



Cisco VDS Service Broker 1.3 API Guide

November 1, 2013

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.

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco VDS Service Broker 1.3 API Guide

© 2013 Cisco Systems, Inc. All rights reserved.



Preface	v
Document Revision History	v
Audience	v
Document Organization	vi
Document Conventions	vi
Related Publications	vii
Obtaining Documentation and Submitting a Service Request	vii

CHAPTER 1

Introduction to Cisco VDS Service Broker APIs	1-1
RESTful APIs	1-1
Authentication and Authorization	1-2
ID Format	1-2
Calling the HTTPS 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

CHAPTER 2

VDS Service Broker APIs	2-1
CDN Management API	2-1
CDN List and Definition	2-1
CDN Adaptation Policy Settings	2-3
BFQDN Management API	2-5
BFQDN List and Definition	2-6
BFQDN-Service Broker Assignment	2-7
BFQDN Policy Settings	2-8
CDN Metrics Provider API	2-10
CDN Metrics Provider List and Definition	2-10
CDN Metrics Provider-BFQDN Assignment	2-11
BFQDN Policy API	2-12
CDN Adaptation Policy API	2-13
CDN Selection Policy API	2-15

Service Broker API	2-18
Service Broker List and Definition	2-18
Service Broker's BFQDN Assignment	2-19
File Management API	2-20
XML Schema for Policy API	2-25
CDNRules.xsd	2-25
CDNPolicies.xsd	2-35



Preface

This preface describes who should read the *Cisco VDS Service Broker 1.3 API Guide*, how it is organized, and its document conventions. It contains the following sections:

- [Audience, page v](#)
- [Document Organization, page vi](#)
- [Document Conventions, page vi](#)
- [Related Publications, page vii](#)
- [Obtaining Documentation and Submitting a Service Request, page vii](#)

Document Revision History

This 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
OL-29910-01	June 2013	Initial release
OL-30254-01	July 2013	Versioned up from 1.1 to 1.2
OL-30254-02	November 2013	Versioned up from 1.2 to 1.3

Audience

This application program interface (API) guide is written for the knowledgeable application programmer who understands the basic architecture of the VDS Service Broker 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 Service Broker software servlets are used.

Document Organization

This API guide includes the following chapters:

Chapter or Appendix	Description
Chapter 1, “Introduction to Cisco VDS Service Broker APIs”	Provides an introduction to the VDS-SB software application program interfaces.
Chapter 2, “VDS Service Broker APIs”	Describes the APIs for VDS-SB.

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 .
<i>italic font</i>	Variables for which you supply values are in <i>italics</i> . Directory names and filenames are also in italics.
screen font	Terminal sessions and information the system displays are printed in <code>screen font</code> .
boldface screen font	Information you must enter is in boldface screen font .
<i>italic screen font</i>	Variables you enter are printed in <i>italic screen font</i> .
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 Service Broker and are available from the Cisco.com site:

- *Cisco VDS Service Broker 1.3 Software Configuration Guide*
- *Cisco VDS Service Broker 1.3 Command Reference*
- *Cisco VDS Service Broker 1.2 Alarms and Error Messages Guide*
- *Release Notes for Cisco VDS Service Broker 1.3*
- *Open Sources Used in VDS Service Broker Release 1.2*

You can access the software documents at the following URL:

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

Documentation for the UCS hardware platforms can be found 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
- *Cisco UCS B200 M3 Blade Server Installation and Service Note*
http://www.cisco.com/en/US/docs/unified_computing/ucs/hw/chassis/install/B200M3.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 VDS Service Broker APIs

Cisco Videoscape Distribution Suite Service Broker (VDS-SB) software 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.

This chapter contains the following sections:

- [RESTful APIs, page 1-1](#)
- [Calling the HTTPS APIs, page 1-2](#)
- [API Status Codes, page 1-3](#)

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.

VDSM 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 VDSM provides the following RESTful HTTPS APIs:

- CDN Management
- BFQDN Management
- BFQDN Policy
- CDN Adaptation Policy
- CDN Selection Policy
- Service Broker
- File Management

All URLs starting with “/api/” are RESTful APIs that are intercepted by the VDSM REST service.

Authentication and Authorization

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

```
https://admin:default@<vds_m_ip>:8443/api/CDN
```

For more information, see the *Cisco VDS SB 1.3 Software Configuration Guide* or the VDSM 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 CDN URL is `https://admin:default@<vds_m_ip>:8443/api/CDN/{id}`; so for a VDSM with an IP address of 192.168.1.25 and an CDN ID of 193, the URL is `https://admin:default@192.168.1.25:8443/api/BFQDNs/193`.



Note

The admin password, *default*, is the built-in admin password, and is used as an example. We strongly recommend that you change the built-in admin password as soon as possible. To do so, log in to the CLI of the VDSM 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 HTTPS APIs

You can execute the VDSM 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 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 HTTPS API requests:

- **Accept**—Accept header must be specified to indicate what content type the API client expects to receive from VDSM 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 VDSM API services, supported values are: `application/xml`, or `application/json`

Interactive Calls

Use a REST client plug-in for a browser or Lynx command to execute the API interactively. The browser address bar only supports the GET method; however, the VDSM 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 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.

API Status Codes

The VDSM return 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/LiveChannels/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/LiveChannels/22",  
  "error": "The requested object is not found: 22"  
}
```

Success Status Codes

Success status codes consist of 200 OK, 201 Created (with Location header: the URI of the new created resource) 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.



VDS Service Broker APIs

This chapter describes the HTTPS RESTful APIs for VDS-SB and the XML schema.

- [CDN Management API, page 2-1](#)
- [BFQDN Management API, page 2-5](#)
- [BFQDN Policy API, page 2-12](#)
- [CDN Metrics Provider API, page 2-10](#)
- [CDN Adaptation Policy API, page 2-13](#)
- [CDN Selection Policy API, page 2-15](#)
- [Service Broker API, page 2-18](#)
- [File Management API, page 2-20](#)
- [XML Schema for Policy API, page 2-25](#)

CDN Management API

The CDN API supports listing and provisioning of the CDN and has the following API calls:

- [CDN List and Definition](#)
- [CDN Adaptation Policy Settings](#)

CDN List and Definition

The CDN List and Definition part of the CDN Management API provides parity for the following VDSM GUI pages:

- **Services > Content Delivery Networks**
- **Services > Content Delivery Networks > Definition**

Table 2-1 Content Delivery Networks List and Definition API Calls

Resource URL	Method	Function Description
/api/CDNs	GET	Gets the list of all CDNs the API user has access to, which is determined by the Domain Management configured for the user.
	POST	Creates a new CDN. The VDSM allocates a long type of positive integer ID for the newly created CDN. The “Location” header in the response returns the URI of the created CDN, which includes the ID (for example, Location: /api/CDNs/235).
/api/CDNs/{id}	GET	Gets the definition settings of an CDN identified by {id}.
	PUT	Modifies the definition settings of an CDN identified by {id}.
	DELETE	Deletes an CDN identified by {id}.

GET Definition Response Example

Following is an example of a response to get CDNs with the ID 235.

```
<!-- Data Model (response): GET list of CDNs -->
<CDNs uri="/api/CDNs">
  <!-- To be noticed: password field is not shown in response -->
<CDNs-row id="235" uri="/api/CDNs/235">
<name>cdn1</name>
<!--0:cisco;1:velocix;2:generic-->
  <providerType>1</providerType>
<rFqdn>10.74.13.12</rFqdn>
<port>1099</port>
<!-- 0:in service; 1:out of service -->
<serviceState>0</serviceState>
  <carrier>6666</carrier>
  <cost>0.123</cost>
  <enableStatusTrack>1</enableStatusTrack>
<pollInterval>5</pollInterval>
<!--lowWatermark and highWatermark are not available for generic CDN-->
<lowWatermark>50</lowWatermark>
<highWatermark>90</highWatermark>
<!--snmpAgentIp, snmpPort and snmpCommunity are for Velocix CDN only-->
<snmpAgentIP>10.75.4.56</snmpAgentIP>
<snmpPort>9999</snmpPort>
<snmpCommunity>testsnmp</snmpCommunity>
<description>cdn1</description>
</CDNs-row >
<CDNs-row id="236" uri="/api/CDNs/236">
<name>cdn2</name>
<!--0:cisco;1:velocix;2:generic-->
  <providerType>0</providerType>
<rFqdn>10.74.13.42</rFqdn>
<port>1099</port>
<!-- 0:in service; 1:out of service -->
<serviceState>0</serviceState>
<carrier>6666</carrier>
<cost>0.123</cost>
<enableStatusTrack>1</enableStatusTrack>
<pollInterval>5</pollInterval>
<lowWatermark>50</lowWatermark>
<highWatermark>90</highWatermark>
```

```

<description>cdn2</description>
</CDNs-row >
<CDNs-row id="237" uri="/api/CDNs/237">
<name>cdn3</name>
<!--0:cisco;1:velocix;2:generic-->
  <providerType>2</providerType>
<rFqdn>10.74.13.52</rFqdn>
<port>1099</port>
<!-- 0:in service; 1:out of service -->
<serviceState>0</serviceState>
<carrier>6666</carrier>
<cost>0.123</cost>
<description>cdn3</description>
</CDNs-row >
</CDNs>

```

POST or PUT Definition Request Example

Following is an example of a request to create or modify the definition of CDNs.

