a CRL File Schedule Task.

Table 2-56 CRL File Schedule Tasks API calls

Resource URL	Method	Function Description
/api/crlScheduleTasks/{id}	GET	Get a CRL File Schedule Task identified by {id}.
	PUT	Modify a CRL File Schedule Task identified by {id}.
	DELETE	Remove a CRL File Schedule Task identified by {id}.

Examples

- Get CRL File Schedule Tasks

  Get list of all CRL File Schedule Tasks.

  Request URL: /api/crlScheduleTasks

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<crlScheduleTasks uri="/api/crlScheduleTasks">
  <crlScheduleTasks-row id="471" uri="/api/crlScheduleTasks/471">
    <triggerTime>4/18/2014 17:14:47</triggerTime>
    <isImmediate>0</isImmediate>
    <isExpired>0</isExpired>
    <serviceEngines>234</serviceEngines>
  </crlScheduleTasks-row>
</crlScheduleTasks>
```

- Create CRL File Schedule Task

  Create a CRL File Schedule Task for an SE with id 234. When an SE is assigned to a ” and this task is in **Active** status, then this SE cannot be assigned to other Tasks.

  Request URL: /api/crlScheduleTasks

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<crlScheduleTasks>
    <triggerTime>5/1/2014 17:14:47</triggerTime>
    <isImmediate>0</isImmediate>
    <isExpired>0</isExpired>
    <serviceEngines>234</serviceEngines>
</crlScheduleTasks>
```

  Response XML: None

- Modify CRL File Schedule Task

  Modify a CRL File Schedule Task identified by id 239.

  Request URL: /api/crlScheduleTasks/239

  Request Method: PUT

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<crlScheduleTasks>
  <triggerTime>5/2/2014 17:14:47</triggerTime>
  <isImmediate>0</isImmediate>
  <isExpired>0</isExpired>
  <serviceEngines>234</serviceEngines>
</crlScheduleTasks>
```

Response XML: None

- Remove CRL Schedule Task

  Remove a CRL Schedule Task identified by id 239.

  Request URL: /api/crlScheduleTasks/239

  Request Method: DELETE

  Request XML: None

  Response XML: None

Show Commands

This API provides parity for the following CDSM GUI page:

Devices>Devices>Monitoring>Show/Clear Commands/Show Commands

Table 2-57 Show Commands API Calls

Resource URL	Method	Function Description
/api/cli/show	POST	Execute show command on a specified device.

Examples

- Execute a Show command

  Request URL: /api/cli/show

  Request Method: POST

  Request XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CliExec cli="show alarms">
  <device id="165" />
  <device id="215" />
</CliExec>
```

  Response XML: None

```
<CliExecResponse cli="show alarms">
<device message="Executed" status="1" ip="10.79.53.183" id="165">
<!-- the output of the cli command -->
</device>
<device message=" CLI not executed, device offline." status="2" ip="10.79.53.182"
id="215" />
</CliExecResponse>
```

File Management

This API calls are used to manage all file types.

Table 2-58 File Management API calls

Resource URL	Method	Function Description
/api/FileMgmt/types	GET	Get all file types.
/api/FileMgmt/files;type={type}	GET	List all files identified by {type}.
	POST	Create a new file with given {type}.
/api/FileMgmt/files;type={type}/{id}	GET	Get a file identified by {type} and {id}.
	PUT	Modify a file identified by {type} and {id}.
	DELETE	Remove a file identified by {type} and {id}.
/api/FileMgmt/validate;type={type}	GET	Validate all files identified by {type}.
/api/FileMgmt/validate;type={type}/{id}	GET	Validate a file identified by {type} and {id}.
/api/FileMgmt/refetch;type={type}/{id}	GET	Re-fetch a file identified by {type} and {id}.

Examples

- Get all file types

  Request URL: /api/FileMgmt/types

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileMgmt uri="/api/fileMgmt">
  <result message="Show all supported file types" status="success"/>
  <types>
    <type value="1">Coverage Zone File</type>
    <type value="17">Geo/IP File</type>
    <type value="19">CDN Selector File</type>
    <type value="20">Service Rule File</type>
    <type value="22">NAS File</type>
    <type value="26">Root CA File</type>
    <type value="27">CRL File</type>
  </types>
</fileMgmt>
```

- Get files identified by type

  Get the files identified by type 19 (CDN Selector File).

  Request URL: /api/FileMgmt/files;type=19

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileMgmt uri="/api/fileMgmt">
<result message="4 CDN Selector Files are displayed." status="success"/>
  <file type="19" id="560">
    <originUrl>CDNFile1 </originUrl>
    <destName>CDNFile1 </destName>
    <username>null</username>
    <password>null</password>
    <ttl>10</ttl>
  </file>
  <file type="19" id="554">
    <originUrl>CDNFile2</originUrl>
    <destName>CDNFile2</destName>
    <username>null</username>
    <password>null</password>
    <ttl>10</ttl>
  </file>
  <file type="19" id="555">
    <originUrl>CDNFile3</originUrl>
    <destName>CDNFile3</destName>
    <username>null</username>
    <password>null</password>
    <ttl>10</ttl>
  </file>
</fileMgmt>
```

- Get file identified by type and file id.

  Get a file identified by file type 19 and file id 560

  Request URL: /api/FileMgmt/files;type=19/560

  Request Method: GET

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileMgmt uri="/api/fileMgmt">
<result message="1 CDN Selector File is displayed." status="success"/>
  <file type="19" id="560">
    <originUrl>CDNFile1</originUrl>
    <destName>CDNFile1</destName>
    <username>null</username>
    <password>null</password>
    <ttl>10</ttl>
  </file>
</fileMgmt>
```

- Remove a file identified by file type and id

  Remove a file identified by file type 19 and file id 560.

  Request URL: /api/FileMgmt/files;type=19/560

Request Method: DELETE

Request XML: None

Response XML: None

- Validate files identified by file type and id

  This example tries to re-fetch a file identified by type 19 and id 560, but this file is invalid file. So that we get a response with “fail” status.

  Request URL: /api/FileMgmt/validate;type=19/560

  Request Method: POST

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileMgmt uri="/api/fileMgmt">
  <result message="Invalid File" status="fail"/>
  <error message="ERROR: Parser Fatal Error at (line 1, char 1): An exception
occurred! Type:UTFDataFormatException, Message:invalid byte 1 () of a 1-byte
sequence."/>
</fileMgmt>
```

- Re-fetch a file identified by type and id

  This example tries to re-fetch a file identified by type 19 and id 560, but this file is an uploaded file cannot be re-fetched. So that we get a response with “fail” status.

  Request URL: /api/FileMgmt/refetch;type=19/560

  Request Method: POST

  Request XML: None

  Response XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fileMgmt uri="/api/fileMgmt">
  <result message="Uploaded file could not be refetched" status="fail"/>
  <error message="Uploaded file could not be refetched"/>
</fileMgmt>
```

CDSM Legacy APIs

Replication Status APIs

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

Replication Status API Actions

The Replication Status API is the ReplicationStatusApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- getDeliveryServices
- getSEsOfDeliveryService
- getDelivheryServicesOfSE
- getReplicatedContent
- getNonReplicatedContent
- getContent
- getStatusOfContentItems
- getStatusOfContentItemInDeliveryService

getDeliveryServices

Obtains the status of content replication of specified delivery services.

Parameter

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

Return

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

Syntax

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

getSEsOfDeliveryService

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

Parameter

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

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

Return

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

Syntax

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

getDelivheryServicesOfSE

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

Parameter

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

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

Return

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

Syntax

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

getReplicatedContent

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

Parameter

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

  One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards * or ?.
- Refetch (optional)—The default is false.

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

Return

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

Syntax

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

getNonReplicatedContent

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

Parameter

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

  One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards * or ?.
- Refetch (optional)—The default is false.

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

Return

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

Syntax

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

getContent

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

Parameter

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

  One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards * or ?.
- Refetch (optional)—The default is false.

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

Return

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

Syntax

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

getStatusOfContentItems

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

Parameter

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

  One or more content names or patterns must each be separated by a comma. Patterns can contain the wildcards * or ?.
- Refetch (optional)—The default is false.

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

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

Return

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

Syntax

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

getStatusOfContentItemInDeliveryService

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

Parameter

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

Return

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

Note This action must be called after the getStatusOfContentItems action; otherwise, unexpected output results. The URL must be one of the URLs listed in the output of the getStatusOfContentItems action.

Syntax

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

XML-Formatted Output for Replication Status

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

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

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

Provisioning APIs

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

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

Delivery Service Provisioning API Actions

The Delivery Service Provisioning API is the ChannelApiServlets.

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

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

Table 3-1 lists the unused output fields.

Table 3-1 Output Fields Not Used in the CDS

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

Syntax

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

This servlet performs one or more of the following actions:

- createDeliveryService
- createDeliveryServiceLoc
- createDeliveryServiceGenSettings
- addManifest
- assignSEs
- assignDeliveryServiceIp
- fetchNow
- modifyDeliveryService
- modifyDeliveryServiceLoc
- modifyDeliveryServiceGenSettings
- modifyManifest
- unassignSEs

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

createDeliveryService

Creates a delivery service.

Parameter

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

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

Note The mcastEnable parameter is supported in Release 3.1.1.

Return

The newly created delivery service ID.

Syntax

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

createDeliveryServiceLoc

Creates a delivery service location object.

Parameter

- hssStreamingFromNas(option)—The default is false

Return

The newly created delivery service location object value.

Syntax

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

createDeliveryServiceGenSettings

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

Parameter

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

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

Return

The newly created delivery service general settings.

Syntax

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

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

addManifest

Adds a Manifest file to a specified delivery service.

Parameter

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

Return

The updated delivery service record.

Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=addManifest& deliveryService=<deliveryService_ID>&manifest=<manifest_URL>&ttl=<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=
<truelfalse>]

assignSEs

Assigns Service Engines to a specified delivery service.

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

Parameter

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

Rules

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

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

Note A cluster is the same thing as a Service Engine.

Return

None.

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

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

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

Syntax

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

assignDeliveryServiceIp

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

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

Parameter

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

Rules

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

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

Return

None.

Syntax

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

fetchNow

Immediately fetches the Manifest file.

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

Parameter

Delivery service ID (required)

Return

None.

Syntax

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

modifyDeliveryService

Modifies delivery service settings.

Parameter

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

Note The mcastEnable parameter is supported in Release 3.1.1.

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

Return

The updated delivery service record.

Syntax

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

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

modifyDeliveryServiceLoc

Modifies delivery service location.

Parameter

- hssStreamingFromNas(option)—The default is false

Return

The modified location object value.

Syntax

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

modifyDeliveryServiceGenSettings

Modifies delivery service general settings.

Parameter

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

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

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

Return

The updated delivery service general settings record.

Syntax

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

modifyManifest

Modifies Manifest file settings.

Parameter

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

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

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

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

Return

The updated delivery service record.

Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyManifest&deliveryService=<deliveryService_ID>[&manifest=<manifest_URL>][&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>

addContentItem

Creates and adds a content item to a delivery service.

Parameter

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

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

Return

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

Error message is returned in each of below conditions:

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

Syntax

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

HH:MM:SS>][&stopServTime=<MM/DD/YYYYHH:MM:SS>][&username=<user_name_to_fetch_content>][&password=<password_to_fetch_content>][&ntlmUserDomain=<user_name_in_the_domain>][&ignoreQueryStr=<true|false>][&playDura=<item_type_of_content>][&ttl=<frequency_of_rechecking_in_minutes>][&retryInterval=<frequency_of_rechecking_if_acquisition_fails_in_minutes>][&requireAuth=<true|false>][&rMimeType<1_to_5>=<mime_type_of_acquisition_rule>][&rExtension<1_to_5>=<file_extension_of_acquisition_rule>][&rTimeBefore<1_to_5>=<file_modified_before_MM/DD/YYYY HH:MM:SS>][&rTimeAfter<1_to_5>=<file_modified_after_MM/DD/YYYY HH:MM:SS>][&rMinimumSize<1_to_5>=<minimum_size_of_file_in_MB>][&rMaxSize<1_to_5>=<maximum_size_of_file_in_MB>

modifyContentItem

Modifies a content item to a delivery service.

Parameter

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

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

Return

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

Error message is returned in each of below conditions:

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

Syntax

https://<cdsmIpAddress>:
8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=modifyContentItem&deliveryService=<deliveryService_ID>&sourceURL=<source_url_of_content>[&depth=<link_depth_to_be_crawled>][&highPriority=<true|false>][&disableBasicAuth=<true|false>][&weakCert=<true|false>][&startServTime=<MM/DD/YYYY
HH:MM:SS>][&stopServTime=<MM/DD/YYYYHH:MM:SS>][&username=<user_name_to_fetch_content>][&password=<password_to_fetch_content>][&ntlmUserDomain=<user_name_in_the_domain>][&ignoreQueryStr=<true|false>][&playDura=<item_type_of_content>][&ttl=<frequency_of_rechecking_in_minutes>][&retryInterval=<frequency_of_rechecking_if_acquisition_fails_in_minutes>][&requireAuth=<true|false>][&rMimeType<1_to_5>=<mime_type_of_acquisition_rule>][&rExtension<1_to_5>=<file_extension_of_acquisition_rule>][&rTimeBefore<1_to_5>=<file_modified_before_MM/DD/YYYY HH:MM:SS>][&rTimeAfter<1_to_5>=<file_modified_after_MM/DD/YYYY
HH:MM:SS>][&rMinimumSize<1_to_5>=<minimum_size_of_file_in_MB>][&rMaxSize<1_to_5>=<maximum_size_of_file_in_MB>]

deleteContentItem

Deletes content items.

Parameter

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

Return

Error message is returned in each of below conditions:

- Mandatory parameter is missing.

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

Syntax

https://<cdsmIpAddress>:

8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=deleteContentItem&deliveryService=<deliveryService_ID>&deleteItemUrls=<csv_list_of_source_url_to_delete>

processContentChanges

Processes the content changes to the delivery service.

Parameter

- Delivery service ID (required)

Return

Error message is returned in each of below conditions:

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

Syntax

https://<cdsmIpAddress>:

8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=processContentChanges&deliveryService=<deliveryService_ID>

manageHostProxySettings

Parameter

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

Return

Error message is returned in each of below conditions:

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

- Invalid value of parameter is given.

Syntax

https://<cdsmIpAddress>:

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

createContentOrigin

Creates a content origin.

Parameter

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

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

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

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

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

Return

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

Syntax

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

modifyContentOrigin

Modifies content origin settings.

Parameter

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

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

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

Return

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

Syntax

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

deleteContentOrigin

Deletes content origins.

Parameter

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

Return

A confirmation that the content origins have been deleted.

Syntax

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

applyRuleFile

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

Parameter

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

Return

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

Syntax

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

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

applyGeoIpFile

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

Parameter

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

Return

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

Syntax

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

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

configFailoverSettings

parameter

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

Return

Error message is returned in each of below conditions:

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

Syntax

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

createFailoverOS

Parameter

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

Return

Error message is returned in each of below conditions:

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

Syntax

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

modifyFailoverOS

Parameter

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

Return

Error message is returned in each of below conditions:

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

Syntax

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

deleteFailoveOS

Parameter

- originServer (required)

Return

Error message is returned in each of below conditions:

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

Syntax

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

switchToOS

Parameter

- contentOrigin (required)
- originServer (required)

Return

Error message is returned in each of below conditions:

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

Syntax

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

createChannelDeviceMcastConfig

Parameter

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

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

Return

Error message is returned in each of below conditions:

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

Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.ChannelApiServlet?action=createChannelDeviceMcastConfig&deliveryService=<Channel_id>&seDevice=<CeConfig_id>&senderEnabled=<true|false>&receiverEnabled =<true|false>[&unicastSenderEnabled=<true|false>][&unicastReceiverEnabled =<true|false>]

getChannelDeviceMcastConfig

Parameter

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

Return

Error message is returned in each of below conditions:

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

Syntax

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

modifyChannelDeviceMcastConfig

Parameter

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

- Unicast receiver configuration (required)

Return

Error message is returned in each of below conditions:

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

Syntax

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

deleteChannelDeviceMcastConfig

Parameter

- Configuration ID (required)

Return

Error message is returned in each of below conditions:

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

Syntax

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

Location Provisioning API Actions

The Location Provisioning API is the LocationApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- createLocation
- modifyLocation
- deleteLocation

createLocation

Creates a specified location.

Parameter

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

Return

The newly created location object.

Syntax

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

modifyLocation

Modifies a specified location.

Parameter

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

Return

The modified location object.

Syntax

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

deleteLocation

Deletes a specified location.

Parameter

Location ID (required)

Return

A message that the specified location has been deleted.

Syntax

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

Service Engine Provisioning API Actions

The Service Engine Provisioning API is the CeApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- activateSe
- changeSeLocation
- deleteSe
- setSeMgmtIp
- setMulticast

activateSe

Activates a specified Service Engine.

Parameter

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

Return

The modified Service Engine object.

Syntax

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

changeSeLocation

Changes the location of a specified Service Engine.

Parameter

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

Return

The modified Service Engine object.

Syntax

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

deleteSe

Deletes a specified Service Engine.

Parameter

Service Engine ID (required)

Return

A message that the specified Service Engine has been deleted.

Syntax

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

setSeMgmtIp

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

Parameter

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

Return

The modified Service Engine object.

Syntax

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

setMulticast

Enables an SE as a multicast sender and multicast receiver.

Parameter

- Service Engine (required)—CeConfig_Id is the same as the se_ID
- Enable Sender (optional)
- Enable Receiver (optional)

Return

The modified Service Engine object.

Syntax

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

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 VDS Software," provides a DTD for the information that is returned.

Syntax

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

validateProgramFile

Fetches a program file using HTTP and validates it.

Parameter

Program file URL (required)

Return

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

Syntax

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

assignDeliveryService

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

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

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

Parameter

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

Return

The updated program record.

Syntax

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

assignSEs

Assigns Service Engines to a program.

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

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

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

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

Parameter

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

Return

The updated program record.

Syntax

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

fetchNow

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

Parameter

Program ID (required)

Return

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

Syntax

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

modifyProgramFile

Modifies program file settings.

Parameter

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

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

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

Return

The updated program record.

Syntax

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

unassignDeliveryService

Removes a delivery service from the specified program.

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

Parameter