```

<!-- Data Model (request): POST(create) or PUT(modify) a single CDN -->
<CDN id="236" uri="/api/CDNs/236">
<name>cdn2</name>
<!--0:cisco;1:velocix;2:generic-->
  <providerType>0</providerType>
<rFqdn>10.74.13.42</rFqdn>
<port>1099</port>
<!-- 0:in service; 1:out of service -->
<serviceState>0</serviceState>
<carrier>6666</carrier>
<cost>0.123</cost>
<!-- 0:disable; 1:enable -->
  <enableStatusTrack>1</enableStatusTrack>
<pollInterval>5</pollInterval>
<lowWatermark>50</lowWatermark>
<highWatermark>90</highWatermark>
<description>cdn2</description>
</CDN>

```

CDN Adaptation Policy Settings

The General Settings part of the CDN Management API provides parity for the following VDSM GUI page:

- **Services > Content Delivery Networks (for list)**
- **Services > Content Delivery Networks > Adaptation Policy Settings**

Table 2-2 *CDN Adaptation Policy Settings API Call*

Resource URL	Method	Function Description
/api/CDNs/{id}/AdptPolicySettings/Method	GET	Gets the current policy configuration method of a CDN identified by {id}. The return value will be “File” or “Policies”
/api/CDNs/{id}/AdptPolicySettings/Method/File	POST	Sets “File” as the policy configuration method of a CDN identified by {id}.

Table 2-2 CDN Adaptation Policy Settings API Call (continued)

Resource URL	Method	Function Description
/api/CDNs/{id}/AdptPolicySettings/Method/Policies	POST	Sets “Policies” as the policy configuration method of a CDN identified by {id}.
/api/CDNs/{id}/AdptPolicySettings/File	GET	Gets the CDN Adaptation Policy File of a CDN identified by {id}.
	DELETE	Sets the CDN Adaptation Policy File of a CDN identified by {id} to None.
/api/CDNs/{id}/AdptPolicySettings/File/{field}	POST	Sets a file Identified by {field} as the CDN Adaptation File of a CDN identified by {id}
/api/CDNs/{id}/AdptPolicySettings/Policies	GET	Gets the CDN Adaptation Policy setting of a CDN identified by {id}.
	POST	Modifies the CDN Adaptation Policy setting of a CDN identified by {id}.
	DELETE	Deletes the CDN Adaptation Policy setting of a CDN identified by {id}.
/api/CDNs/{id}/CDNAdptPolicies/Default	GET	Gets the CDN Adaptation Policy of a CDN identified by {id}.
	PUT	Modifies the CDN Adaptation Policy of a CDN identified by {id}.

GET CDN Adaptation Policy Response Example

Following is an example of a response to get the Policies of a VDSM.

```
<!-- Data Model (request): GET current policy configuration method of a CDN -->
<CDN id="236" uri="/api/CDNs/236/AdptPolicySettings/Method">
  <!-- Possible value could be: File or Policies-->
  <policyMethod>File</policyMethod>
</CDN>
```

GET CDN Adaptation Policy File Request Example

Following is an example of a response to get the Adaptation Policy File of a VDSM.

```
<!-- Data Model (response): GET CDN Adaptation Policy File of CDN-->
<CDNAdptPolicies uri="/api/CDNs/233/AdptPolicySettings/File">
  <file name="cdnAdptPolicy.js" id="533" url="/api/FileMgmt/files?type=403/533"/>
</CDNAdptPolicies>
```

POST or PUT Definition Request Example

Following is an example of a request to create or modify the definition of CDN Adaptation Policy.

```
<!-- Data Model (request): POST(Create)/PUT(Modify) assign pre-defined CDN Adaptation
Policies to a CDN -->
<CDNAdptPolicies uri="/api/CDNs/233/AdptPolicySettings/Policies">
<policy id="123" order="1"/>
<policy id="124" order="2"/>
<policy id="125" order="3"/>
</CDNAdptPolicies>
```


GET CDN Adaptation Policy settings Request Example

Following is an example of a response to get the Adaptation Policy settings of a VDSM.

```
<!-- Data Model (response):GET CDN Adaptation Policy settings of a CDN -->
<CDNAdptPolicies uri="/api/CDNs/233/AdptPolicySettings/Policies">
  <policy id="123" order="1"/>
  <policy id="124" order="2"/>
  <policy id="125" order="3"/>
</CDNAdptPolicies>
```

GET or PUT a default CDN Adaptation Policy Request Example

Following is an example of a request to get or modify the default of CDN Adaptation Policy.

```
<!-- Data Model for GET/ PUT a default CDN Adaptation Policy-->
<CDNAdptPolicy uri="/api/CDNs/233/AdptPolicySettings/Policies/Default">
  <urlRewrite>
    <match>redbox.com</match>
    <substitute>redbox.abc.com</substitute>
  </urlRewrite>
  <urlResign>
    <!--0: Cisco, 1: Level3, 2: LimeLight -->
    <vendor>0</vendor>
    <keyNumber>1</keyNumber>
    <keyOwner>1</keyOwner>
    <!--0: include Client IP, 1: exclude Client IP -->
    <excludeClientIp>0</excludeClientIp>
    <!--0: include Domain, 1: exclude Domain, -->
    <excludeDomain>0</excludeDomain>
    <!--0: include Expire Time, 1: exclude Expire Time -->
    <excludeExpiryTime>0</excludeExpiryTime>
    <expiryTime>1</expiryTime>
    <version>0</version>
  </urlResign>
</CDNAdptPolicy >
```

BFQDN Management API

The BFQDN Management API supports listing, provisioning, and control of the BFQDNs and has the following API calls:

- [BFQDN List and Definition](#)
- [BFQDN-Service Broker Assignment](#)
- [BFQDN Policy Settings](#)

BFQDN List and Definition

The BFQDN List and Definition part of the BFQDN API provides parity for the following VDSM GUI pages:

- **Services > BFQDNs**

Table 2-3 BFQDN List and Definition API Calls

Resource URL	Method	Function Description
/api/BFQDNs	GET	Gets the list of all the BFQDNs the API user has access to, which is determined by the Domain Management configured for the user.
	POST	Creates a new BFQDN. The VDSM allocates a long type of positive integer ID for the newly created BFQDN. The “Location” header in the response returns the URI of the created BFQDN, which includes the ID (for example, Location: /api/BFQDNs/235).
/api/BFQDNs/{id}	GET	Gets the BFQDN identified by {id}, a long type of positive integer number allocated by the VDSM when the BFQDN is created.
	PUT	Modifies the BFQDN identified by {id}.
	DELETE	Deletes the BFQDN identified by {id}.

GET List of BFQDNs Response Example

Following is an example of a response to get the list of BFQDNs.

```
<!-- Data Model (response): GET the list of all BFQDNs-->
<BFQDNs uri="/api/ BFQDNs">
<BFQDNs-row id="235" uri="/api/BFQDNs/235">
  <name>domain1</name>
  <description></description>
  <bFqdn>test.com</ bFqdn>
  <!-- 1:Protocol Redirect;2:DNS IP Redirect;3:DNS CNAME Redirect;4: DNS Proxy -->
  <redirectMode>1</redirectMode>
  <!-- 1:IP Packet Header;2:X-Forwarded-For Http Header;3:Via Http Header;4:Http Query
String -->
<ipSourceType>1</ipSourceType>
  <!--IPv4 address, must be configured only when redirect mode is "Protocol redirect"-->
<dnsAResponse>1.2.3.4</dnsAResponse>
  <!--0: None, otherwise id of CDN Metrics Provider-->
<cdnMetricInfoId>0</cdnMetricInfoId>
</BFQDNs-row>
</BFQDNs>
```

POST or PUT BFQDN Definition Request Example

Following is an example of a request to create or modify the definition settings for a BFQDN.