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

Return

The updated program record.

Syntax

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

unassignSEs

Removes Service Engines from the specified program.

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

Parameter

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

Return

The updated program record.

Syntax

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

deletePrograms

Deletes programs.

Parameter

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

Return

None.

Syntax

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

Media API Actions for Programs

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

Note If the Media API is used to change the media list and there is an associated program file (XML file) that is used with the Program API, the program file must be updated with the changed media list. Use the getPrograms List API action to get the media list for the program, and insert this list in the XML program file. For more information, see the "getPrograms" section on page 3-60.

Syntax

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

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

This servlet performs one or more of the following actions:

- addMedia
- updateMedia
- deleteMedia

addMedia

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

Parameters

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

Return

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

Syntax

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

updateMedia

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

Note The list of media files in the update action must have the same media IDs and number of objects as the 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 3-60.

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://<cdsmIpAdress>: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.HttpsURLConnection;
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.

Cache Storage Priority Class API Actions

The Cache Storage Priority Class API is the StoragePrioClassApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- createStoragePrioClass
- modifyStoragePrioClass
- deleteStoragePrioClass

createStoragePrioClass

Creates a storage priority class.

Parameter

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

Return

The newly created storage priority class with a StoragePriorityClass ID.

Syntax

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

modifyStoragePrioClass

Modifies a storage priority class.

Parameter

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

Return

The modified storage priority class.

Syntax

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

deleteStoragePrioClass

Deletes a storage priority class.

Parameter\

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

Return

None.

Syntax

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

Multicast Cloud API Actions

The Multicast Cloud API is the MCastApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- createCloud
- modifyCloud
- deleteCloud
- assignReceiverSe
- unassignReceiverSe
- assignDeliveryService
- unassignDeliveryService
- modifyChannelMCast

createCloud

Creates a multicast cloud.

Parameter

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

- PGM router assist—True means IP routers are used to assist in distribution of content. (optional)
- Description—(optional)
- defaultDSDataRate—Default multicast data rate for delivery services in kilo bits per second (optional)
- maxConcurrentSessions—Maximum concurrent session (optional)

Note The primarySenderSe and backupSenderSe needs a clusterId; for example, ClusterConfig_2221. The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine.The Listing API can be used to get the cluster ID. For more information, see the "getSEs" section on page 3-56 and "getClusters" section on page 3-57.

Note **Carousel passes** and **Multicast Rearm Timer** fields will take the default values while creating multicast cloud. To modify the default values, use **modifyCloud** API.

Return

The newly created multicast cloud.

Syntax

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

modifyCloud

Modifies a multicast cloud.

Parameter

- Cloud ID—Multicast cloud ID (required)
- Name—Multicast cloud name (optional)
- Advertisement IP—Unique advertisement address (optional)
- Port—Port used for file addresses (optional)
- Start IP address— Start of the IP address range, which must be within the range 224.0.0.0 to 239.255.255.255 (optional)
- End IP address—End of the IP address range (optional)
- Primary sender SE—Primary sender SE (optional)
- Default multicast out bandwidth—Maximum multicast rate in kilobits per second (optional)
- Multicast medium—Means of transmitting the multicast (Satellite or Terrestrial). Satellite is default (optional)

- FEC transmission group—Size of the FEC (forward error correction) block in packets. Allowable inputs are 2, 4, 8, 16, 32, 64, and 128. Default is 16. (optional)
- Carousel passes—Maximum number of times a multicast sender sends missing content (optional)
- Multicast Rearm Timer—Timer value after which multicast carousel is reset (optional)
- Carousel delay—Delay, in minutes, between file transmissions (optional)
- Backup sender SE—Backup sender SE (optional)
- Failover grace period—Period of time backup sender goes without getting heartbeat from primary sender before taking over (optional)
- Fallback grace period—Period of time primary sender goes without getting heartbeat from backup sender before taking over (optional)
- PGM router assist—True means IP routers are used to assist in distribution of content. (optional)
- Description—(optional)
- defaultDSDataRate—Default multicast data rate for delivery services in kilo bits per second (optional)
- maxConcurrentSessions—Maximum concurrent session (optional)

Note The primarySenderSe and backupSenderSe needs a clusterId; for example, ClusterConfig_2221. The Service Engine and cluster form a one-to-one relationship. A cluster is considered a wrapper around the Service Engine.The Listing API can be used to get the cluster ID. For more information, see the "getSEs" section on page 3-56 and "getClusters" section on page 3-57.

Return

The modified multicast cloud.

Syntax

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

deleteCloud

Deletes a multicast cloud.

Parameter

- Cloud ID—Multicast cloud ID (required)

Return

None.

Syntax

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

assignReceiverSe

Assigns a receiver SE to multicast cloud.

Parameter

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

Note The SE cluster ID is the needed to identify the SE. The getSEs action of the Listing API can be used to get the cluster ID. For more information, see the "getSEs" section on page 3-56.

Return

None.

Syntax

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

unassignReceiverSe

Removes a receiver SE from a multicast cloud.

Parameter

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

Note The SE cluster ID is the needed to identify the SE. The getSEs action of the Listing API can be used to get the cluster ID. For more information, see the "getSEs" section on page 3-56.

Return

None.

Syntax

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

assignDeliveryService

Assigns a multicast cloud to a delivery service.

Parameter

- Cloud ID—Multicast cloud ID (required)
- Delivery service ID —Format is Channel_*xxx*, where *xxx* is the ID of the delivery service (required)
- Multicast IP address—Multicast IP address assigned to this delivery service from multicast cloud address range (required)
- carouselPass—Carousel pass value (optional)
- Multicast Rearm Timer—Timer value after which multicast carousel is reset (optional)
- maxDataRate—Maximum multicast data rate for the delivery services in kilo bits per second (required)
- maxConcurrentSessions—Maximum concurrent session (optional)
- fecTransmissionGroup—FEC transmission group value (optional)

Note The assignDeliveryService action expects the deliveryService parameter to be in the form Channel_*xxx*, where *xxx* is the ID of the delivery service.

Return

None.

Syntax

https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=assignDeliveryService&cloud=<MCastCloud_Id>&Delivery Service=<channel_Id>[&mcastIp=<multicast_Ip>][&carouselPass=<carousel_Pass>][&mcastRearmTimer=<mcast_Rearm_Timer>][&maxDataRatemax_data_rate=<Maximum_date_rate_control_for_DS>][&maxConcurrentSessions=<Maximum_concurrent_sessions>][&fecTransmissionGroup=<fec_trans_group>]

unassignDeliveryService

Removes a multicast cloud from a delivery service.

Parameter

- Cloud ID—Multicast cloud ID (required)
- Delivery service ID —Format is Channel_*xxx*, where *xxx* is the ID of the delivery service (required)

Note The unassignDeliveryService action expects the deliveryService parameter to be in the form Channel_*xxx*, where *xxx* is the ID of the delivery service.

Return

None.

Syntax

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

modifyChannelMCast

Modifies a ChannelMcast configuration.

Parameter

- Cloud ID—Multicast cloud ID (required)
- Name—Multicast cloud name (required)
- fecTransmissionGroup—FEC transmission group value (optional)
- checkPointTransfer—To configure the FEC proactive parity delay
- fecParitySize—FEC proactive parity size value (optional)
- fecBlockDelay—FEC proactive parity delay value (optional)
- carouselPass—Carousel pass value (optional)
- Multicast Rearm Timer—Timer value after which multicast carousel is reset (optional)
- maxDataRate—Maximum multicast data rate for the delivery services in kilo bits per second (required)
- maxConcurrentSessions—Maximum concurrent session (optional)

Return

The modified ChannelMcast configuration.

Syntax

https://<cdsmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=modifyChannelMCast&DeliveryService=<Channel_id>&cloud=<MCastCloud_id>&maxDataRate=<Maximum_date_rate_control>[&maxConcurrentSessions=<Maximum_concurrent_sessions>]

https://<cdmIpAddress>:8443/servlet/com.cisco.unicorn.ui.MCastApiServlet?action=modifyChannelMCast&DeliveryService=<channel_Id>&cloud=<cloud_Id>[&fecTransmissionGroup=<fec_transmission_group>][&checkPointTransfer=<true|false>][&fecParitySize=<fec_parity_size>][&fecBlockDelay=<fec_block_delay>][&carouselPass=<carousel_pass>][&mcastRearmTimer=<mcast_Rearm_Timer>]&maxDataRate=<Maximum_date_rate_control>[&maxConcurrentSessions=<Maximum_Concurrent_Sessions>]

External System API Actions

The External System API is the ExternalSysApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- create
- modify

- delete

create

Creates an external system.

Parameter

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

Return

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

Syntax

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

modify

Modifies external system.

Parameter

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

- Prime Central Fault Manager port

Note The external system configuration ID is returned when the external system creation is successful. Alternatively, you can use the getExternalSystems action of the Listing API servlet. For more information, see the "getExternalSystem" section on page 3-62.

Return

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

Syntax

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

delete

Deletes an external system.

Parameter

- External System Configuration ID

Return

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

Syntax

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

Listing APIs

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

- Listing API Actions, page 3-54
- Device API Actions, page 3-62

Listing API Actions

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

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

- getSEs
- getDeliveryServices
- getContentOrigins

Table 3-2 lists the unused output fields.

Table 3-2 Output Fields Not Used in the CDS

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

Syntax

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

This servlet performs one or more of the following actions:

- getContentOrigins
- getDeliveryServices
- getSEs
- getClusters
- getLocations
- getDeviceGroups
- getObjectById
- getObjectByName
- getPrograms
- getPgmMcastAddrInUse
- getMcastAddrInUse
- getMCastClouds

- getStoragePrioClasses
- getExternalSystem

getContentOrigins

Lists selected content origin names or lists every content origin.

Parameter

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

Return

A list of all content origins specified and their details.

Syntax

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

getDeliveryServices

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

Parameter

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

Return

A list of all delivery services specified and their details.

Syntax

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

getSEs

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

Parameter

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

Return

A list of all Service Engines specified and their details.

Syntax

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

getClusters

Lists selected cluster names or lists every cluster.

Parameter

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

Note A cluster is the same thing as a Service Engine.

Return

A list of all clusters specified and their details.

Syntax

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

getLocations

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

Parameter

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

Return

The requested location record.

Syntax

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

getDeviceGroups

Lists selected device group names or lists all device groups.

Parameter

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

Return

A list of all device groups specified and their details.

Syntax

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

getObjectById

Lists an object, based on its string ID.

Parameter

Object string ID

The following are the object types:

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

Return

The requested object.

Syntax

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

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

getObjectByName

Lists an object, based on its name.

Parameter

Object type:object name

The following are the object types:

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

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

Return

The requested object.

Syntax

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

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

getPrograms

Lists all programs specified or all programs and their details.

Parameter

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

Return

A list of all programs specified and their details.

Syntax

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

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

getPgmMcastAddrInUse

Lists all multicast addresses currently in use by programs.

Parameter

None.

Return

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

Syntax

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

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

getMcastAddrInUse

Lists all multicast addresses currently in use.

Parameter

None.

Return

A list of all the multicast addresses currently in use.

Syntax

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

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

getMCastClouds

Lists all the multicast clouds.

Parameter

mcastCloud_name

Return

A list of all the multicast clouds.

Syntax

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

getStoragePrioClasses

Lists the storage priority classes and their StoragePriorityClass IDs.

Parameter

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

Return

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

Syntax

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

getExternalSystem

Lists the external system and their ExternalSysConfig IDs.

Parameter

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

Return

The information of the external systems is returned.

Syntax

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

Device API Actions

The Device API is the DeviceApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- getDeviceStatus
- getDevices

getDeviceStatus

Lists the status of a device by name.

Parameter

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

Note The name of the device is case sensitive.

Return

A list of devices and their status.

Syntax

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

getDevices

Provides information about the devices in the CDS.

Parameters

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

- Name—Device name
- ID—Device ID

Return

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

Syntax

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

Statistics APIs

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

- Monitoring Statistics API Actions, page 3-63
- Streaming Statistics API Actions, page 3-66

Monitoring Statistics API Actions

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

Syntax

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

This servlet performs one or more of the following actions:

- getSeStats
- getLocationStats
- getCdnStats

getSeStats

Obtains monitoring statistics information for the specified Service Engine.

Parameter

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

  The monitoring statistics types are:

  - bytes_served
  - bandwidth_efficiency_gain
  - streaming_sessions
  - cpu_utilization

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

  The options are:

  - last_hour
  - last_day
  - last_week
  - last_month
  - custom

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

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

  The time zones are:

  - utc—Time in UTC (Coordinated Universal Time)
  - se_local_time—Time zone specified on the Service Engine
  - cdsm_local_time—Time zone specified on the CDSM

Return

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

Syntax

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

getLocationStats

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

Parameter

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

  The monitoring statistics types are:

  - bytes_served

  - bandwidth_efficiency_gain
  - streaming_sessions
- Time frame (optional)—The time period over which monitoring statistics are obtained. The default is last_hour.

  The options are:
  - last_hour
  - last_day
  - last_week
  - last_month
  - custom

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

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

  The time zones are:
  - utc—Time in UTC (Coordinated Universal Time)
  - cdsm_local_time—Time zone specified on the CDSM

Return

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

Syntax

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

getCdnStats

Obtains monitoring statistics information for the entire CDS network.

Parameter

- Monitoring statistics type (required)

  The monitoring statistics types are:

  - bytes_served
  - bandwidth_efficiency_gain
  - streaming_sessions
- Time frame (optional)—The time period over which monitoring statistics are obtained. The default is last_hour.

  The options are:
  - last_hour
  - last_day
  - last_week
  - last_month
  - custom

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

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

  The time zones are:
  - utc—Time in UTC (Coordinated Universal Time)
  - cdsm_local_time—Time zone specified on the CDSM

Return

Requested monitoring statistics data for the selected time period for all the Service Engines in the CDS network.

Syntax

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

Streaming Statistics API Actions

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

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

Syntax

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

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

- getHttp
- getMovieStreamer
- getWmt

getHttp

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

Parameter

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

Return

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

Syntax

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

getMovieStreamer

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

Parameter

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

Return

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

Syntax

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

getWmt

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

Parameter

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

Return

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

Syntax

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

XML-Formatted Output for Streaming Statistics

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

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

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

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

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

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

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

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

]>
```

File Management APIs

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

- File Management API Actions, page 3-69
- Certificate and Key File Management API, page 3-78

Using Multipart/Form-Data Request to Upload a File

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

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

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

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

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

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

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

- registerFile
- validateFile
- modifyFile

The refetchFile only uses importMethod=import.

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

- registerFile
- modifyFile

File Management API Actions

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

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

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

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

- createContentOrigin, page 3-24
- modifyContentOrigin, page 3-25

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

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

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

- registerFile
- modifyFile
- listFile

Table 6-1 lists the unused output fields.

Table 3-3 Output Fields Not Used in the CDS

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

Syntax

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

The servlet performs one or more of the following actions:

- listTypes
- registerFile
- validateFile
- refetchFile
- modifyFile
- deleteFile
- listFile
- applyCZ
- applyCdnSelector

listTypes

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

Parameter

None.

Return

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

Syntax

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

registerFile

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

Parameter

- File type (required)

  The settings are:

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

  The settings are:

  - Import
  - Upload

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

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

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

Rule

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the "Using Multipart/Form-Data Request to Upload a File" section on page 3-69.

Return

A confirmation that the file has been registered.

Note In the XML file returned, an internal reference is assigned to the file in the format of FileInfo_*xxx,* where *xxx* is the ID of the file.

Syntax

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

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

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

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

validateFile

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

Parameter

- File type (required)

  The settings are:

  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
- File ID (required if the destination filename is not specified)—The format is FileInfo_*xxx,* where *xxx* is the ID of the file.
- Import method (required if ID is not specified)

  The settings are:

  - Import
  - Upload
- Destination filename (required if ID is not specified)

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

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

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

Rules

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

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the "Using Multipart/Form-Data Request to Upload a File" section on page 3-69.

Return

A confirmation that the file is valid.

Syntax

To validate a registered file, use the following syntax:

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

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

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

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

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

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

refetchFile

Notifies the CDSM to refetch a registered file immediately.

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

Parameter

- File type (required)

  The settings are:

  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (required)—The format is FileInfo_*xxx,* where *xxx* is the ID of the file.

Return

A confirmation that the file will be refetched shortly.

Syntax

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

modifyFile

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

Parameter

- File type (required)

  The settings are:

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

  The settings are:

  - Import

  - Upload
- Destination filename

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

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

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

Rules

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

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the "Using Multipart/Form-Data Request to Upload a File" section on page 3-69.

Return

A confirmation that the file was successfully modified.

Syntax

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

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

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

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

deleteFile

Removes a registered file from the CDSM.

Parameter

- File type (required)

  The settings are:

  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (required)—The format is FileInfo_*xxx,* where *xxx* is the ID of the file.

Rule

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

Return

A confirmation that the file has been deleted.

Syntax

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

listFile

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

Parameter

- File type (required)

  The settings are:

  - 1—Coverage Zone file
  - 17—Geo/IP file
  - 19—CDN Selector file
  - 20—Rule file
  - 22—NAS file
  - 26—HTTPS Root Certificate file
- File ID (optional)—The format is FileInfo_*xxx,* where *xxx* is the ID of the file.

Return

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

Syntax

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

applyCZ

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

Parameter

- Target—Deploy a Coverage Zone file globally to the entire CDS network or deploy a Coverage Zone file on the specified SR only. Valid values are:
  - Global
  - SR ID —The format is CrConfig_*xxx*, where *xxx* is the ID of the active SR.
- Coverage Zone file ID. Valid values are:
  - None—Removes the association of the Coverage Zone file with the target.
  - File ID (required)—The format is FileInfo_*xxx,* where *xxx* is the ID of the file.

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

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

Return

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

Syntax

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

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

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

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

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

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

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

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

applyCdnSelector

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

Parameter

- SR ID— The format is CrConfig_*xxx,* where *xxx* is the ID of the SR.
- CDN Selector file. Valid values are:

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

Return

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

Syntax

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

Certificate and Key File Management API

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

The Certificate and Key File Management API uses the CertKeyFileMgmtApiServlet.

Syntax

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

This servlet performs one or more of the following actions:

- registerFile
- modifyFile
- deleteFile
- listFile

registerFile

Uploads the certificate and key files.

Parameters

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

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

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

Rule

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the "Using Multipart/Form-Data Request to Upload a File" section on page 3-69.

Return

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

Syntax

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

modifyFile

Uploads and overwrites the certificate and key files.

Parameters

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

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

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

Rule

When the import method is set to upload, the source file is required when posting a multi-part form-data request. For more information, see the "Using Multipart/Form-Data Request to Upload a File" section on page 3-69.

Return

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

Syntax

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

deleteFile

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

Parameters

None.

Return

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

Syntax

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

listFile

Provides information about uploaded certificate file and the key file.

Parameters

None.

Return

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

Syntax

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

Request Routing Engine APIs

This chapter describes the following APIs for the Request Routing Engine:

- Request Routing Engine APIs, page 4-1
- Last-Resort URL Translator Web Services API, page 4-2

Request Routing Engine APIs

The Request Routing Engine APIs allows another platform's software client to make queries, in the form of an HTTP request, to the Request Routing Engine about which Service Engine the Request Routing Engine selects.

Note The Request Routing Engine APIs does not support service-aware routing.

Parameter

The Request Routing Engine determines the best Service Engine based on the client IP address and requested URL; therefore, the platform's software client must include these two parameters in the API query.

- Service Router IP address (required)
- CDNURL (required)
- ClientIP (required)

Request Syntax

http://*ServiceRouterIP*/routeURL?CDNURL=<*Requested URL*>&ClientIP=<*ClientIP*>

Request Example

The following is an example of a client's API HTTP request:

```
http://10.252.250.118/routeURL?CDNURL=http://cds.mov/smooth_linear/cnn.isml/victory&Client
IP=165.137.13.117
```

In this example, the request parameters have the following values:

- Service Router IP address—10.252.250.118
- CDNURL—http://cds.mov/smooth_linear/cnn.isml/victory
- ClientIP—192.0.2.121

Return

The Request Routing Engine returns an HTTP response with an XML payload with the following information:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RoutedURLResponse>
<primaryContentRoutedURL>SRresponse</primaryContentRoutedURL>
</RoutedURLResponse>
```

The response identifies the name of the Service Engine the Request Routing Engine selected.

Return Example

For the example HTTP request described above, the HTTP response would be the following:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RoutedURLResponse>
<primaryContentRoutedURL>http://str01-hub03.se.cds.mov/smooth_linear/cnn.isml/victory</pri
maryContentRoutedURL>
</RoutedURLResponse>
```

In this example, str-01-hub03 is the Service Engine the Request Routing Engine selected.

Note When there are API query requests for clientaccesspolicy.xml or crossdomain.xml, the requests are treated like normal API request and the XML payload is sent in the HTTP response as described above.

If requests for clientaccesspolicy.xml or crossdomain.xml are sent directly to the Request Routing Engine (non-API request), then these files are served. For more information about the Cross-Domain Policy, refer to *VDS Internet Streamer 3.3 Software Configuration Guide*. See the "Related Publications" section on page xiii for links to documentation online.