```
<!-- Data Model (request): POST(create) or PUT(modify) a BFQDN-->
<BFQDNs-row id="235" uri="/api/BFQDNs/235">
  <name>domain1</name>
  <description></description>
  <bFqdn>test.com</ bFqdn>
  <!-- 1:Protocol Redirect;2:DNS IP Redirect;3:DNS CNAME Redirect;4: DNS Proxy -->
```

```

        <redirectMode>1</redirectMode>
        <!-- 1:IP Packet Header;2:X-Forwarded-For Http Header;3:Via Http Header;4:Http Query
String -->
<ipSourceType>1</ipSourceType>
        <!--IPv4 address, must be configured only when redirect mode is "Protocol redirect"-->
<dnsAResponse>1.2.3.4</dnsAResponse>
        <!--0: None, otherwise id of CDN Metrics Provider-->
<cdnMetricInfoId>0</cdnMetricInfoId>
</BFQDNs-row>

```

BFQDN-Service Broker Assignment

The BFQDN Service Broker API provides parity for the following VDSM GUI page:

- **Services > BFQDNs > Live Channel > (Select one BFQDN) > Assign Service Brokers**

Table 2-4 BFQDN-Service Broker Assign BFQDN API Call

Resource URL	Method	Function Description
/api/BFQDNs/{id}/SBs	GET	Gets the Service Broker nodes assigned to the BFQDN identified by the {id}.
	POST	Assigns or Unassigns Service Broker nodes to or from the BFQDN identified by {id}.

GET Assigned Service Brokers of BFQDN Response Example

Following is an example of a response to get the assigned Service Brokers for a BFQDN.

```

<!-- Data Model (response): GET the assigned Service Brokers of BFQDN -->
<SBs uri="/api/BFQDNs/244/SBs">
<SB id="245" uri="/api/SBs/245">
    <Name>SPCDN-DE-20</Name>
    <Description>n/a</Description>
    <location>246</location>
    <status>Online</status>
    <ipAddress>10.74.14.141</ipAddress>
    <version>1.1.0.b.37</version>
</SB>
</SBs>

```

POST Definition Request Example

Following is an example of a request to create the definition settings for a BFQDN-Service Broker.

```

<!-- Data Model (request): POST (assign/unassign) Service Brokers to a BFQDN -->
<assign>
    <assignIds>215,216,217,218</assignIds>
<removeIds>228,229</removeIds>
</assign>

```

BFQDN Policy Settings

The General Settings part of the BFQDN Policy API provides parity for the following VDSM GUI page:

- Services > BFQDN (for list)
- Services > BFQDN > Policy

Table 2-5 BFQDN Policy Settings Resources API Calls

Resource URL	Method	Function Description
/api/BFQDNs/{id}/PolicySettings/Method	GET	Gets the current policy configuration method of a BFQDN identified by {id}.
/api/BFQDNs/{id}/PolicySettings/Method/File	POST	Sets "File" as the policy configuration method of a BFQDN identified by {id}.
/api/BFQDNs/{id}/PolicySettings/Method/Policies	POST	Sets "Policies" as the policy configuration method of a BFQDN identified by {id}.
/api/BFQDNs/{id}/PolicySettings/File	GET	Gets BFQDN Policy File of a BFQDN identified by {id}.
	DELETE	Sets the BFQDN policy File of BFQDN identified by {id}.
/api/BFQDNs/{id}/PolicySettings/File/{field}	POST	Sets a identified by {field} as the BFQDN Policy File of a BFQDN identified by {id}.
/api/BFQDNs/{id}/PolicySettings/Policies	GET	Gets the policy settings of a BFQDN identified by {id}.
	POST	Modifies the policy settings of a BFQDN identified by {id}.
	DELETE	Deletes the policy settings of a BFQDN identified by {id}.
/api/BFQDNs/{id}/BFQDNPolicies/Default	GET	Gets the default policy of a BFQDN identified by {id}
	PUT	Modifies the default policy of a BFQDN identified by {id}

GET BFQDN Policy Response Example

Following is an example of a response to get a policy configuration of a BFQDN.

```
<!-- Data Model (request): GET current policy configuration method of a BFQDN -->
<BFQDN id="236" uri="/api/BFQDNs/236/PolicySettings/Method">
  <!-- Possible value could be: File or Policies-->
  <policyMethod>File</policyMethod>
</BFQDN>
```

GET BFQDN Policy File Response Example

Following is an example of a response to get the BFQDN policy file of a BFQDN.

```
<!-- Data Model (response): GET BFQDN Policy File of BFQDN-->
<BFQDNPolicies uri="/api/BFQDNs/233/PolicySettings/File">
  <file name="bfqdnPolicy.js" id="533" url="/api/FileMgmt/files?type=405/533"/>
</CDNAdptPolicies>
```

POST BFQDN Policies to a BFQDN Request Example

Following is an example of a request to assign or unassign Service Brokers to a BFQDN.

```
<!-- Data Model (request): POST(Modify) assign pre-defined BFQDN Policies to a BFQDN -->
<BFQDNPolicies uri="/api/BFQDNs/233/PolicySettings/Policies">
  <policySettings>
    <!--0: disable, 1: enable -->
      <urlValidEnable>1</urlValidEnable>
    <!--0: cisco, 1: velocix -->
      <urlValidVendor>0</urlValidVendor>
      <urlValidKeyIdNum>1</urlValidKeyIdNum>
      <urlValidKeyIdOwner>1</urlValidKeyIdOwner>
    <!--0: include Client IP, 1: exclude ClientIP -->
      <urlValidExcludeClientIp>0</urlValidExcludeClientIp>
    <!--0: include Domain, 1: exclude Domain -->
      <urlValidExcludeDomain>0</urlValidExcludeDomain>
    <!--0: include ExpireTime, 1: exclude ExpireTime -->
      <urlValidExcludeExpiryTime>0</urlValidExcludeExpiryTime>
    <queryParams>
      <QueryParam name="aaa" />
    </queryParams>
  </policySettings>
  <policy id="123" order="1"/>
  <policy id="124" order="2"/>
  <policy id="125" order="3"/>
</BFQDNPolicies>
```

GET BFQDN Policy Settings Response Example

Following is an example of a response to get the BFQDN policy settings of a BFQDN.

```
<!-- Data Model (response): GET Policy settings of a BFQDN -->
<BFQDNPolicies uri="/api/BFQDNs/233/PolicySettings/Policies">
  <policySettings>
    <!--0: disable, 1: enable -->
      <urlValidEnable>1</urlValidEnable>
    <!--0: cisco, 1: velocix -->
      <urlValidVendor>0</urlValidVendor>
      <urlValidKeyIdNum>1</urlValidKeyIdNum>
      <urlValidKeyIdOwner>1</urlValidKeyIdOwner>
    <!--0: include Client IP, 1: exclude ClientIP -->
      <urlValidExcludeClientIp>0</urlValidExcludeClientIp>
    <!--0: include Domain, 1: exclude Domain -->
      <urlValidExcludeDomain>0</urlValidExcludeDomain>
    <!--0: include ExpireTime, 1: exclude ExpireTime -->
      <urlValidExcludeExpiryTime>0</urlValidExcludeExpiryTime>
    </policySettings>
    <queryParams>
      <QueryParam name="aaa" />
    </queryParams>
    <policy id="123" order="1"/>
    <policy id="124" order="2"/>
    <policy id="125" order="3"/>
  </BFQDNPolicies>
```

GET or PUT BFQDN Policy Response Example

Following is an example of a response to get or modify the BFQDN policy.

```
<!-- Data Model: GET/ PUT a default BFQDN Policy-->
<BFQDNPolicy uri="/api/BFQDNs/233/PolicySettings/Policies/Default">
  <!--Allow or Deny -->
  <bfqdnAction allow="Allow"/>
```

```
</BFQDNPolicy>
```

CDN Metrics Provider API

The CDN Metrics Provider API supports listing, provisioning, and control of the CDN Metric Providers and has the following API calls:

- [CDN Metrics Provider List and Definition](#)
- [CDN Metrics Provider-BFQDN Assignment](#)

CDN Metrics Provider List and Definition

The CDN Metrics Provider List and Definition part of the CDN Metrics API provides parity for the following VDSM GUI pages:

- **Services > CDN Metrics Providers**

Table 2-6 CDN Metrics Provider List and Definition API Calls

Resource URL	Method	Function Description
/api/CDNMetrics	GET	Gets the list of created CDN Metrics Providers.
	POST	Creates a new CDN Metrics Provider. CDSM allocates a long type of positive integer ID for the newly-created CDN Metrics Provider. The “Location” header in the response returns the created CDN Metrics Provider’s URI, which includes the ID (for example, Location: /api/CDNMetrics/235).
/api/CDNMetrics/{id}	GET	Get a CDN Metrics Provider identified by {id}.
	PUT	Modify a CDN Metrics Provider identified by {id}.
	DELETE	Delete a CDN Metrics Provider identified by {id}.

GET List of CDN Metrics Provider Response Example

Following is an example of a response to get the list of CDN Metrics Providers.