Last-Resort URL Translator Web Services API

The last-resort URL translator provides a way to dynamically translate the client request URL in order to redirect the client to a different CDN. With the URL translator option, the following occurs if the Request Routing Engine uses last-resort routing for a client request:

1. The Request Routing Engine contacts the third-party URL translator through the web service API.
2. The third-party URL translator sends the translated URL in the response to the Request Routing Engine.
3. The Request Routing Engine sends a 302 redirect message to the client with the translated URL it received from the third-party URL translator.

The timeout for connecting to the URL translator server is 500 milliseconds. There are no retries if the URL translator cannot be reached.

If there is no configuration on the URL translator for the requested domain or the connection timeout threshold has been reached, the Request Routing Engine last-resort routing falls back to the alternate domain configuration.

The Request Routing Engine sends the client IP address and the requested URL to the URL translator. The URL translator returns the translated URL and a signed URL.

Note Customer ID and CDN name fields are not currently supported.

The Web Services Description Language (WSDL) for the Request Routing Engine to URL translator communication is the following:

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="CDNUrlTranslationDefinitions"
    targetNamespace="http://cisco/CDS/CDNUrlTranslation"
    xmlns:ut="http://cisco/CDS/CDNUrlTranslation"
    xmlns:utxsd="http://schemas.cisco/CDS/CDNUrlTranslation/Schema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

    <wsdl:types>
        <xsd:schema targetNamespace="http://schemas.cisco/CDS/CDNUrlTranslation/Schema"
                    xmlns:xsd="http://www.w3.org/2001/XMLSchema">

              <!-- UrlTranslationRequest [ClientIP string, Url string, CustomerId string, CDNName string] -->

               <xsd:element name="UrlTranslationRequest">
                    <xsd:complexType>
                       <xsd:annotation>
                           <xsd:documentation>Request to translate the url for a specific
CDN</xsd:documentation>
                       </xsd:annotation>
                       <xsd:sequence>
                           <xsd:element name="ClientIP" minOccurs="1" nillable="false" type="xsd:string">
                                <xsd:annotation>
                                   <xsd:documentation>Get Client IP address</xsd:documentation>
                                </xsd:annotation>
                           </xsd:element>

                           <xsd:element name="Url" minOccurs="1" nillable="false" type="xsd:string">
                                 <xsd:annotation>
                                     <xsd:documentation>Client requested url</xsd:documentation>
                                 </xsd:annotation>
                           </xsd:element>

                           <xsd:element name="CustomerId" type="xsd:string">
                               <xsd:annotation>
                                   <xsd:documentation>Customer Identifier</xsd:documentation>
                               </xsd:annotation>
                           </xsd:element>

                           <xsd:element name="CDNName" type="xsd:string">
                               <xsd:annotation>
                                   <xsd:documentation>CDN Name</xsd:documentation>
                               </xsd:annotation>
                           </xsd:element>

                           <xsd:element name="SignUrl" minOccurs="1" nillable="false" type="xsd:boolean">
                               <xsd:annotation>
                                   <xsd:documentation>Generate url signature</xsd:documentation>
                               </xsd:annotation>
                           </xsd:element>

                       </xsd:sequence>
                  </xsd:complexType>
               </xsd:element>
               <!-- UrlTranslationResponse [TranslatedUrl string] -->
               <xsd:element name="UrlTranslationResponse">
                    <xsd:complexType>
                         <xsd:annotation>
                              <xsd:documentation>Response with the translated url</xsd:documentation>
                         </xsd:annotation>
                         <xsd:sequence>
```

```
                              <xsd:element name="TranslatedUrl" minOccurs="1" nillable="false"
type="xsd:string">
                                   <xsd:annotation>
                                        <xsd:documentation>Translated CDN Url</xsd:documentation>
                                   </xsd:annotation>
                              </xsd:element>
                              <xsd:element name="SignedUrl" minOccurs="1" nillable="false" type="xsd:boolean">
                                   <xsd:annotation>
                                        <xsd:documentation>Signed Url</xsd:documentation>
                                   </xsd:annotation>
                              </xsd:element>
                         </xsd:sequence>
                    </xsd:complexType>
               </xsd:element>

        </xsd:schema>
    </wsdl:types>

    <wsdl:message name="UrlTranslationRequest">
       <wsdl:part name="parameters" element="utxsd:UrlTranslationRequest"/>
    </wsdl:message>

    <wsdl:message name="UrlTranslationResponse">
       <wsdl:part name="parameters" element="utxsd:UrlTranslationResponse"/>
    </wsdl:message>

    <wsdl:portType name="UrlTranslationPortType">
       <wsdl:operation name="GetUrlTranslation">
          <wsdl:input message="ut:UrlTranslationRequest"/>
          <wsdl:output message="ut:UrlTranslationResponse"/>
       </wsdl:operation>
    </wsdl:portType>

    <wsdl:binding name="UrlTranslationBinding" type="ut:UrlTranslationPortType">
      <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="GetUrlTranslation">
            <soap:operation soapAction="" style="document"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
      </wsdl:binding>

      <wsdl:service name="CDNUrlTranslationService">
        <wsdl:documentation>CDN Url Translation Service</wsdl:documentation>
        <wsdl:port name="UrlTranslationPort" binding="ut:UrlTranslationBinding">
            <soap:address location="http://localhost/CDNUrlTranslationService/"/>
        </wsdl:port>
      </wsdl:service>
</wsdl:definitions>
```

Proximity Engine SOAP APIs

This chapter provides an overview of the Proximity Engine, describes the SOAP (Simple Object Access Protocol) APIs exposed by the Proximity Engine, and presents the Web Service Definition Language (WSDL) file used by Proximity Engine to define proximity services. This chapter contains the following sections:

- Routing Concepts and Overview, page 5-1
- Proximity SOAP API Actions, page 5-2
- Proximity Engine WSDL File, page 5-7

Routing Concepts and Overview

You should be familiar with the basics of IP routing and routing protocols, such as Open Shortest Path First (OSPF), Intermediate System-to-Intermediate System (IS-IS), and Border Gateway Protocol (BGP). Each Proximity Engine operates in an IP routing domain where the Interior Gateway Protocol (IGP) or BGP is used to distribute routing information across the domain.

Routers running OSPF or IS-IS establish adjacencies with their directly connected neighbors and exchange their connectivity view (that is, each router advertises its visibility about its adjacencies). Advertisements are flooded throughout the whole routing area and each router stores each received advertisement in a link-state database (LSDB).

The LSDB contains the topology of the whole network and each router uses it in order to compute the Shortest Path Tree and the Routing Information Base (RIB) that contains each known IP prefix in the network and its corresponding next-hop.

Each Proximity Engine leverages the LSDB in order to deliver a proximity service to its clients (Service Routers [SRs]). In order to build the LSDB, the Proximity Engine establishes adjacencies with routers running IGP. In the absence of Proximity Engine IGP peering and an LSDB, the Proximity Engine can still leverage the BGP attributes to deliver a proximity service.

Terminology

An SR sends a proximity request to a Proximity Engine. The proximity request specifies a source IP address and a set of one or more target IP addresses. The following terminology is used for these items:

- *Proximity source address* (PSA)—IP address from which the proximity needs to be computed.
- *Proximity target address* (PTA)—IP address to which (from the PSA) proximity has to be computed.
- *Proximity target list* (PTL)—List of PTAs that need to be evaluated (that is, the proximity from each of these proximity target addresses to the proximity source address needs to be computed).
- *Ranking depth*—Integer number that determines the length of the ranking list. For example, an SR can request the proximity of 10 nodes out of a PTL of 20 IP addresses.

The scope of the proximity service is to determine the distance between two IP addresses in a routing area. The SR requests the Proximity Engine to rank a list of IP addresses (PTL) based on the individual distance of each PTA from the PSA.

The Proximity function takes into account:

- Routing topology
- Inter-Autonomous System (AS) reachability
- Optimal path taken by the requested data

Proximity SOAP API Actions

Note The Proximity Engine APIs are available on the CDE205 and CDE220-2G2 platforms.

The Proximity Engine exposes a SOAP interface on port 7003. The SOAP interface of the Proximity Engine implements a rate operation to calculate the proximity rating of a group of PTAs. For additional information on the services of the SOAP interface, see the "Proximity Engine WSDL File" section on page 5-7.

Note The SOAP API and associated port 7003 are only available when proximity-based routing is enabled on the SR.

Note The Proximity Engine implements a legacy group operation to return grouping information for a PSA. This operation may be required in a future release, but is not required in the current release because the rate operation returns grouping information for the PSA and the PTAs. The group operation is not specified in this document.

rate API

Requests the Proximity Engine to calculate the proximity of a list of PTAs to a PSA.

Request

The following are the input parameters of the rate request:

- _client—String representing the IP address of the PSA in the format of A.B.C.D.
- _destinations—PTL representing the list of PTAs.
- _category—Type of proximity service requested. Currently, the Proximity Engine only supports network-routing-based proximity which is represented by a value of 2.
- _resultCount—Number of results (that is, PTAs) the client requests to be returned. It refers to Ranking Depth concept described in the "Routing Concepts and Overview" section on page 5-1.

The rate operation is invoked by sending the following input message:

```
struct ns1__rate
{
        char *_client;
        struct ns2__ArrayOfJavaLangstring_USCOREliteral *_destinations;
        int _category;
        int _resultCount;
};
```

The array pointed to by the '_destinations' parameter is represented by an array in the following format:

```
struct ns2__ArrayOfJavaLangstring_USCOREliteral
{
        int __sizeJavaLangstring;
        char **JavaLangstring;
};
```

Request Example

The following section shows an example of a rate request:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:prox="http://com/cisco/topos/proximity" xmlns:java="java:com.cisco.topos.proximity">
   <soapenv:Header/>
   <soapenv:Body>
      <prox:rate>
         <client>209.165.201.1</client>
         <destinations>
            <!--Zero or more repetitions:-->
            <java:JavaLangstring>209.165.201.10</java:JavaLangstring>
            <java:JavaLangstring>209.165.201.8</java:JavaLangstring>
            <java:JavaLangstring>209.165.201.11</java:JavaLangstring>
            <java:JavaLangstring>209.165.201.16</java:JavaLangstring>
            <java:JavaLangstring>209.165.201.5</java:JavaLangstring>
            <java:JavaLangstring>209.165.201.7</java:JavaLangstring>
         </destinations>
         <category>2</category>
         <resultCount>100</resultCount>
      </prox:rate>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

The following are the output parameters of the rate response:

- Address—IP address of the PTA.
- Masklen—Prefix length of the subnet to which the PTA IP Address belongs.
- GroupId (optional)—Configured group for each IP address. For example, AS number, Masklen, or a community. Currently, the Proximity Engine does not return an Group Id.
- Rating—Proximity rating of the PTA based on the proximity algorithm.

The data structure of the rate response is as follows:

```
struct ns1__rateResponse
{
        struct ns2__ArrayOfServiceRatedAddress_USCOREliteral *_RatedAddressArray;
};
```

The ranked PTAs are returned in the ServiceRatedAddress array pointed to by the RatedAddressArray in the following format:

```
struct ns2__ArrayOfServiceRatedAddress_USCOREliteral
{
        int __sizeServiceRatedAddress;
        struct ns2__ServiceRatedAddress *ServiceRatedAddress;
};
```

Note The first member of the ServiceRatedAddress array is the PSA.

Each member of the ServiceRatedAddress is returned in the following format:

```
struct ns2__ServiceRatedAddress
{
        char *Address;
        int Masklen;
        LONG64 *GroupId;
        int Rating;
};
```

The array of PTAs is returned in ranked order of ascending *rating* with the PSA being the first member of the array.

Response Example

The following section shows an example of a rate response:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:ns2="java:com.cisco.topos.proximity"
xmlns:ns1="http://com/cisco/topos/proximity">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <ns1:rateResponse>
         <RatedAddressArray>
            <ns2:ServiceRatedAddress>
               <ns2:Address>209.165.201.1</ns2:Address>
               <ns2:Masklen>32</ns2:Masklen>
               <ns2:Rating>0</ns2:Rating>
            </ns2:ServiceRatedAddress>
            <ns2:ServiceRatedAddress>
               <ns2:Address>209.165.201.5</ns2:Address>
               <ns2:Masklen>32</ns2:Masklen>
```

```
                  <ns2:Rating>10</ns2:Rating>
               </ns2:ServiceRatedAddress>
               <ns2:ServiceRatedAddress>
                  <ns2:Address>209.165.201.7</ns2:Address>
                  <ns2:Masklen>32</ns2:Masklen>
                  <ns2:Rating>10</ns2:Rating>
               </ns2:ServiceRatedAddress>
               <ns2:ServiceRatedAddress>
                  <ns2:Address>209.165.201.8</ns2:Address>
                  <ns2:Masklen>20</ns2:Masklen>
                  <ns2:Rating>20</ns2:Rating>
               </ns2:ServiceRatedAddress>
               <ns2:ServiceRatedAddress>
                  <ns2:Address>209.165.201.10</ns2:Address>
                  <ns2:Masklen>129</ns2:Masklen>
                  <ns2:Rating>70</ns2:Rating>
               </ns2:ServiceRatedAddress>
               <ns2:ServiceRatedAddress>
                  <ns2:Address>209.165.201.11</ns2:Address>
                  <ns2:Masklen>129</ns2:Masklen>
                  <ns2:Rating>70</ns2:Rating>
               </ns2:ServiceRatedAddress>
               <ns2:ServiceRatedAddress>
                  <ns2:Address>209.165.201.16</ns2:Address>
                  <ns2:Masklen>129</ns2:Masklen>
                  <ns2:Rating>70</ns2:Rating>
               </ns2:ServiceRatedAddress>
            </RatedAddressArray>
         </ns1:rateResponse>
      </SOAP-ENV:Body>
   </SOAP-ENV:Envelope>
```

Fault Message

The gSOAP server supports a number of faults, each representing a different error scenario. In most cases, a fault is simply a string describing a fault condition. Table 5-1 presents the list of faults specific to the Proximity Engine and provides an explanation for each fault.

Table 5-1 Proximity Engine Faults

Proximity Engine Fault	Explanation
Proximity Service Failed: Urib[1] failed to send request to protocol	There is a communication problem between the URIB and the IGP or BGP daemons.
Proximity Service Failed: Urib failed to receive from protocol	There is a communication problem between the URIB and the IGP or BGP daemons.
Proximity Service Failed: Route lookup failed in URIB	The PSA cannot be resolved by URIB. That is, the PSA cannot be found in the IP routing table.
Proximity Service Failed: Unrecognized exception	An unknown error has occurred.

1. URIB = unicast routing information base

Fault Response

The data structures of a SOAP fault response are as follows:

```
/* SOAP Fault Code: */
```

```
struct SOAP_ENV__Code
{
        char *SOAP_ENV__Value;
        struct SOAP_ENV__Code *SOAP_ENV__Subcode;
};

/* SOAP-ENV:Detail */
struct SOAP_ENV__Detail
{
        int __type;
        void *fault;            char *__any;
};
/* SOAP-ENV:Reason */
struct SOAP_ENV__Reason
{
        char *SOAP_ENV__Text;
};
/* SOAP Fault: */
struct SOAP_ENV__Fault
{
        char *faultcode;
        char *faultstring;
        char *faultactor;
        struct SOAP_ENV__Detail *detail;
        struct SOAP_ENV__Code *SOAP_ENV__Code;
        struct SOAP_ENV__Reason *SOAP_ENV__Reason;
        char *SOAP_ENV__Node;
        char *SOAP_ENV__Role;
        struct SOAP_ENV__Detail *SOAP_ENV__Detail;
};
```

A redirect fault occurs when the Proximity Engine does not consider itself the most appropriate Proximity Engine to service the request, and redirects the proximity client to a set of Proximity Engines it considers more appropriate.

Redirect Response

The Proximity Engine uses the following redirect response data structure wrapped inside the SOAP_ENV_FAULT data structure to send a redirect fault:

```
struct ns2__RedirectResponse
{
/// @brief PSA grouping info.
/// Element psa of type java:com.cisco.topos.proximity":ServiceGroupRange.
    struct ns2__ServiceGroupRange*       psa                             1;  ///< Required
element.
/// @brief SG Endpoints for the redirect.
/// Element EndPoints of type "java:com.cisco.topos.proximity":EndPoints.
    struct ns2__EndPoints*               EndPoints                       1;  ///< Required
element.
};
```

The Proximity Engine returns as part of this fault the group information of the PSA and the list of EndPoints. The data structure of the PSA group information is as follows:

struct ns2__ServiceGroupRange

```
{
        char *Address;
        int Masklen;
        LONG64 *GroupId;
};
```

The Proximity Engine also returns as part of this fault the list of EndPoints containing the list of IP addresses of other Proximity Engines in the network that are more appropriate for this proximity request. The data structure of the list of EndPoints is as follows:

```
struct _ns2__EndPoints
{
        int __sizeEndPoint;
        char **EndPoint;
};
```

Proximity Engine WSDL File

A WSDL file is an XML-formatted file that describes the details as a web service and specifies the operations it knows how to perform. The WSDL file for the SOAP interface of the Proximity Engine is as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<s0:definitions name="NetworkProximityServiceDefinitions"
targetNamespace="http://com/cisco/topos/proximity" xmlns=""
xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="http://www.w3.org/2001/XMLSchema"
xmlns:s2="java:com.cisco.topos.proximity" xmlns:s3="http://com/cisco/topos/proximity"
xmlns:s4="http://schemas.xmlsoap.org/wsdl/soap/">
  <s0:types>
    <xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="java:com.cisco.topos.proximity"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:complexType name="ServiceRatedAddress">
          <xs:annotation>
            <xs:documentation>Rated IP address.</xs:documentation>
          </xs:annotation>
          <xs:sequence>
              <xs:element minOccurs="1" name="Address" nillable="false" type="xs:string">
                <xs:annotation>
                  <xs:documentation>Get IP address.</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element minOccurs="1" name="Masklen" nillable="false" type="xs:int">
                <xs:annotation>
                  <xs:documentation>Get subnet masklen.</xs:documentation>
                </xs:annotation>
              </xs:element>
                <xs:element minOccurs="0" name="GroupId" nillable="false" type="xs:long">
                <xs:annotation>
                  <xs:documentation>Get group Id.</xs:documentation>
                </xs:annotation>
              </xs:element>
            <xs:element minOccurs="1" name="Rating" nillable="false" type="xs:int">
              <xs:annotation>
                <xs:documentation>Get rating.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      <xs:complexType name="ServiceGroupRange">
        <xs:annotation>
          <xs:documentation>Group Id with addresss range.</xs:documentation>
        </xs:annotation>
        <xs:sequence>
          <xs:element minOccurs="1" name="Address" nillable="false" type="xs:string">
            <xs:annotation>
```

```
              <xs:documentation>Get IP address.</xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element minOccurs="1" name="Masklen" nillable="false" type="xs:int">
            <xs:annotation>
              <xs:documentation>Get subnet masklen.</xs:documentation>
            </xs:annotation>
          </xs:element>
            <xs:element minOccurs="0" name="GroupId" nillable="false" type="xs:long">
            <xs:annotation>
              <xs:documentation>Get group Id.</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="psa" type="java:ServiceGroupRange"
xmlns:java="java:com.cisco.topos.proximity">
            <xs:annotation>
                <xs:documentation>PSA grouping info</xs:documentation>
            </xs:annotation>
      </xs:element>
        <xs:complexType name="EndPoints">
            <xs:annotation>
                <xs:documentation>SG EndPoints for the redirect</xs:documentation>
            </xs:annotation>
            <xs:sequence>
                <xs:element name="EndPoint" maxOccurs="unbounded" minOccurs="1"
nillable="false" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
        <xs:complexType name="RedirectResponse">
          <xs:annotation>
            <xs:documentation>Redirect response.</xs:documentation>
          </xs:annotation>
          <xs:sequence>
              <xs:element maxOccurs="1" minOccurs="1" name="psa" nillable="false"
              type="java:ServiceGroupRange"
              xmlns:java="java:com.cisco.topos.proximity">
                <xs:annotation>
                  <xs:documentation>PSA grouping info.</xs:documentation>
                </xs:annotation>
              </xs:element>
              <xs:element maxOccurs ="1" minOccurs="1" name="EndPoints" nillable="false"
              type="java:EndPoints" xmlns:java="java:com.cisco.topos.proximity">
                <xs:annotation>
                  <xs:documentation>SG Endpoints for the redirect.</xs:documentation>
                </xs:annotation>
              </xs:element>
          </xs:sequence>
        </xs:complexType>
      <xs:complexType name="ArrayOfServiceRatedAddress_literal">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="ServiceRatedAddress"
nillable="true" type="java:ServiceRatedAddress"
xmlns:java="java:com.cisco.topos.proximity"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="ArrayOfServiceRatedAddress_literal"
type="java:ArrayOfServiceRatedAddress_literal"
xmlns:java="java:com.cisco.topos.proximity"/>
      <xs:complexType name="ArrayOfServiceGroupRange_literal">
        <xs:sequence>
```