```
<!-- Data Model (response): GET the list of all CDN Metrics Providers-->
<CDNMetrics uri="/api/CDNMetrics/">
  <CDNMetrics-row id="599" uri="/api/CDNMetrics/599">
    <name>bb</name>
    <!-- 1:Cedexis '
    <vendor>1</vendor>
    <!-- 1:Postal Code;2:City;3:State;4:Country -->
    <granularity>1</granularity>
    <cacheTimeOut>14</cacheTimeOut>
    <cacheMaxEntries>10000</cacheMaxEntries>
    <hostHeader>aa</hostHeader>
    <serverHost>aa</serverHost>
    <serverPort>80</serverPort>
    <serverTimeout>50</serverTimeout>
    <connectCount>10</connectCount>
```

```

    </CDNMetrics-row>
</CDNMetrics>

```

POST or PUT CDN Metrics Provider Definition Request Example

Following is an example of a request to create or modify the definition settings for a CDN Metrics Provider.

```

<<!-- Data Model (request): POST(create) or PUT(modify) a CDN Metrics Provider-->
<CDNMetrics-row id="599" uri="/api/CDNMetrics/599">
  <name>bb</name>
  <!-- 1:Cedexis '
  <vendor>1</vendor>
  <!-- 1:Postal Code;2:City;3:State;4:Country -->
  <granularity>1</granularity>
  <cacheTimeOut>14</cacheTimeOut>
  <cacheMaxEntries>10000</cacheMaxEntries>
  <hostHeader>aa</hostHeader>
  <serverHost>aa</serverHost>
  <serverPort>80</serverPort>
  <serverTimeout>50</serverTimeout>
  <connectCount>10</connectCount>
</CDNMetrics-row>

```

CDN Metrics Provider-BFQDN Assignment

The CDN Metrics Provider BFQDN API provides parity for the following VDSM GUI page:

- **Services > CDN Metrics Providers > Live Channel > (Select one CDN Metrics Provider) > BFQDN Assignment**

Table 2-7 CDN Metrics Provider-BFQDN Assign BFQDN API Call

Resource URL	Method	Function Description
/api/CDNMetrics/{id}/BFQDNs	GET	Gets the BFQDNs assigned to the CDN Metrics Provider identified by {id}.
	POST	Assigns / Unassigns BFQDNs to/from the CDN Metrics Provider identified by {id}.

GET Assigned Service Brokers of CDN Metrics Provider Response Example

Following is an example of a response to get the assigned Service Brokers for a CDN Metrics Provider.

```

<!-- Data Model (response): GET the assigned BFQDNs of a CDN Metrics Provider -->
<BFQDNs uri="/api/CDNMetrics/599/BFQDNs">
  <BFQDNs-row id="629" uri="/api/BFQDNs/629">
    <name>bbb</name>
    <bFqdn>bbb.com</bFqdn>
    <redirectMode>1</redirectMode>
    <ipSourceType>1</ipSourceType>
    <description></description>
    <cdnMetricInfoId>599</cdnMetricInfoId>
  </BFQDNs-row>
</BFQDNs>

```

POST Definition Request Example

Following is an example of a request to create the definition settings for a CDN Metrics Provider-Service Broker.

```
<!-- Data Model (request): POST (assign/unassign) BFQDNs to a CDN Metrics Provider -->
<assign>
  <assignIds>215,216,217,218</assignIds>
<removeIds>228,229</removeIds>
</assign>
```

BFQDN Policy API

The BFQDN Policy API provides parity function on the following VDSM GUI pages:

- **Services > BFQDNs > BFQDN Policies**

Table 2-8 BFQDN Policy API Calls

Resource URL	Method	Function Description
/api/BFQDNPolicies	GET	Gets all BFQDN Policies
	POST	Creates a BFQDN Policy by specifying Rules and Actions
/api/BFQDNPolicies/{id}	GET	Gets the BFQDN Policy identified by {id}
	PUT	Modifies the BFQDN Policy identified by {id}
	DELETE	Deletes the BFQDN Policy identified by {id}
/api/BFQDNPolicies/{id}/BFQDNs	GET	Gets the assigned BFQDNs for the BFQDN Policy identified by {id}
	POST	Unassigns BFQDNs for the BFQDN Policy identified by {id}

GET BFQDN Policy Response Example

Following is an example of a response to get the BFQDN Policies.

```
<!-- Data Model (response): GET the list of all BFQDNs assigned to a BFQDN policy -->
<BFQDNs uri="/api/BFQDNPolicies/233/ BFQDNs">
<BFQDNs-row id="235" uri="/api/BFQDNs/235">
  <name>domain1</name>
  <description></description>
  <bFqdn>test.com</ bFqdn>
  <!-- 1:Protocol Redirect;2:DNS IP Redirect;3:DNS CNAME Redirect;4: DNS Proxy -->
  <redirectMode>1</redirectMode>
</BFQDNs-row>
</BFQDNs>
```

GET or POST or PUT BFQDN Policy Response Example

Following is an example of a response to get or unassign or modify the BFQDN Policy.

```
<!-- Data Model: GET/POST/PUT a BFQDN Policy-->
<BFQDNPolicy uri="/api/BFQDNPolicies/233">
  <name>test</name>
  <description>Created by API</description>
```



```

<rule>
<cpeType>
  <operation>EQ</operation>
  <values>
    <value>ABC</value>
  </values>
</cpeType>
</rule>
<rule>
  <city>
    <operation>IN</operation>
    <values>
      <value>San Jose</value>
    </values>
  </city>
  <state>
    <operation>EQ</operation>
    <values>
      <value>California</value>
    </values>
  </state>
</rule>
<!--Allow or Deny -->
<bfqdnAction allow="Allow"/>
</BFQDNPolicy>

```

POST BFQDNs to a BFQDN Policy Request Example

Following is an example of a request to unassign BFQDNs to a BFQDN policy.

```

<!-- Data Model (request): POST (unassign) BFQDNs to a BFQDN policy-->
<assign>
<removeIds>235,239</removeIds>
</assign>

```

CDN Adaptation Policy API

The CDN Adaptation Policy APIs manages CDN Adaptation policies by specifying rules and actions. The API provides parity for the following VDSM GUI page:

- **System > Content Delivery Networks > CDN Adaptation Policies**

Table 2-9 *CDN Adaptation Policy API Calls*

Resource URL	Method	Function Description
/api/CDNAdptPolicies	GET	Gets all CDN Adaptation Policies.
	POST	Creates a CDN Adaptation Policy by specifying rules and Actions.
/api/CDNAdptPolicies/{id}	GET	Gets a CDN Adaptation Policy identified by {id}.
	PUT	Modifies a CDN Adaptation Policy identified by {id}.
	DELETE	Deletes a CDN Adaptation Policy identified by {id}.

Table 2-9 *CDN Adaptation Policy API Calls (continued)*

Resource URL	Method	Function Description
/api/CDNAdptPolicies/{id}/CDNs	GET	Gets assigned CDNs for the CDN Adaptation Policy identified by {id}.
	POST	Unassigns CDNs for the CDN Adaptation Policy identified by {id}.

GET or POST or PUT CDN Policy Response Example

Following is an example of a response to get or unassign or modify the CDN Policy.