```
          <xs:element maxOccurs="unbounded" minOccurs="0" name="ServiceGroupRange"
nillable="true" type="java:ServiceGroupRange"
xmlns:java="java:com.cisco.topos.proximity"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="ArrayOfServiceGroupRange_literal"
type="java:ArrayOfServiceGroupRange_literal" xmlns:java="java:com.cisco.topos.proximity"/>
      <xs:complexType name="ArrayOfJavaLangstring_literal">
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="JavaLangstring"
nillable="true" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
      <xs:element name="ArrayOfJavaLangstring_literal"
type="java:ArrayOfJavaLangstring_literal" xmlns:java="java:com.cisco.topos.proximity"/>
    </xs:schema>
  </s0:types>
  <s0:message name="rate">
    <s0:part name="client" type="s1:string"/>
    <s0:part name="destinations" type="s2:ArrayOfJavaLangstring_literal"/>
    <s0:part name="category" type="s1:int"/>
    <s0:part name="resultCount" type="s1:int"/>
  </s0:message>
  <s0:message name="rateResponse">
    <s0:part name="RatedAddressArray" type="s2:ArrayOfServiceRatedAddress_literal"/>
  </s0:message>
  <s0:message name="group">
    <s0:part name="addresses" type="s2:ArrayOfJavaLangstring_literal"/>
  </s0:message>
  <s0:message name="groupResponse">
    <s0:part name="GroupRangeArray" type="s2:ArrayOfServiceGroupRange_literal"/>
  </s0:message>
  <s0:portType name="NetworkProximityPortType">
    <s0:operation name="rate" parameterOrder="client destinations category resultCount">
      <s0:input message="s3:rate"/>
      <s0:output message="s3:rateResponse"/>
    </s0:operation>
    <s0:operation name="group" parameterOrder="addresses">
      <s0:input message="s3:group"/>
      <s0:output message="s3:groupResponse"/>
    </s0:operation>
  </s0:portType>
  <s0:binding name="NetworkProximityServiceSoapBinding"
type="s3:NetworkProximityPortType">
    <s4:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <s0:operation name="rate">
      <s4:operation soapAction="" style="rpc"/>
      <s0:input>
        <s4:body namespace="http://com/cisco/topos/proximity" parts="client destinations
category resultCount" use="literal"/>
      </s0:input>
      <s0:output>
        <s4:body namespace="http://com/cisco/topos/proximity" parts="RatedAddressArray"
use="literal"/>
      </s0:output>
    </s0:operation>
    <s0:operation name="group">
      <s4:operation soapAction="" style="rpc"/>
      <s0:input>
        <s4:body namespace="http://com/cisco/topos/proximity" parts="addresses"
use="literal"/>
      </s0:input>
      <s0:output>
```

```
        <s4:body namespace="http://com/cisco/topos/proximity" parts="GroupRangeArray"
use="literal"/>
      </s0:output>
    </s0:operation>
  </s0:binding>
  <s0:service name="NetworkProximityService">
    <s0:port binding="s3:NetworkProximityServiceSoapBinding"
name="NetworkProximityServicePort">
      <s4:address location="http://localhost:7003/"/>
    </s0:port>
  </s0:service>
</s0:definitions>
```

Program Files in the VDS Software

VDS software uses programs to enable support for live multicast and scheduled rebroadcast events. A program in the VDS software is defined as a scheduled event in which the content is presented to the end user. The three attributes of a program are:

- Schedule—Defines when the content is presented to the end user.
- Content—Defines what is presented to the end user. in the VDS software, this can be pre-positioned or live content.
- Presentation—Defines how the content is presented to the end user. The presentation attributes include the set of Service Engines that know about the program, and a service type that identifies the streaming server used to deliver the content. The streaming server can exist in the Service Engine (Windows Media Technology [WMT] or Movie Streamer).

A program file contains the elements that define the schedule, content, and presentation parameters. It is a text file written in XML format, similar to the Manifest file. For more information about Manifest files, refer to *Cisco VDS Internet Streamer 3.3 Software Configuration Guide*. See the "Related Publications" section on page xiii for links to documentation online.

Program types determine the hardware or software component involved in delivering content to the user. Different program types are:

- Movie Streamer
- WMT

The CDSM manages multicast addresses to be used for programs. Each Service Engine assigned to the program uses the multicast address for broadcast. The Service Engine determines which multicast address is to be used based on the program data. A set of multicast addresses can be specified either in the Program API or by using the CDSM. Each time a program requires a multicast address, the CDSM associates one of the addresses with the program. Addresses are allocated for the life of a program. Programs can be configured with an auto-delete feature, which allows program addresses to be freed up automatically about 24 hours after a program schedule is complete.

When you request a specific address or a set of addresses to be used for a program, VDS software issues only those addresses that are not used by any of the existing programs. You receive an error message if there is no multicast address associated with the imported program file and no addresses are available to be configured from the pool or if the multicast pool has not been configured.

When you define a Movie Streamer live program using the createProgram API, you can specify a single backup broadcast server for the program. To do this, you must specify the IP addresses of the primary and backup broadcast servers in the program file using the <media> tag. The <media> tag in the program file should be in the following format:

<media index="*number*" src="*primary_broadcast_server*:*port*;*backup_broadcast_server*:*port*"/>

Program File DTD

The following is the Document Type Definition (DTD) for VDS program files. You can use the DTD to create program files for importing programs from third-party systems.

```
<?xml version="1.0"?>
<!DOCTYPE program[
  <!ELEMENT program (media*, ucastInfo?, mcastInfo?, schedule*, attribute?)>
  <!ATTLIST program
       version               CDATA "1.0"
       name                  CDATA #REQUIRED
       serviceType           (wmt | movieStreamer) "wmt"
       description           CDATA #IMPLIED
       playTime              CDATA #IMPLIED
       lastModificationTime  CDATA #IMPLIED
       gracefulExit          (false | true) "false"
       shuffle               (false | true) "false"
       autoDelete            (false | true | default) "default"
       blockPerSchedule      (false | true) "false"
       live                  (false | true) "false"
  >
  <!ELEMENT media EMPTY>
  <!ATTLIST media
       index                 CDATA #IMPLIED
       src                   CDATA #REQUIRED
       id                    CDATA #IMPLIED
       playTime              CDATA #IMPLIED
>
  <!ELEMENT ucastInfo EMPTY>
  <!ATTLIST ucastInfo
       referenceUrl          CDATA #REQUIRED
  >
  <!ELEMENT mcastInfo (addrPort*)>
  <!ATTLIST mcastInfo
       referenceUrl          CDATA #REQUIRED
       TTL                   CDATA #IMPLIED
  >
  <!ELEMENT addrPort EMPTY>
  <!ATTLIST addrPort
       addrTypeVal           (ipv4 | ipv6) "ipv4"
       addrVal               CDATA #REQUIRED
       portVal               CDATA #REQUIRED
       id                    CDATA #IMPLIED
  >
<!ELEMENT schedule (repeats?)>
  <!ATTLIST schedule
       timeSpec        (local | gmt) "local"
       startTime       CDATA #IMPLIED
       activeDuration  CDATA #IMPLIED
  >
  <!ELEMENT repeats (dayOffset*)>
  <!ATTLIST repeats
       type            (timeInterval | days) "timeInterval"
       interval        CDATA #IMPLIED
       endTime         CDATA #IMPLIED
  >
  <!ELEMENT dayOffset EMPTY>
  <!ATTLIST dayOffset
       value        (0 | 1 | 2 | 3 | 4 | 5 | 6) "0"
  >
  <!ELEMENT attribute EMPTY>
  <!ATTLIST attribute
       value         CDATA #REQUIRED
```