```

<!-- Data Model (response): GET/POST/PUT a CDN Adaptation Policy-->
<CDNAdptPolicy uri="/api/CDNAdptPolices/233">
  <name>test</name>
  <description>Created by API</description>
  <!--0: URL Rewrite, 1: URL Generate -->
  <urlActionType>1</urlActionType>
<rule>
<cpeType>
  <operation>EQ</operation>
  <values>
    <value>ABC</value>
  </values>
</cpeType>
</rule>
<rule>
  <city>
    <operation>IN</operation>
    <values>
      <value>San Jose</value>
    </values>
  </city>
  <state>
    <operation>EQ</operation>
    <values>
      <value>Califonia</value>
    </values>
  </state>
</rule>
<urlResign>
  <!--0: Cisco, 1: Level3, 2: LimeLight, 3: Velocix -->
  <vendor>0</vendor>
  <keyNumber>1</keyNumber>
  <keyOwner>1</keyOwner>
  <!--0: include Client IP, 1: exclude Client IP -->
  <excludeClientIp>0</excludeClientIp>
  <!--0: include Domain, 1: exclude Domain, -->
  <excludeDomain>0</excludeDomain>
  <!--0: include Expire Time, 1: exclude Expire Time -->
  <excludeExpiryTime>0</excludeExpiryTime>
  <expiryTime>1</expiryTime >
  <version>0</version>
</urlResign>
<urlInHttp>302</urlInHttp>
<urlGenerate>
  <domain>redbox.com</domain>
  <path>test</path>
  <queryParam>
    <name>aaa</name>
    <source>REQUEST</source>

```

```

        <attribute>HEADER</attribute>
        <value>bbb</value>
    </queryParam>
</urlGenerate>
</CDNAdptPolicy >

```

GET CDN Adaptation Policy Response Example

Following is an example of a response to get all the Network Storage Shares. In this example, there is only one Network Storage Share configured.

```

<!-- Data Model (response): GET all CDNs assigned to a CDN adapt policy -->
<CDNs uri="/api/CDNAdptPolicies/233/CDNs">
  <CDN id="236" uri="/api/CDNs/236">
    <name>cdn2</name>
    <!-- 0:cisco;1:velocix;2:generic-->
    <providerType>0</providerType>
    <rFqdn>10.74.13.42</rFqdn>
    <port>1099</port>
    <!-- 0:in service; 1:out of service -->
    <serviceState>0</serviceState>
    <!-- 0:disable; 1:enable -->
    <enableStatusTrack>1</enableStatusTrack>
    <pollInterval>5</pollInterval>
    <lowWatermark>50</lowWatermark>
    <highWatermark>90</highWatermark>
    <description>cdn2</description>
  </CDN>
</CDNs>

```

POST CDN Adaptation Policy Request Example

Following is an example of a request to POST (create) a CDN Adaptation Policy.

```

<!-- Data Model (request): POST (unassign) CDNs to a CDN adapt policy-->
<assign>
<removeIds>236,238</removeIds>
</assign>

```

CDN Selection Policy API

The CDN Selection Policy API provides parity on the following VDSM GUI page:

- **Services > CDN Selection Policies**

Table 2-10 *CDN Selection Policy API Calls*

Resource URL	Method	Function Description
/api/CDNSelPolicySettings/Method	GET	Gets the current policy configuration method of a CDN Selection Policy.
/api/CDNSelPolicySettings/Method/File	POST	Sets “File” as the policy configuration method of a CDN Selection Policy.
/api/CDNSelPolicySettings/Method/Policies	POST	Sets “Policies” as the policy configuration method of a CDN Selection Policy.

Table 2-10 CDN Selection Policy API Calls (continued)

Resource URL	Method	Function Description
/api/CDNSelPolicySettings/File	GET	Gets Global CDN Policy File of a CDN Selection Policy.
	DELETE	Sets the Global CDN policy File of CDN Selection Policy to none.
/api/CDNSelPolicySettings/File/{field}	POST	Sets a identified by {field} as the Global CDN Policy File of a CDN Selection Policy.
/api/CDNSelPolicies	GET	Gets all CDN Selection Policies.
	POST	Creates the policy settings of a CDN Selection Policy.
/api/CDNSelPolicies/{id}	GET	Gets the policy settings of a CDN Selection identified by {id}.
	PUT	Modifies the policy settings of a CDN Selection identified by {id}.
	DELETE	Deletes the policy settings of a CDN Selection identified by {id}.
/api/CDNSelPolicySettings/Policies	GET	Gets the order of the defined CDN Selection Policies
	POST	Specifies the order of the defined CDN Selection Policies

GET CDN Selection Policy Response Example

Following is an example of a response to get current policy method of CDN selection Policy.

```
<!-- Data Model (request): GET current policy method of CDN Selection Policy-->
<CDNSelPolicySettings id="236" uri="/api/CDNSelPolicySettings/Method">
  <!-- Possible value could be: File or Policies-->
  <policyMethod>File</policyMethod>
</CDNSelPolicySettings >
```

GET CDN Selection Policy File Response Example

Following is an example of a response to get current policy file of CDN selection Policy.

```
<!-- Data Model (response): GET Global CDN Selection Policy File-->
<CDNSelPolicySettings uri="/api/CDNSelPolicySettings/File">
  <file name="cdnSelPolicy.js" id="533" url="/api/FileMgmt/files?type=402/533"/>
</CDNSelPolicySettings>
```

GET the order of CDN Selection Policies Response Example

Following is an example of a response to get the order of CDN selection Policies.

```
<!-- Data Model (Response): GET the order of defined CDN Selection Policies-->
<CDNSelPolicies uri="/api/CDNSelPolicySettings/Policies/Order">
<policy id="123" order="1"/>
<policy id="124" order="2"/>
<policy id="125" order="3"/>
</CDNSelPolicies>
```

POST Order of defined CDN Selection Policies Request Example

Following is an example of a request to specify the order of defined CDN Selection Policies.

```
<!-- Data Model (request): POST specify the order of defined CDN Selection Policies-->
<CDNSelPolicies uri="/api/CDNSelPolicySettings/Policies/Order">
  <policy id="123" order="1"/>
  <policy id="124" order="2"/>
  <policy id="125" order="3"/>
</CDNSelPolicies>
```

GET or POST or PUT CDN Policy Response Example

Following is an example of a response to get or create or modify the CDN Policy.

```
<!-- Data Model: GET/POST/PUT a CDN Selection Policy-->
<CDNSelPolicy uri="/api/CDNSelPolices/233">
  <name>test</name>
  <description>Created by API</description>
  <!--0: disable, 1: enable -->
    <enableAdvancedAction>1</enableAdvancedAction>
  <!--0: disable, 1: enable -->
    <preferOnNetCDN>1</preferOnNetCDN>
  <rule>
    <cdnMetric>
      <metricName>aaa</metricName>
      <operation>EQ</operation>
      <values>
        <value>1.23</value>
      </values>
    </cdnMetric>
  </rule>
  <rule>
    <city>
      <operation>IN</operation>
      <values>
        <value>San Jose</value>
      </values>
    </city>
    <state>
      <operation>EQ</operation>
      <values>
        <value>California</value>
      </values>
    </state>
  </rule>
  <allocation>
    <beginTime>00:00:00</beginTime>
    <endTime>23:59:59</endTime>
    <cdn name="cdn1" load="100"/>
  </allocation>
  <cdnRankCriteria>
    <preferCost/>
  </cdnRankCriteria>
  <cdnRankCriteria>
    <preferPerformance>
      <metricName>RTT</metricName>
      <sortOrder>ASCENDING</sortOrder>
    </preferPerformance>
  </cdnRankCriteria>
</CDNSelPolicy>
```

Service Broker API

The Service Broker API supports listing, provisioning, and control of the Service Brokers and has the following API calls:

- [Service Broker List and Definition](#)
- [Service Broker's BFQDN Assignment](#)

Service Broker List and Definition

The Service Broker API provides support to list all registered Service Brokers.

Table 2-11 Service Broker List and Definition API Calls

Resource URL	Method	Function Description
/api/SBs	GET	Lists all Service Brokers.
/api/SBs/{id}	PUT	Modifies a Service Broker identified by {id}.
	DELETE	Deletes a Service Broker identified by {id}.

GET Definition Response Example

Following is an example of a response to get Service Brokers with the ID 245.

```
<!-- Data Model (response): GET the list of all Service Brokers -->
<SBs uri="/api/SBs">
  <SB id="245" uri="/api/SBs/245">
    <name>SPCDN-DE-20</name>
    <description>n/a</description>
    <locationId>246</locationId>
    <status>Inactive</status>
    <isActive>false</isActive>
    <isServerOffload>false</isServerOffload>
    <replaceable>0</replaceable>
    <cdnNetworkFileId>249</cdnNetworkFileId>
    <mgmtIncomingType>1</mgmtIncomingType>
    <mgmtIncomingIP></mgmtIncomingIP>
    <MgmtIncomingPort>443</MgmtIncomingPort>
    <dnsTtl>60</dnsTtl>
  </SB>
</SBs>
```

PUT Definition Request Example

Following is an example of a request to create or modify the definition of Service Brokers.

```
<!-- Data Model (request): PUT (modify) a Service Broker -->
<SB id="245" uri="/api/SBs/245">
  <name>SPCDN-DE-20</name>
  <description>n/a</description>
  <locationId>0</locationId>
  <status>Inactive</status>
  <isActive>true</isActive>
  <isServerOffload>false</isServerOffload>
  <replaceable>0</replaceable>
  <cdnNetworkFileId>249</cdnNetworkFileId>
  <mgmtIncomingType>1</mgmtIncomingType>
```

```

    <mgmtIncomingIP></mgmtIncomingIP>
    <MgmtIncomingPort>443</MgmtIncomingPort>
    <dnsTtl>60</dnsTtl>
    <createLocation>true</createLocation>
    <parentLocationId>0</parentLocationId>
  </SB>

```

Service Broker's BFQDN Assignment

The Service Broker API provides support to assign or unassign BFQDN to or from Service Brokers.

Table 2-12 Service Broker's BFQDN Assignment API Calls

Resource URL	Method	Function Description
/api/SBs/{id}/BFQDNs	GET	Lists all assigned BFQDNs
	POST	Assign or Unassigns BFQDNs to or from the Service Broker identified by {id}

GET Definition Response Example

Following is an example of a response to get all assigned BFQDNs with the ID 235.

```

<!-- Data Model (response): GET the list of all assigned BFQDNs-->
<BFQDNs uri="/api/SBs/229/BFQDNs">
<BFQDNs-row id="235" uri="/api/BFQDNs/235">
  <name>domain1</name>
  <description></description>
  <bFqdn>test.com</ bFqdn>
  <!-- 1:HTTP Redirect;2:DNS IP Redirect;3:DNS CNAME Redirect;4: DNS Proxy -->
  <redirectMode>1</redirectMode>
</BFQDNs-row>
</BFQDNs>

```

POST Definition Request Example

Following is an example of a request to assign or unassign BFQDN to a Service Broker.

```

<!-- Data Model (request): POST (assign/unassign) BFQDNs to a Service Broker-->
<assign>
  <assignIds>236,237,238</assignIds>
  <removeIds>235</removeIds>
</assign>

```

File Management API

The File Management API provides support to register a configuration file with the VDSM by using the import or upload method. The File Management API provides parity for the following VDSM GUI pages:

Using Multipart/Form-Data Request to Upload a File

For the upload method, a multipart/form-data request is used. Following is an example of the upload import method for registering a file that uses the curl utility to upload a CDN Network file:

```
curl -k -v -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@cov1.xml;type=application/xml" -F "request=@cov1.xml;type=application/xml" -u
admin:default "https://10.74.61.199:8443/api/FileMgmt/files?type=1"
```

In this example, the curl utility uploads the file cov1.xml.

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

Supported curl Version

In very old version of curl (for example, 7.11.0 January 2004), the HTTP multipart/form-data request sent by curl misses the boundary Content-Type header. This does not occur in versions 7.15.x or later.

Add “;type=application/xml” to the -F argument for XML files. For example,

```
curl -k -v -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@tstvl.xml;type=application/xml" -F
"request=@tstvl.xml;type=application/xml" -u admin:default
"https://1.2.3.4:8443/api/FileMgmt/files?type=301"
```



Note

The curl version 7.19 or later detects an XML file and adds the “type=application/xml” itself.

For more information on the **curl** utility, see the manual (man) pages located at the following URL:

<http://curl.haxx.se/docs/manpage.html>

Table 2-13 File Management API Calls

Resource URL	Method	Function Description
/api/FileMgmt/types	GET	Lists all file types supported by VDSM. The response returns the file type value for each kind of supported file.
/api/FileMgmt/files?type={type}	GET	Lists all registered files of type {type}.
	POST	Registers a file of type {type} using the import method (imports the file from an external HTTP or FTP file server).
	POST (multipart/form-data)	Registers a file of type {type} using the upload method (uploads the file from the local disk of the API client). The request Content-Type must be multipart/form-data. See the “POST Upload a File Example” section on page 2-23.

Table 2-13 File Management API Calls (continued)

Resource URL	Method	Function Description
/api/FileMgmt/files;type={type}/{id}	GET	Gets the registered file of type {type}, and file ID {id}. Both {type} and {id} are integer numbers, allocated by VDSM.
	PUT	Modifies a file using the import method.
	PUT (multipart/form-data)	Modifies a file using the upload method. The request Content-Type is multipart/form-data. See the “PUT Modify a File Example (Upload Method)” section on page 2-23.
	DELETE	Deletes a file from VDSM.
/api/FileMgmt/validate;type={type}	POST	Validates a file on a file server, using import method. The file is not registered at this point, so there is no file ID. The file is fetched from the file server and then validated, but VDSM does not store it.
/api/FileMgmt/validate;type={type}/{id}	POST	Validate an existing file that is registered to the VDSM and has a file ID.
/api/FileMgmt/refetch;type={type}/{id}	POST	Refetches a registered file from the file server. If the file on the file server has changes, the file stored on the VDSM is updated.
/api/FileMgmt/GlobalFiles;type={type}	GET	Gets selected global files of type {type}.
	DELETE	Sets global files of specified file type to none.
/api/FileMgmt/GlobalFiles;type={type}/{id}	POST	Sets the file identified by file ID as global configuration file for specific type {type}

GET All File Types Response Example

Following is an example of a response to get all the file types for VDSM (GET /api/FileMgmt/types).

```
<!-- Data Model (response): GET all file types -->
<fileMgmt uri="/api/fileMgmt">
  <result message="Show all supported file types" status="success">
    <types>
      <type value="401">Service Broker Policy File</type>
      <type value="402">CDN Selection Policy File</type>
      <type value="403">CDN Adaptation Policy File</type>
      <type value="404">CDN Network File</type>
      <type value="405">BFQDN policy file</type>
    </types>
  </result>
</fileMgmt>
```

GET All Files of Specified Type Response Example

Following is an example of a response to get all CDN Network files; the type is 404 (GET /api/FileMgmt/files;type=404).

```
<!-- Data Model (response): GET files by file type -->
<fileMgmt uri="/api/fileMgmt">
  <result message="2 CDN Network Files are displayed." status="success">
    <file type="404" id="401">
      <originUrl>cdnNetworkFile1.xml</originUrl>
    </file>
  </result>
</fileMgmt>
```

```

        <destName>cdnNetworkFile1.xml</destName>
        <username />
        <password />
        <ttl>10</ttl>
    </file>
    <file type="404" id="402">
        <originUrl>cdnNetworkFile2.xml</originUrl>
        <destName>cdnNetworkFile2.xml</destName>
        <username />
        <password />
        <ttl>10</ttl>
    </file>
</fileMgmt>

```

GET a Single File Response Example

Following is an example of a response to get a CDN Network file (type = 404) with ID 401 (GET /api/FileMgmt/files;type=404/401)

```

<!-- Data Model (response): GET a single file -->
<fileMgmt uri="/api/fileMgmt">
    <result message="1 CDN Network File is displayed." status="success">
        <file type="404" id="401">
            <originUrl>cdnNetworkFile1.xml</originUrl>
            <destName>cdnNetworkFile1.xml</destName>
            <username />
            <password />
            <ttl>10</ttl>
        </file>
    </result>
</fileMgmt>

```

POST Import a File Request Example

Following is an example of a request to register a CDN Network file by the import method (POST /api/FileMgmt/files;type=404). The ID allocated to the file is returned in the response body as the ID attribute of the <file> element.

```

<!-- Data Model (request): register a file by import method -->
<file>
    <originUrl>http://1.2.3.4/pub/cdnNetworkFile1.xml</originUrl>
    <destName>cdnNetworkFile1.xml</destName>
    <username>admin</username>
    <password>password</password>
    <ttl>10</ttl>    <!-- update interval, in minutes -->
</file>

```

PUT Modify a File Request Example (Import Method)

Following is an example of a request to modify a CDN Network file with ID 401 by using the import method (PUT /api/FileMgmt/files;type=404/401).

```

<!-- Data Model (request): modify a file by import method -->
<file>
    <originUrl>http://1.2.3.4/pub/cdnNetworkFile1.xml</originUrl>
    <destName>cdnNetworkFile1.xml</destName>
    <username>admin</username>
    <password>password</password>
    <ttl>20</ttl>    <!-- update interval, in minutes -->
</file>

```

POST Upload a File Example

Following is an example of registering a file by using the upload method (POST `/api/FileMgmt/files?type=402` with multipart/form-data request).

In the following example, the `cdnSelPlociyFile.xml` is the CDN Selection Policy file to be uploaded to the VDSM, and the `cdnSelPlociyFile.xml` specifies the parameters (also known as file information metadata) used to register the CDN Selection Policy file.