```
  >
]>
```

Table A-1 describes the elements in the DTD and their attributes.

Table A-1 Program File DTD Elements and Attributes

Element	Attributes	Description
program	version	Version of the program file. VDS software generates playlist files with a version level of 1.
	name	Name of the program.
	serviceType	Type of program, which dictates the mode of delivery. This element identifies the software or hardware component involved in delivering the content to the user.
	description	Brief description of the program.
	playTime	Total playtime in seconds. This is the sum of the playtime values of the media files, if set. If there are files in the program that have invalid playtimes, then this field is set to –1.
	lastModificationTime	Time when the playlist was created or modified last, as recorded in the CDSM. The format is hh:mm:ss. The assumption is that all devices in the VDS network are time-synchronized (for example, using the NTP[1].
	gracefulExit	Specifies how to handle scheduled exits. Options are: • True—Exit after the current media file is played completely. • False—Exit immediately.
	shuffle	Specifies whether the media files should play in any order. Options are: • True—Play media files at random. • False—Play media files in order. When this attribute is not specified, it is set to false by default.
	autoDelete	Specifies whether the program should be automatically deleted 24 hours after it is last played. Options are: • True—Delete the program 24 hours after it is last played. • False—Retain the program for more than 24 hours after it is last played. • Default—When the value for the live attribute is set to true, the default value is true for *autoDelete*, and false if the live attribute is set to false.

Table A-1 Program File DTD Elements and Attributes (continued)

Element	Attributes	Description
	blockPerSchedule	Specifies whether active streams should be terminated when the scheduled program ends. This attribute is used only when a live unicast event is scheduled to be delivered by the WMT streaming server. Options are: • True—WMT terminates active streams when the scheduled program ends. • False—WMT does not terminate active streams when the scheduled program ends. When this attribute is not specified, it is set to false by default.
	live	Specifies whether the program contains live content. Options are: • True—The program contains live content. • False—The program does not contain live content.
media	index	Order of the media file in the list of files, ranging from 1 to mediaCount (the number of media files in the program). The index attribute specifies the order of the media files when the shuffle attribute in the <media> tag is set to false.
	src	Reference to the source of the media file. • For live content, this field contains information about how the streaming server will correlate with the live feed. • For prefetched content, this field contains the portion of the URL that follows the origin server; that is, the FQDN[2]. For example, if the source file URL is http://mycontentorigin/mydirectory/myfile, the value assigned to this field is mydirectory/myfile. **Note** When prefetched content is exported, this field contains the URL for the file that can be routed in the VDS network, without the protocol specification. • Live source failover is supported. For WMT live, multiple encoders or streaming servers can be specified. src="http://encoder_1:8080;rtsp://source_hostip/filename" For Movie Streamer live, only a single backup can be specified. src="sourceaddress1:destinationport1;sourceaddress2:destinationport2"
	id	Media file identifier. For WMT rebroadcast events, this field contains the ID of the delivery service containing this media file. For Movie Streamer rebroadcast events, this field contains the track number. In the case of live events, this field is used to correlate a stream source with a multicast address. **Note** For live unicast programs, do not include the ID attribute.

Table A-1 Program File DTD Elements and Attributes (continued)

Element	Attributes	Description
	playTime	Playtime for the file in seconds, when it is known. This attribute is used only for MPG media files. Options are: • –2—If the file is not an MPG file • –1—If the file is an MPG file but the VDS software cannot determine the playtime • 0 or greater—If the playtime is correctly determined from the file
ucastInfo	referenceUrl	URL used by the end user to request this program over the network using unicast. **Note** All letters in the reference URLs must be in lowercase.
mcastInfo	referenceUrl	URL used by the end user to request this program over the network using multicast. **Note** All letters in the reference URLs must be in lowercase.
	TTL	Multicast TTL[3] value to be used for the packets sent using multicast.
addrPort	addrval	Address to be used when this program is multicast.
	portVal	Port (within the multicast address) to be used when this program is multicast.
	id	Address and port identifier. For rebroadcast events, this field contains the ID of the delivery service using this address and port. In the case of live events, this field is used to correlate a stream source with a multicast address.
schedule	timeSpec	Specifies how time values should be interpreted. Options are: • Local • GMT[4]
	startTime	Time (in seconds) since the epoch (January 1, 1970) when the program should start playing. **Tip** For UNIX operating systems, the epoch is 00:00:00 GMT, January 1, 1970. This represents the time and date corresponding to 0 in the UNIX operating system's date and time stamp. System time is measured in seconds past the epoch.
	activeDuration	Duration of the program (in seconds). For a scheduled rebroadcast, this value specifies how long the files should loop (that is, loop for *x* seconds). If there is no looping, this value is 0. For live programs, this value is the duration of the event.
repeats	type	Type of repeat. For example, you can set the program to repeat every *x* seconds, or repeat on specified days of the week at the same time specified in the start time. Options are: • TimeInterval • Days

Table A-1 Program File DTD Elements and Attributes (continued)

Element	Attributes	Description
	interval	Time interval (in seconds) for the repeat broadcast of the program. For example, if this value to 28800 seconds, the program repeats every 8 hours.
	endTime	Time (in seconds) since the epoch (January 1, 1970) when program repeats should end. For a program that repeats forever, enter the value zero (0).
dayOffset	value	Day to repeat the program, for example, every Monday. The time (during the day) of the repeat is inherited from the startTime attribute.
attribute	value	Element used if a third-party device is used to import some data that is transparent to a VDS network, and that is directly used by the software or hardware component involved in delivering the content to the user. The CMS[5] relays the data without interpreting it. A recommended method for encoding this field is to use a name/value pair in the string, for example, name1=value1; name2=value2.

1. NTP = Network Time Protocol.
2. FQDN = fully qualified domain name.
3. TTL = time-to-live.
4. GMT = Greenwich Mean Time.
5. CMS = Centralized Management System.

Program File Examples

This section contains program file examples, each describing the contents for specific event types. The examples are provided for the following event types:

- WMT Multicast Live Event, page A-6
- WMT Multicast Rebroadcast Event, page A-7
- Movie Streamer Multicast Event, page A-7
- Movie Streamer Live-Split Event, page A-8

WMT Multicast Live Event

The following example shows the program file for a WMT multicast live event in which the multicast address is specified using the addrPort element:

```
<?xml version="1.0"?>
<!DOCTYPE program SYSTEM "program.dtd">
<program version="1.0" name="liveProgram" serviceType="wmt" description="test"
autoDelete="true" blockPerSchedule="true" live="true">
<media index="1" src="http://WMT_encoder:8080" id="media0"/>
<mcastInfo referenceUrl="http://contentacquirer/liveprogram.nsc" TTL="22">
<addrPort addrVal="239.232.25.95" portVal="61248" id="media0"/>
</mcastInfo>
<schedule timeSpec="gmt" startTime="0" activeDuration="0"/>
</program>
```

WMT Multicast Rebroadcast Event

This example shows the program file for a WMT multicast rebroadcast event:

```
<?xml version="1.0"?>
<!DOCTYPE program SYSTEM "program.dtd">
<program version="1.0" name="chanrebroad" serviceType="wmt" description="test"
autoDelete="false" live="false">
<media index="1" src="sen/beck.asf" id="Channel_35748"/>
<media index="2" src="sen/CSCin53585.wmv" id="Channel_35748"/>
<media index="3" src="sen/starsnstripes.asf" id="Channel_35749"/>
<mcastInfo referenceUrl="http://contentacquirer/chanrebroad.nsc">
<addrPort addrVal="239.232.25.195" portVal="61248" id="Channel_35748"/>
</mcastInfo>
<schedule timeSpec="local" startTime="1010170800" activeDuration="1300">
<repeats type="timeInterval" interval="2600"/>
</schedule>
</program>
```

The *referenceUrl* attribute is the link that the user clicks to join the program. You can provide the external IP address of the Content Acquirer (for example, http://ServiceEngine/prog1.nsc) in the *referenceUrl* attribute.

Note A media file can be uniquely identified using a URL of the form <protocol>://<FQDN>/<relative_URL>. The *id* attribute in the media element specifies the ID of the delivery service containing the media file. Each delivery service is associated with the FQDN of a Service Engine or that of an origin server. The *src* attribute in the media element provides the relative part of the URL, which along with the *id* attribute identifies the file.

You can provide the FQDN of the Service Engine that hosts the media file if a Service Router is used to direct the user request to the appropriate Service Engine. In this case, the FQDN must be associated with a website or delivery service that maps to the same Service Engines that can serve the program.

You can provide the name of the Service Engine if the user request goes to a preselected Service Engine. If a third-party device assigns the Service Engines directly to the program, you can use any one of the Service Engines assigned to the program in the *referenceUrl* attribute. If the third-party device assigns a delivery service to the program, you can use the name of any Service Engine in that delivery service (for example, the Content Acquirer) in the *referenceUrl* attribute.

Movie Streamer Multicast Event

This example shows the program file for a Movie Streamer multicast event. This event can also be accessed using unicast by specifying the *referenceUrl* attribute in the ucastInfo element.

```
<?xml version="1.0" ?>
<!DOCTYPE program SYSTEM "program.dtd">
<program version="1.0" name="prog5lfs_1673" serviceType="movieStreamer"
description="prog5lfs" playTime="3600" autoDelete="false" live="true">
<media index="1" src="source_ip_address:destination_port" id="media0"/>
<media index="2" src="source_ip_address:destination_port" id="media1"/>
<ucastInfo referenceUrl="rtsp://pm_fqdn_or_ip_addr/pm_1673.sdp"/>
<mcastInfo referenceUrl="http://pm_fqdn_or_ip_addr/programs/1673" TTL="15">
<addrPort addrVal="224.2.250.195" portVal="61036" id="media0"/>
```

```
<addrPort addrVal="224.2.250.196" portVal="61038" id="media1"/>
</mcastInfo>
<schedule timeSpec="gmt" startTime="3264429600"/>
<attribute value="unicastPushSDP:http://2.43.12.6/programs/1673"/>
</program>
```

Note The media source (src) is the live feed. The *src* attribute contains the IP address of the Broadcast Server and the destination port of the Content Acquirer. The Content Acquirer listens for the program stream on the specified destination port. There is more than one media source, because audio, video, and other feeds may be broadcast on a separate stream, using a separate multicast address. The *id* attribute in the media element and the *id* attribute in the addrPort element are used to correlate the address to the stream.

Movie Streamer Live-Split Event

This example shows the program file for a Movie Streamer live-split event:

```
<?xml version="1.0" ?>
<!DOCTYPE program SYSTEM "program.dtd">
<program version="1.0" name="prog5lfs_1674" serviceType="movieStreamer"
description="prog52fs" playTime="3600" autoDelete="false" live="true">
<media index="1" src="source_ip_address:destination_port" />
<media index="2" src="source_ip_address:destination_port" />
<ucastInfo referenceUrl="rtsp://pm_fqdn_or_ip_addr/pm_1674.sdp"/>
<schedule timeSpec="gmt" startTime="3264429600" activeDuration="7200"/>
<attribute value="unicastPushSDP:http://2.43.12.6/programs/1673"/>
</program>
```

Note Attributes for the schedule element must be specified for the Movie Streamer streaming server. The *id* attribute is not required because there are no separate multicast addresses for the program streams.