```
# POST multipart/form-data: register a file by upload method
curl -k -X POST -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@cdnSelPolicyFile.xml ;type=application/xml" -F
"request=@cdnSelPolicyFileInfo.xml;type=application/xml" -u admin:default
"https://1.2.3.4:8443/api/FileMgmt/files?type=402"
```

Currently, only the `destName` element (destination name) needs to be provided for the file metadata.

```
# cdnSelPolicyFileInfo.xml
<FileInfo>
  <!-- namespace is only applied to file types of 402,403,405 -->
  <namespace>ns_bfqdn_test1</namespace>
  <destName>destCdnSelPolicyFile.xml</destName>
</FileInfo>
```

PUT Modify a File Example (Upload Method)

Following is an example of modifying a registered file by using the import method (PUT `/api/FileMgmt/files?type={type}/{id}` with multipart/form-data request).

E.g., PUT `/api/FileMgmt/files?type=402`

In the following example, the `cdnSelPlociyFile.xml` file is the CDN Selection Policy file to be uploaded to the VDSM, and the `cdnSelPlociyFile.xml` specifies the parameters (also known as file information metadata) used to register the CDN Selection Policy file.

```
# PUT multipart/form-data: modify a file by upload method
curl -k -X PUT -H "Accept: application/xml" -H "Content-Type: multipart/form-data" -F
"file=@ cdnSelPolicyFile.xml;type=application/xml" -F
"request=@ cdnSelPolicyFileInfo.xml;type=application/xml" -u username:password
"https://1.2.3.4:8443/api/FileMgmt/files?type=402/307"
```

Currently, only `destName` element (destination name) needs to be provided for the file metadata.

```
# cdnSelPolicyFileInfo.xml
<FileInfo>
  <!-- namespace is only applied to file types of 402,403,405 -->
  <namespace>ns_bfqdn_test1</namespace>
  <destName>destCdnSelPolicyFile.xml</destName>
</FileInfo>
```

DELETE a File Example

Following is an example of a response to delete a CDN Selection Policy file (type = 404) with ID 501 (DELETE `/api/FileMgmt/files?type=404/501`).

The request body is empty. The response returns the deletion result.

```
<!-- Data Model (response): delete a file. -->
<fileMgmt uri="/api/fileMgmt">
  <result message="The file is deleted." status="success">
</fileMgmt>
```

POST Validate a File Example (Import Method)

Following is an example of a request to validate a file by using the import method (POST: /api/FileMgmt/files;type={type}). The file has not been registered with the VDSM; therefore it does not have a file ID.

```
<!-- Data Model (request): validate a file by import method -->
<file>
  <originUrl>http://1.2.3.4/pub/cdnNetworkFile1.xml</originUrl>
  <destName>cdnNetworkFile1.xml</destName>
  <username>admin</username>
  <password>password</password>
</file>
```

POST Validate a File Example (Upload Method)

Following is an example of a response to validate a registered CDN Selection Policy file with ID 408 by using the upload method (POST: /api/FileMgmt/validate;type=404/408). The request body is empty. The response returns the validation result.

```
<!-- Data Model (response): validation result. -->
<fileMgmt uri="/api/fileMgmt">
  <result message="Valid file" status="success">
</fileMgmt>
```

POST Refetch a File Response Example

Following is an example of a response to refetch a CDN Selection Policy file with ID 501 (POST /api/FileMgmt/refetch;type=404/501). The request body is empty. The response returns the refetch operation result.

```
<!-- Data Model (response): refetch operation result -->
<fileMgmt uri="/api/fileMgmt">
  <result message="The file will be refetched soon." status="success">
</fileMgmt>
```

GET Gobal Files a File Example

Following is an example of a response to get a Global Files file (type = 401) (GET /api/FileMgmt/GlobalFiles;type=401). The response returns configured global Service Broker file is any.

```
<!-- Data Model (response): GET selected Global File of Service Broker Policy Files -->
<GlobalFileConfig uri="/api/FileMgmt/GlobalFiles;type=401">
  <file name="sbPolicy.js" id="534" url="/api/FileMgmt/files;type=401/534"/>
</ GlobalFileConfig>
```

XML Schema for Policy API

The XML schema is broken into the following files:

- [CDNRules.xsd](#)
- [CDNPolicies.xsd](#)

CDNRules.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!--W3C Schema generated by XMLSpy v2009 sp1 (http://www.altova.com)-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:documentation> Revision: 1.0 </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation> Schema used to validate a CDN Policy </xs:documentation>
  </xs:annotation>
  <xs:complexType name="IntegerListType">
    <xs:sequence>
      <xs:element name="value" type="xs:nonNegativeInteger" minOccurs="1"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="DoubleListType">
    <xs:sequence>
      <xs:element name="value" type="xs:double" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="StringType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="StringListType">
    <xs:sequence>
      <xs:element name="value" type="StringType" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

<xs:complexType name="LineSpeedListType">
  <xs:sequence>
    <xs:element name="value" type="LineSpeedValueType" minOccurs="1"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="SortType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="ASCENDING"/>
    <xs:enumeration value="DESCENDING"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StringOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQ"/>
    <xs:enumeration value="NOTEQ"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="NOTIN"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="StringExpOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQ"/>
    <xs:enumeration value="NOTEQ"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="NOTIN"/>
    <xs:enumeration value="CONTAINS"/>
    <xs:enumeration value="STARTS"/>
    <xs:enumeration value="ENDS"/>
    <xs:enumeration value="REGEX"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="DateOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LEQ"/>
    <xs:enumeration value="GEQ"/>
  </xs:restriction>
</xs:simpleType>

```

```
<xs:simpleType name="TimeOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LEQ"/>
    <xs:enumeration value="GEQ"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="SatLocOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="LEQ"/>
    <xs:enumeration value="GEQ"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="IntegerOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQ"/>
    <xs:enumeration value="NOTEQ"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="NOTIN"/>
    <xs:enumeration value="LEQ"/>
    <xs:enumeration value="GEQ"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="MapOperationType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQ"/>
    <xs:enumeration value="NOTEQ"/>
    <xs:enumeration value="IN"/>
    <xs:enumeration value="NOTIN"/>
    <xs:enumeration value="CONTAINS"/>
    <xs:enumeration value="STARTS"/>
    <xs:enumeration value="ENDS"/>
    <xs:enumeration value="REGEX"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="LineSpeedValueType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="low"/>
    <xs:enumeration value="medium"/>
  </xs:restriction>
</xs:simpleType>
```

```

        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="CityType">
    <xs:sequence>
        <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="values" type="StringListType" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="StateType">
    <xs:sequence>
        <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="values" type="StringListType" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CountryType">
    <xs:sequence>
        <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
        <xs:element name="values" type="StringListType" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="LongLatType">
    <xs:sequence>
        <xs:element name="distance" type="xs:positiveInteger" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="longitude" use="required" type="xs:float"/>
    <xs:attribute name="latitude" use="required" type="xs:float"/>
</xs:complexType>
<xs:complexType name="SatLocValueType">
    <xs:sequence>
        <xs:element name="longLat" type="LongLatType" minOccurs="1"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CpeType">
    <xs:sequence>
        <xs:element name="operation" type="StringExpOperationType" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="values" type="StringListType" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>

```



```
</xs:sequence>
</xs:complexType>
<xs:complexType name="ContentType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="BFQDNTType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="OnNetNameType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ConnectionType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ASNType">
  <xs:sequence>
    <xs:element name="operation" type="IntegerOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="IntegerListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="MSAType">
  <xs:sequence>
    <xs:element name="operation" type="IntegerOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="IntegerListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

```

<xs:complexType name="HTTPHeaderType">
  <xs:sequence>
    <xs:element name="headerName" type="StringType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="operation" type="MapOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RequestTimeType">
  <xs:sequence>
    <xs:element name="operation" type="TimeOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="value" type="xs:string" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RequestDateType">
  <xs:sequence>
    <xs:element name="operation" type="DateOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="value" type="xs:string" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ContentUrlType">
  <xs:sequence>
    <xs:element name="operation" type="StringExpOperationType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="SatLocType">
  <xs:sequence>
    <xs:element name="operation" type="SatLocOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="value" type="SatLocValueType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="LineSpeedType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="LineSpeedListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

```

<xs:complexType name="PostalCodeType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CarrierType">
  <xs:sequence>
    <xs:element name="operation" type="StringExpOperationType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="values" type="StringListType" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="QueryType">
  <xs:sequence>
    <xs:element name="operation" type="StringOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="StringListType" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CDNMetricType">
  <xs:sequence>
    <xs:element name="metricName" type="StringType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="operation" type="IntegerOperationType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="values" type="DoubleListType" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="PreferCostType"/>
<xs:complexType name="PreferPerformanceType">
  <xs:sequence>
    <xs:element name="metricName" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="sortOrder" type="SortType" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RankingCriteriaType">
  <xs:choice>
    <xs:element name="preferCost" type="PreferCostType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="preferPerformance" type="PreferPerformanceType" minOccurs="1"
maxOccurs="1"/>
  </xs:choice>
</xs:complexType>

```

```

</xs:choice>
</xs:complexType>
<xs:complexType name="RuleType">
  <xs:choice>
    <xs:element name="cpeType" type="CpeType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="contentType" type="ContentType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="bfqdn" type="BFQDNType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="onNetName" type="OnNetNameType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="offNetName" type="OnNetNameType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="connectionType" type="ConnectionType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="asn" type="ASNType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="msa" type="MSAType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="httpHeader" type="HTTPHeaderType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="queryParameter" type="HTTPHeaderType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="requestTime" type="RequestTimeType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="requestDate" type="RequestDateType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="contentUrl" type="ContentUrlType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="country" type="CountryType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="state" type="StateType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="city" type="CityType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="distance" type="SatLocType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="lineSpeed" type="LineSpeedType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="postalCode" type="PostalCodeType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="carrier" type="CarrierType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="queryType" type="QueryType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="cdnMetric" type="CDNMetricType" minOccurs="1" maxOccurs="1"/>
  </xs:choice>
</xs:complexType>
<xs:complexType name="CDNTimeSplitType">
  <xs:attribute name="name" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="64"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>

```

```

<xs:attribute name="load" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
<xs:complexType name="TimeSplitType">
  <xs:sequence>
    <xs:element name="beginTime" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="endTime" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="cdn" type="CDNTimeSplitType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:simpleType name="BooleanType">
  <xs:restriction base="xs:boolean">
    <xs:pattern value="0|1"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="URLResignType">
  <xs:sequence>
    <xs:element name="cdnVendor" type="xs:nonNegativeInteger" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="keyIdOwmer" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="keyIdNum" type="xs:integer" minOccurs="1" maxOccurs="1"/>
    <xs:element name="excludeClientIp" type="BooleanType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="excludeDomain" type="BooleanType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="excludeExpiryTime" type="BooleanType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="expiryTime" type="xs:positiveInteger" minOccurs="0" maxOccurs="1"/>
    <xs:element name="version" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="URLRewriteType">
  <xs:sequence>
    <xs:element name="pattern" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="substitute" type="xs:string" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>

```

```

</xs:complexType>
<xs:simpleType name="QueryParamSourceType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="REQUEST"/>
    <xs:enumeration value="USER"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="QueryParamAttributeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BFQDN"/>
    <xs:enumeration value="PATH"/>
    <xs:enumeration value="HEADER"/>
    <xs:enumeration value="QUERYPARAM"/>
    <xs:enumeration value="EXPIRYTIME"/>
    <xs:enumeration value="ASN"/>
    <xs:enumeration value="CARRIER"/>
    <xs:enumeration value="CITY"/>
    <xs:enumeration value="COUNTRY"/>
    <xs:enumeration value="STATE"/>
    <xs:enumeration value="CONNECTION"/>
    <xs:enumeration value="LINESPEED"/>
    <xs:enumeration value="MSA"/>
    <xs:enumeration value="POSTALCODE"/>
    <xs:enumeration value="CLIENTIP"/>
    <xs:enumeration value="GMTOFFSET"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="QueryParamType">
  <xs:sequence>
    <xs:element name="name" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="source" type="QueryParamSourceType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="attribute" type="QueryParamAttributeType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="value" type="xs:string" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="URLGenerateType">
  <xs:sequence>
    <xs:element name="domain" type="xs:string" minOccurs="1" maxOccurs="1"/>

```

```

<xs:element name="path" type="xs:string" minOccurs="0" maxOccurs="1"/>
<xs:element name="queryParam" type="QueryParamType" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="BFQDNAccessType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Allow"/>
    <xs:enumeration value="Deny"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="BFQDNActionType">
  <xs:attribute name="allow" use="required" type="BFQDNAccessType"/>
</xs:complexType>
</xs:schema>

```

CDNPolicies.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace=""
xmlns=""
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:annotation>
  <xs:documentation>Schema used to specify a CDN Policy</xs:documentation>
</xs:annotation>
<xs:include schemaLocation="CDNRules.xsd" />
<xs:complexType name="BFQDNPolicyType">
  <xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="bfqdnAction" type="BFQDNActionType" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:positiveInteger"/>
  <xs:attribute name="uri" type="xs:string"/>
</xs:complexType>

```

```

<xs:complexType name="CDNAdaptPolicyType">
  <xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <xs:element name="urlActionType" type="xs:nonNegativeInteger" />
    <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="urlRewrite" type="URLRewriteType" maxOccurs="unbounded"/>
    <xs:element name="urlResign" type="URLResignType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="urlInHttp" type="xs:positiveInteger" minOccurs="0" maxOccurs="1"/>
    <xs:element name="urlGenerate" type="URLGenerateType" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:positiveInteger"/>
  <xs:attribute name="uri" type="xs:string"/>
</xs:complexType>
<xs:complexType name="CDNSelectPolicyType">
  <xs:sequence>
    <xs:element name="name" type="xs:string" />
    <xs:element name="description" type="xs:string" minOccurs="0"/>
    <xs:element name="priority" type="xs:positiveInteger"/>
    <xs:element name="enableAdvancedAction" type="BooleanType" />
    <xs:element name="preferOnNetCDN" type="BooleanType" minOccurs="0"/>
    <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="allocation" type="TimeSplitType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="cdnRankCriteria" type="RankingCriteriaType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:positiveInteger"/>
  <xs:attribute name="uri" type="xs:string"/>
</xs:complexType>
<xs:complexType name="BFQDNPoliciesType">
  <xs:sequence>
    <xs:element name="BFQDNPolicy" type="BFQDNPolicyType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CDNAdaptPoliciesType">
  <xs:sequence>

```



```

        <xs:element name="CDNAdptPolicy" type="CDNAdaptPolicyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="CDNSelectPoliciesType">
    <xs:sequence>
        <xs:element name="CDNSelPolicy" type="CDNSelectPolicyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="BFQDNPolicies" type="BFQDNPoliciesType" />
<xs:element name="BFQDNPolicy">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="name" type="xs:string" />
            <xs:element name="description" type="xs:string" minOccurs="0"/>
            <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="bfqdnAction" type="BFQDNActionType" minOccurs="0"
maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="id" type="xs:positiveInteger"/>
        <xs:attribute name="uri" type="xs:string"/>
    </xs:complexType>
</xs:element>
<xs:element name="CDNAdptPolicies" type="CDNAdaptPoliciesType" />
<xs:element name="CDNAdptPolicy">
<xs:complexType>
    <xs:sequence>
        <xs:element name="name" type="xs:string" />
        <xs:element name="description" type="xs:string" minOccurs="0"/>
        <xs:element name="urlActionType" type="xs:nonNegativeInteger" />
        <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="urlRewrite" type="URLRewriteType" maxOccurs="unbounded"/>
        <xs:element name="urlResign" type="URLResignType" minOccurs="0" maxOccurs="1"/>
        <xs:element name="urlInHttp" type="xs:positiveInteger" minOccurs="0" maxOccurs="1"/>
        <xs:element name="urlGenerate" type="URLGenerateType" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:positiveInteger"/>

```

```

    <xs:attribute name="uri" type="xs:string"/>
  </xs:complexType>
</xs:element>
<xs:element name="CDNSelPolicies" type="CDNSelectPoliciesType" />
<xs:element name="CDNSelPolicy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string" />
      <xs:element name="description" type="xs:string" minOccurs="0"/>
      <xs:element name="enableAdvancedAction" type="BooleanType" />
      <xs:element name="preferOnNetCDN" type="BooleanType" minOccurs="0"/>
      <xs:element name="rule" type="RuleType" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="allocation" type="TimeSplitType" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element name="cdnRankCriteria" type="RankingCriteriaType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:positiveInteger"/>
    <xs:attribute name="uri" type="xs:string"/>
  </xs:complexType>
</xs:element>
</xs:schema>

```